KEDR (OSMP)

- Установка и обновление
  - Гайд по установке Kaspersky EDR Expert (on-premise) 8.0
- Инструкции по настройке
  - Оптимизация дисковой подсистемы
  - Настройка доменной аутентификации Kerberos
- Диагностика и решение проблем
  - KEDR Expert on-premise: общие рекомендации по диагностике и устранению неполадок
  - Очистка событий в хранилище (освобождение места)

?????????? ? ???????????

???? ?? ?????????? Kaspersky EDR Expert (on-premise) 8.0

> Инструкция составлена на основе официальной документации и опыта эксплуатации

????? 1. ???? ?????????

Параметр	Стандартная конфигурация (несколько узлов)	Демонстрационная конфигурация (один узел)
СУБД PostgreSQL	Устанавливается **вне кластера Kubernetes** на отдельном сервере	Устанавливается на одном хосте с **кластером Kubernetes**
Минимальное количество узлов	1 первичный + 3 рабочих узла + сервер СУБД + 1 узел администратора (опционально)	1 узел (все компоненты на одном устройстве) + 1 узел администратора (опционально)
Назначение	Промышленная эксплуатация	Тестирование, демонстрация, обучение

???? ?????:

- **Узел администратора** - устройство с утилитой KDT (Kaspersky Deployment Toolkit) для развертывания и управления компонентами OSMP.
- **Primary/master/controller/первичный рабочий узел** - узел контроллера, осуществляющий управление кластером k0s.
- **Worker/рабочий узел** - узел кластера k0s с полезной нагрузкой.
- **DB/СУБД** - сервер с СУБД для кластера OSMP.
- **KUMA services/устройство с сервисами KUMA** - устройства с установленными сервисами KUMA: коллектор, коррелятор, хранилище (в случае KEDR входят в состав Kubernetes кластера).
- **Целевые устройства** - устройства, на которых устанавливается OSMP (все вышеперечисленные узлы)

????? 2. ??????????? ?????????? ?????????????

2.1. ???? PostgreSQL

Параметр	Требование

Версия	15.7 или выше
Расположение	Вне кластера Kubernetes (стандартная конфигурация)
Привилегированная учётная запись	Требуется учётная запись с правами суперпользователя для создания баз данных во время развертывания
Поддержка кластеров	Поддерживается синхронная репликация (минимум 3 узла, максимум 15).
Дисковая подсистема	SSD/NVMe рекомендуется

2.2. ??????? ??????????

Требование	Описание
Широковещательный домен	Все целевые устройства кластера Kubernetes **должны находиться в одном широковещательном домене** (одна L2-сеть)
Статические IPv4-адреса	Все узлы кластера и шлюз Kubernetes должны иметь статические IPv4-адреса в одной подсети
Синхронизация времени	Разница во времени между узлами не должна превышать 5 секунд (рекомендуется использовать NTP)
DNS	Должна быть настроена зона для домена smp_domain (например, smp.local) с записями для всех сервисов

2.3. ?????????? ?????????? (???????????)

Компонент	Процессор	ОЗУ	Дисковая подсистема
Все компоненты на одном узле (демонстрационная конфигурация)	12 ядер	56 Гб	1300 Гб

Для корректного развертывания решения убедитесь, что процессор целевого устройства (компонентов KEDR) поддерживает набор инструкций BMI, AVX и SSE 4.2.

2.4. ??????????? ??????????

Требования к программному обеспечению и поддерживаемым системам и платформам

<table>
<tr><td>Операционная система</td><td>OSMP с компонентами KUMA

Поддерживаются следующие 64-разрядные версии операционных систем:
Ubuntu Server 22.04 LTS.
Ubuntu Server 24.04 LTS.
Debian GNU/Linux 12.x (Bookworm).
Astra Linux Special Edition РУСБ.10015-01 (очередное обновление 1.8.1).
<blockquote>На целевых устройствах с операционными системами семейства Ubuntu версия ядра Linux должна быть 5.15.0.107 или выше.</blockquote></td></tr>
<tr><td>Платформы виртуализации</td><td><table>
<tr><th>Название платформы виртуализации</th><th>OSMP</th><th>Sandbox (опционально)</th></tr>
<tr><td>VMware ESXi 6.7.0 или 7.0;</td><td>Есть.</td><td>Есть.</td></tr>
<tr><td>KVM;</td><td>Есть.</td><td>Нет.</td></tr>
<tr><td>ПК СВ "Брест" 3.3;</td><td>Есть.</td><td>Есть.</td></tr>
<tr><td>"РЕД Виртуализация" 7.3;</td><td>Есть.</td><td>Есть.</td></tr>
<tr><td>zVirt Node 4.2.</td><td>Есть.</td><td>Есть.</td></tr>
</table></td></tr>
<tr><td>Система управления базами данных (СУБД)</td><td>PostgreSQL 15.x 64-разрядная
PostgreSQL 16.x 64-разрядная
PostgreSQL 17.x 64-разрядная
Postgres Pro 15.x (все редакции) 64-разрядная
Postgres Pro 16.x (все редакции) 64-разрядная
Postgres Pro 17.x (все редакции) 64-разрядная
Postgres Pro 16.x Enterprise 64-разрядная (кластер Built-in High Availability).
Postgres Pro 17 Enterprise 64-разрядная (кластер Built-in High Availability).</td></tr>
</table>

