

# KSC PostgreSQL

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

## Настройка PostgreSQL

Настройки на сервере БД PostgreSQL можно выполнять в консоли (SSH, терминал ОС) или средствами утилиты pgAdmin (требуется установка).

В данной статье сервер БД PostgreSQL работает под управлением ОС Astra Linux, а все настройки выполняются **в консоли**.

Для удобства в базе данных PostgreSQL Kaspersky Security Center предусмотрен набор публичных представлений. Таким образом можно создавать запросы непосредственно к публичным представлениям и извлекать из них данные о событиях. "Коробочный" коннектор KUMA **[OOTB] KSC PostgreSQL** содержит уже готовые запросы к публичным представлениям **v\_akpub\_ev\_event** и **v\_akpub\_host**.

Коннектор **[OOTB] KSC PostgreSQL** позволяет экспортировать события из БД PostgreSQL Kaspersky Security Center (KSC) версии 15.0.

Проверена работоспособность коннектора с KSC 14.2 Windows (БД PostgreSQL).

## Проверка имени БД KSC

Чтобы проверить имя базы данных KSC можно воспользоваться следующей статьей:

<https://support.kaspersky.ru/ksc-linux/15/228689> (KSC Linux)

Альтернативным вариантом является проверка имени БД в консоли сервера, на котором установлена БД KSC:

- Измените текущего пользователя на **postgres**

```
sudo -i -u postgres
```

- Запустите интерактивный терминал PostgreSQL и выведите список баз данных сервера

```
psql
```

```
\l # вывод списка баз данных сервера
```

```
postgres=# \l
```

Список баз данных					
Имя	Владелец	Кодировка	LC_COLLATE	LC_CTYPE	Права доступа
KAV	postgres	UTF8	ru_RU.UTF-8	ru_RU.UTF-8	

## Создание роли БД и предоставление прав

- В консоли сервера, где установлена БД PostgreSQL KSC, измените текущего пользователя на **postgres**

```
sudo -i -u postgres
```

- Запустите интерактивный терминал PostgreSQL

```
psql
```

Также для создания роли БД и предоставления ей соответствующих прав можно использовать существующую учетную запись с атрибутом **role creation** и правами доступа к публичным представлениям.

Пример подключения к БД:

```
psql -U <имя УЗ> -d <имя БД KSC>
```

- Создайте роль пользователя kuma

```
CREATE USER kuma WITH PASSWORD '<задайте пароль>';
```

- Подключитесь к БД KSC (см. Раздел "**Проверка имени БД KSC**". По умолчанию KAV)

```
\connect KAV
```

- Предоставьте права роли KUMA

```
GRANT SELECT ON v_akpub_ev_event TO kuma;  
GRANT SELECT ON v_akpub_host TO kuma;  
GRANT SELECT ON v_akpub_virus_activity TO kuma;  
GRANT SELECT ON v_akpub_hst_prdstate TO kuma;  
GRANT SELECT ON v_akpub_host_status TO kuma;
```

## Настройка удаленного доступа к БД PostgreSQL и метода аутентификации

- Откройте файл **/etc/postgresql/<версия БД PostgreSQL>/main/pg\_hba.conf** и в секции **IPv4 local connections** добавьте следующую строку

```
host <имя БД, например, KAV> kuma <IP-адрес коллектора KUMA>/32 scram-sha-256
```

```
# IPv4 local connections:  
host all all 127.0.0.1/32 scram-sha-256  
host KAV kuma <IP-адрес коллектора KUMA>/32 scram-sha-256
```

- Откройте файл конфигурации **/etc/postgresql/<версия БД PostgreSQL>/main/postgresql.conf** и в секции **CONNECTIONS AND AUTHENTICATION** укажите IP-адрес интерфейса сервера БД, на котором будут "прослушиваться" входящие соединения

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
listen_addresses = 'localhost, <IP-адрес интерфейса>' # what IP address(es) to listen on;  
# comma-separated list of addresses;  
# defaults to 'localhost'; use '*' for all  
# (change requires restart)  
port = 5432 # (change requires restart)  
max_connections = 151 # (change requires restart)  
#superuser_reserved_connections = 3 # (change requires restart)  
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories  
# (change requires restart)  
#unix_socket_group = '' # (change requires restart)  
#unix_socket_permissions = 0777 # begin with 0 to use octal notation  
# (change requires restart)  
#bonjour = off # advertise server via Bonjour  
# (change requires restart)  
#bonjour_name = '' # defaults to the computer name  
# (change requires restart)
```

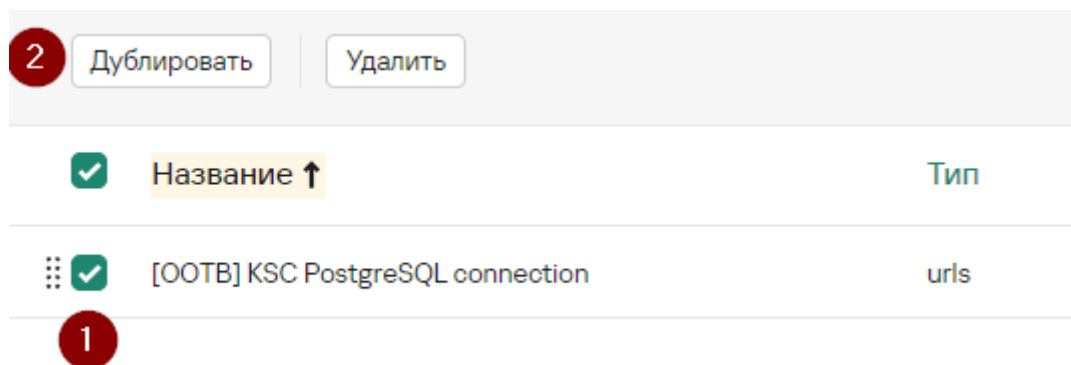
- После внесения изменений и сохранения файла конфигурации выполните рестарт сервиса PostgreSQL