????? 3. ?????????? ?????????

> Отключайте файл подкачки (swap) в продуктовых средах

?????????? ?????????????:

- Установка обязательных пакетов на устройстве администратора:

```
sudo apt update
sudo apt install -y python3
```

Установите пакет для Docker версии 23 или выше, а затем выполните действия после установки, чтобы настроить устройство администрирования для правильной работы с Docker.

Пример:

```
Для Ubuntu:
# Удалите старые версии
sudo apt remove docker docker-engine docker.io containerd runc

# Установите зависимости
sudo apt update
sudo apt install ca-certificates curl gnupg lsb-release

# Добавьте официальный GPG ключ Docker
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

# Добавьте репозиторий
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

# Установите Docker
sudo apt update
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose-plugin

# Добавьте пользователя в группу docker
sudo usermod -aG docker $USER

# Настройте автозапуск
sudo systemctl enable docker.service
sudo systemctl enable containerd.service

# Перезагрузитесь или выполните
newgrp docker
```

Ещё вариант:

```
apt install docker.io
```

- Генерация SSH-ключа (без парольной фразы):

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa -N ""
```

Скопируйте ключ на все целевые устройства:

```
ssh-copy-id username@<IP_целевого_устройства>

# Проверка подключения без пароля
ssh username@<IP_целевого_устройства> "sudo whoami"
```

> При установке от учётной записи **root** убедитесь, что ключ на целевых устройствах располагается по пути **/root/.ssh/authorized_keys**

??????? ?????????? (OSMP):

- Проверка cgroup v2:

```
mount | grep cgroup
# Должно быть: cgroup2 on /sys/fs/cgroup type cgroup2
```

- Отключите SELinux (если установлен)

```
# Проверка статуса
getenforce

# Отключение (требуется перезагрузка)
sudo setenforce 0 sudo sed -i 's/^SELINUX=enforcing/SELINUX=disabled/' /etc/selinux/config
```

- Настройте proxy (если требуется)

```
# Отредактируйте /etc/environment
sudo nano /etc/environment

# Добавьте:
HTTP_PROXY="http://proxy.example.com:8080"
HTTPS_PROXY="http://proxy.example.com:8080"
```

```
NO_PROXY="localhost,127.0.0.1,<IP_адреса_узлов_кластера>"
```

- Настройте firewall (если используется)

```
# Разрешите SSH
sudo ufw allow 22/tcp

# Разрешите Kubernetes порты
sudo ufw allow 6443/tcp
sudo ufw allow 2379:2380/tcp
sudo ufw allow 10250/tcp

# Разрешите PostgreSQL порты
sudo ufw allow 5432/tcp

# Включите IP forwarding для primary/worker node
sudo sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' /etc/sysctl.conf
sudo sysctl -p
```

Для отключения firewall'а выполните:

```
systemctl stop ufw
systemctl disable ufw
```

- Установка обязательных пакетов:

```
# Общие пакеты для всех узлов:
sudo apt update
sudo apt install -y sudo nfs-common tar wireguard wireguard-tools python3-apt

# Для первичного узла дополнительно:
sudo apt install -y curl

# Для рабочих узлов дополнительно (наименования могут отличаться в зависимости от выбранного дистрибутива Linux):
sudo apt install -y libnfs12 iscsi-package
```

- Настройка беспарольного sudo:

```
# Для пользователя, который будет использоваться KDT:
echo "username ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers
```

Это позволит учетной записи иметь возможность повышать привилегии (sudo) без ввода пароля

- Настройка синхронизации времени:

```
sudo timedatectl set-ntp true
```

- Настройка IP-переадресации (только для первичного узла с UFW):

```
# В файле /etc/default/ufw установите:
DEFAULT_FORWARD_POLICY="ACCEPT"

# Примените изменения:
sudo ufw reload
```

???? PostgreSQL:

- Установка обязательных пакетов:

```
sudo apt update
sudo apt install -y postgresql
```

- Настройка параметров в **postgresql.conf:**

```
nano /etc/postgresql/<ВЕРСИЯ>/main/postgresql.conf

# Переопределите следующие параметры:
listen_addresses = '*'
port = 5432
max_connections = 512
shared_buffers = 8GB                  # 25% от ОЗУ, минимум 3 ГБ
effective_cache_size = 24GB           # 75% от ОЗУ
temp_buffers = 24MB
work_mem = 64MB
maintenance_work_mem = 1GB
max_stack_depth = 7MB                 # Для Linux: ulimit -s минус 1 МБ
effective_io_concurrency = 200        # 200 для SSD или 2 для HDD
max_parallel_workers_per_gather = 0
wal_buffers = 64MB
max_wal_size = 4GB
min_wal_size = 1GB
random_page_cost = 1.1                # 1.1 для SSD или 4.0 для HDD
```

```
log_hostname = 1
standard_conforming_strings = on      # Обязательно должно быть 'on'
```

- Разрешение удаленного подключения к СУБД:

```
nano /etc/postgresql/<ВЕРСИЯ>/main/pg_hba.conf

# В секции 'IPv4 local connections' переопределите значение:
host   all   all   0.0.0.0/0    scram-sha-256
```

- Перезапуск службы:

```
systemctl restart postgresql
```

- Создание привилегированной учётной записи:

```
# Подключитесь к PostgreSQL:
sudo -u postgres psql

# Создайте учётную запись с правами суперпользователя и базу, например:
CREATE USER <kaspersky_admin> WITH PASSWORD '<StrongPassword123!>' SUPERUSER;
CREATE DATABASE <kaspersky_admin> OWNER <kaspersky_admin>;
```

?????????? ??????????????:

1. Зарезервируйте IP-адрес из той же подсети, что и у серверов Primary/Worker. Адрес должен быть свободен и будет назначен в процессе установки (указывается в файле **param.yaml** в поле **ingress_ip**).

2. Добавьте в DNS-зону вашей организации (предпочтительно создать поддомен) следующие записи:

```
console.<smp_domain> <ingress_ip>
admsrv.<smp_domain> <ingress_ip>
api.<smp_domain> <ingress_ip>
monitoring.<smp_domain> <ingress_ip>
updater.<smp_domain> <ingress_ip>
agentserver.<smp_domain> <ingress_ip>
kuma.<smp_domain> <ingress_ip>
*.kuma.<smp_domain> <ingress_ip>
```

где, **<smp_domain>** - домен или поддомен вашей организации (данный домен должен быть указан в файле параметров (param.yaml) в одноименном поле **smp_domain**), а **<ingress_ip>**

- зарезервированный IP из предыдущего пункта.

> Обратите внимание, что для имени **kuma** необходимо создать wildcard (*) на DNS-сервере

????? 4. ????? ?????????

?????????? ????? param.yaml:

> Для корректной работы KDT с конфигурационным файлом добавьте пустую строку в конце файла

Формат файла для стандартной конфигурации с сервисами KUMA внутри кластера

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
nodes:
  - desc: cdt-primary1
    type: primary
    host: "<IPv4_первичного_узла>"
    access:
      ssh:
        user: "<имя_пользователя>" # По умолчанию root
        key: "<путь_к_закрытому_ключу>" # По умолчанию /root/.ssh/id_rsa
  - desc: cdt-w1
    type: worker
    host: "<IPv4_рабочего_узла_1>"
    access:
      ssh:
        user: "<имя_пользователя>"
        key: "<путь_к_закрытому_ключу>"
    kind: admsrv  # Сервер администрирования будет установлен на этом узле
  - desc: cdt-w2
    type: worker
    host: "<IPv4_рабочего_узла_2>"
```

```
    access:
      ssh:
        user: "<имя_пользователя>"
        key: "<путь_к_закрытому_ключу>"
    kuma_roles: ["storage"]  # Хранилище закреплено за этим узлом
  - desc: cdt-w3
    type: worker
    host: "<IPv4_рабочего_узла_3>"
    access:
      ssh:
        user: "<имя_пользователя>"
        key: "<путь_к_закрытому_ключу>"
parameters:
  - name: psql_dsn
    source:
      value: "postgres://<dbms_username>:<password>@<fqdn_СУБД>:<порт_СУБД>" # Если пароль
содержит спецсимволы, их необходимо привести к URI кодировке
  - name: ingress_ip
    source:
      value: "<IPv4_шлюза_кластера_Kubernetes>" # Ранее зарезервированный ingress_ip
  - name: ssh_pk
    source:
      path: "<путь_к_закрытому_ключу_на_устройстве_администратора>"
  - name: admin_password
    source:
      value: "<пароль_учётной_записи_admin>"
  - name: core_disk_request
    source:
      value: "512Gi"
  - name: inventory
    source:
      value: "/dev/null"  # Все сервисы KUMA будут внутри кластера Kubernetes
  - name: license
    source:
      value: "<путь_к_лицензионному_ключу>" # Рекомендуется файл переименовать в
license.key
  - name: smp_domain
    source:
      value: "<доменное_имя>"
```

```
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring agentserver updater"
  - name: low_resources
    source:
      value: "false"
  - name: nwc-language
    source:
      value: "ruRu"  # Или "enUS"
  - name: skip_preflight_checks
    source:
      value: "false"
  - name: openbao_ha_mode
    source:
      value: "true"
```