```
systemctl restart postgresql
```

При необходимости разрешите входящие соединения на порт БД PostgreSQL (по умолчанию, TCP/5432) в параметрах локального FW.

## Создание секрета KUMA

1. В веб-интерфейсе KUMA перейдите на вкладку **Ресурсы → Секреты**
2. Выберите секрет **[OOTB] KSC PostgreSQL connection** и нажмите **Дублировать**.



3. В появившемся окне задайте:

- Название секрета
- URL (формат URL можно взять из **Описания** к секрету). В поле URL укажите:
  - Имя ранее созданной роли (в нашем примере это **kuma**) и ее пароль;
  - IP-адрес или FQDN сервера БД;
  - Наименование БД KSC (по умолчанию KAV. См. **Проверка имени БД KSC**).

\*Название

[Custom][OOTB] KSC PostgreSQL connection 1

\*Тенант

Main

\*Тип

urls

\*URL

postgres://<user>:<password>@<server>/<database> 2

Описание

KSC PostgreSQL connection.  
postgres://<user>:<password>@<server>/<database>  
By default, the database name is KAV.

Если в пароле используются спецсимволы необходимо перевести данные спецсимволы в URL формат в соответствии с таблицей ниже.


Примеры

postgres://kuma:p%40ssword123%21@server.demo.lab/KAV

!	#	\$	%	&	'	(	)	*	+
%21	%23	%24	%25	%26	%27	%28	%29	%2A	%2B
,	/	:	;	=	?	@	[	]	\
%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D	%5C

Можно использовать следующий ресурс для преобразования <https://www.urlencoder.org/>

Пример:


 To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 ▾ Destination character set.

LF (Unix) ▾ Destination newline separator.

☐ Encode each line separately (useful for when you have multiple entries).

☐ Split lines into 76 character wide chunks (useful for MIME).

 Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

**> ENCODE <** Encodes your data into the area below.

P%40%24%24w0d

По умолчанию коллектор KUMA при обращении к БД PostgreSQL будет пытаться построить TLS-туннель. Если на стороне сервера БД не настроено использование SSL/TLS (что НЕ рекомендуется!) в URL секрета необходимо добавить **"?sslmode=disable"**, чтобы строка приняла следующий вид:  
postgres://user:password@server/database?**sslmode=disable**

4. Нажмите **Сохранить**.

Обратите внимание, после сохранения секрета поле URL будет пустым во избежание утечки чувствительной информации.

## Настройка коннектора

1. В веб-интерфейсе KUMA перейдите в раздел **Ресурсы** → **Коннекторы**

2. В списке коннекторов справа найдите коннектор **[OOTB] KSC PostgreSQL** и нажмите **Дублировать**

2

Дублировать

Удалить

<input checked="" type="checkbox"/>	Название ↑	Тип
<div><div></div><input checked="" type="checkbox"/></div>	[OOTB] KSC PostgreSQL	sql

1

3. В появившемся окне задайте:

- Название коннектора
- На вкладке **Основные параметры** в выпадающих списках **URL** выберите секрет, созданный ранее для подключения к БД PostgreSQL KSC.

\*URL

1

[Custom][OOTB] KSC PostgreSQL conne... ▼

+

\*Столбец идентификатора

externalId

\*Начальное значение идентификатора

1

\*URL

[Custom][OOTB] KSC PostgreSQL conne... ▼

+

\*Столбец идентификатора

infoUpdate

\*Начальное значение идентификатора

2023-01-01T00:00:00.000

4. Нажмите **Сохранить**.

## Настройка коллектора

1. В веб-интерфейсе KUMA перейдите в раздел **Ресурсы** и нажмите **Подключить источник**
2. В появившемся окне задайте **Название коллектора** и **Тенант**

3. На шаге **Транспорт** выберите ранее созданный коннектор
4. На шаге **Парсинг событий** выберите нормализатор **[OOTB] KSC from SQL**.
5. На шаге **Маршрутизация** задайте следующие точки назначения:
  - **Хранилище**. Для отправки обработанных событий в хранилище.
  - **Коррелятор**. Для отправки обработанных событий в коррелятор.
6. На шаге **Проверка параметров** нажмите **Сохранить и создать сервис**.
7. Скопируйте появившуюся команду и выполните установку сервиса коллектора.

## Проверка поступления событий KSC в KUMA

Для проверки, что экспорт событий из БД KSC успешно настроен, перейдите в **Ресурсы -> Активные сервисы** -> выберите ранее созданный коллектор **KSC PostgreSQL** и нажмите **Перейти к событиям**.

[Ресурсы и сервисы](#) > [Сервисы](#) Добавить сервис Обновить

Обновить параметры Перезапустить Копировать идентификатор Перейти к событиям

<input type="checkbox"/>	Статус	Тип	Сервис ↓	Версия	Тенант
<input type="checkbox"/>	●	Хранилище	<a href="#">[OOTB] Storage</a>	2.1.3.49	Main
<input type="checkbox"/>	●	Коллектор	<a href="#">[OOTB] KSC SQL</a>	2.1.3.49	Main
<input type="checkbox"/>	●	Коррелятор	<a href="#">[OOTB] Correlator</a>	2.1.3.49	Main
<input checked="" type="checkbox"/> 1	●	Коллектор	<a href="#">[Custom] KSC PostgreSQL</a>	2.1.3.49	Main

В открывшемся окне **События** убедитесь, что присутствуют события KSC.



События



SELECT \* FROM `events` WHERE ServiceID = '08d27e3e-6b3c-4445-b505-bf481045dd6d' ORDER BY Timestamp DESC LIMIT 250

TenantID	Timestamp ↓	Name	DeviceProduct	DeviceVendor	DeviceEventCategory
Main	09.01.2024 13:23:47	Application_Control	KSC	Kaspersky	Запуск приложения заблокирован в тестовом режиме
Main	09.01.2024 13:23:47	Application_Control	KSC	Kaspersky	Запуск приложения заблокирован в тестовом режиме
Main	09.01.2024 13:23:07	Application_Control	KSC	Kaspersky	Запуск приложения заблокирован в тестовом режиме
Main	09.01.2024 13:23:07	Application_Control	KSC	Kaspersky	Запуск приложения заблокирован в тестовом режиме

Revision #15  
Created 28 December 2023 14:05:59 by Dmitry Borisov  
Updated 7 July 2024 08:57:02 by Koala