Формат файла для демонстрационной конфигурации с сервисами KUMA внутри кластера

```
schemaType: ParameterSet
schemaVersion: 1.0.1
namespace: ""
name: bootstrap
project: xdr
nodes:
  - desc: "single-node"
    type: primary-worker
    host: "<IPv4_единственного_узла>"
    access:
      ssh:
        user: "<имя_пользователя>" # По умолчанию root
        key: "<путь_к_закрытому_ключу>" # По умолчанию /root/.ssh/id_rsa
parameters:
  - name: psql_dsn
    source:
      value: "postgres://<dbms_username>:<password>@<fqdn_СУБД_в_кластере>:<порт>"  # Если пароль содержит спецсимволы, их необходимо привести к URI кодировке
  - name: ingress_ip
```

```
    source:
      value: "<IPv4_шлюза_кластера_Kubernetes>" # Ранее зарезервированный ingress_ip
  - name: ssh_pk
    source:
      path: "<путь_к_закрытому_ключу>"
  - name: admin_password
    source:
      value: "<пароль_учётной_записи_admin>"
  - name: inventory
    source:
      value: "/dev/null" # Все сервисы KUMA будут внутри кластера Kubernetes
  - name: license
    source:
      value: "<путь_к_лицензионному_ключу>" # Рекомендуется файл переименовать в
license.key
  - name: smp_domain
    source:
      value: "<доменное_имя>"
  - name: pki_host_list
    source:
      value: "admsrv api console kuma monitoring agentserver updater"
  - name: low_resources
    source:
      value: "true"
  - name: openbao_ha_mode
    source:
      value: "false"
  - name: openbao_standalone
    source:
      value: "true"
  - name: default_class_replica_count
    source:
      value: 1
```

?????????

1. Загрузите на устройство администратора следующие файлы:

- `bin.tar.gz` - архив с утилитой KDT, используется для развертывания и изменения параметров инсталляции

- `xdr-<version>.tar.gz` - транспортный архив с компонентами SMP
- `license.key` - ключ лицензии

2. Распакуйте архив `bin.tar.gz`

```
tar -xzf bin.tar.gz
```

3. Дайте файлу kdt права на выполнение:

```
chmod +x ./kdt
```

4. Запустите установку командой:

```
./kdt apply -k <полный путь к транспортному архиву> -i <полный путь к файлу конфигурации param.yaml> --accept-eula
```

????? 5. ??????? ???????? ????? ?????????

Первый вход в веб-интерфейс:

1. Откройте в браузере **https://console.<smp_domain>**
2. Войдите под учётной записью **admin** с паролем из параметра **admin_password** и завершите настройку 2FA.

Активация лицензии:

1. В главном окне приложения нажмите на имя Сервера администрирования. Откроется окно свойств Сервера администрирования
2. На вкладке **Общие** выберите раздел **Лицензионные ключи**
3. В разделе **Действующая лицензия** нажмите на кнопку **Выбрать**
4. В открывшемся окне выберите лицензионный ключ, который вы хотите использовать для активации Kaspersky EDR Expert (on-premise) 8.0. Если лицензионного ключа нет в списке, нажмите на кнопку **Добавить лицензионный ключ** и укажите новый лицензионный ключ
5. Нажмите на кнопку **Сохранить**

Предоставление доступа к функционалу Kaspersky EDR:

1. В главном окне приложения перейдите в **Параметры**, выберите раздел **Тенанты** и нажмите на **Root tenant**
2. На вкладке **Права доступа** нажмите на кнопку **Добавить пользователя**
3. Выберите пользователя **admin** и предоставьте ему роль **Главный администратор**
4. Нажмите кнопку **Сохранить** и далее **Сохранить и закрыть**

Установка плагинов управления:

1. Загрузите плагин управления конечным устройством для OSMP с расширением .tar на узел администратора
2. Выполните установку командой:

```
./kdt apply -k <полный путь к tar-архиву с плагином управления>
```

Обновление баз Kaspersky EDR:

1. В главном окне приложения перейдите в **Параметры**, выберите раздел **Тенанты** и нажмите на **Root tenant**
2. На вкладке **Параметры** перейдите к **Параметры сканирования** и на вкладку **Обновить базы**
3. Запустите обновление и дождитесь успешного статуса

Подключение Sandbox:

1. В главном окне приложения перейдите в **Параметры**, выберите раздел **Тенанты** и нажмите на **Root tenant**
2. На вкладке **Параметры** перейдите к **Серверы Sandbox** и нажмите кнопку **Добавить**
3. Укажите IP-адрес KATA Sandbox 8 версии, получите отпечаток, задайте произвольное имя, нажмите кнопку **Добавить**
4. В браузере перейдите и авторизуйтесь в веб-консоли Sandbox и примите запрос на подключение в разделе **Авторизация**
5. В консоли OSMP дождитесь, что статус авторизации изменился на **Одобрено** и отобразились наборы виртуальных машин

Настройка политик для подключения телеметрии с конечных устройств:

1. В главном окне приложения перейдите в **Управление активами - Политики**
2. Создайте политику или откройте её свойства и перейдите в **Параметры приложения**
3. Раскройте ветку **Встроенные агенты** и перейдите в параметры для **Endpoint Detection and Response Expert (on-premise)**
4. Установите переключатель в статус **ВКЛЮЧЕН.** Установите переключатель замка, чтобы изменение применялось на конечном узле
5. Выберите **Endpoint Detection and Response Expert (версия 8.0 и выше)**
6. Для раздела **Подключение к серверам сбора телеметрии** нажмите кнопку **Добавить** и нажмите **Выбрать сервер**
7. В открывшемся окне отметьте единственный сервер коллектора и нажмите кнопку **Добавить**
8. Ниже повторите действия для указания сервера реагирования
9. При необходимости настройте другие пункты политики и сохраните изменения

Настройка политик для подключения Sandbox к конечным устройствам:

1. В главном окне приложения перейдите в **Параметры**, выберите раздел **Тенанты** и нажмите на **Root tenant**
2. На вкладке **Параметры** перейдите к **Сертификаты** и нажмите кнопку **Экспортировать** сертификат сервера и ниже сертификат Endpoint Agent, указав пароль
3. В главном окне приложения перейдите в **Управление активами - Политики**
4. Создайте политику или откройте её свойства и перейдите в **Параметры приложения**
5. Раскройте ветку **Встроенные агенты** и перейдите в параметры для **Sandbox**
6. Установите переключатель в статус **ВКЛЮЧЕН.** Установите переключатель замка, чтобы изменение применялось на конечном узле
7. Выберите режим интеграции **KATA Sandbox**
8. Для раздела **Подключение к серверам Sandbox** нажмите кнопку **Добавить** и укажите адрес сервера **agentserver.<smp_domain>**
9. Нажмите **Настройки подключения** и загрузите экспортированный сертификат сервера. В этом же поле обязательно включите использование двусторонней аутентификации и подставьте сертификат Endpoint Agent с паролем, указанным при его экспорте
10. При необходимости настройте другие пункты политики и сохраните изменения

Базовая настройка завершена. Подключите совместимые приложения KES для Windows 12.11 и выше, KES для Linux 12.4, KES для Mac 12.2.1 и приступайте к работе с Kaspersky EDR Expert (on-premise).

?????????? ?? ??????????

Инструкции по настройке

?????????? ???????? ??????????

[В данной статье](#) на XDR описаны общие рекомендации, которые также применимы к функциональности KEDR. Нас интересует именно требования к дискам и хранилищам.

Перед дальнейшими изменениями можете выполнить команду на сервере СУБД и записать результаты:

```
fio --name=test --directory=/var/lib/postgresql/<ваша_версия>/main --rw=randrw --bs=8k --
size=1G --iodepth=16 --ioengine=libaio --direct=1 --fsync=1 --runtime=60
```

Для начала выполним быструю проверку типа диска:

```
lsblk -d -o name,rota,TYPE,MODEL,SIZE
```

Ожидаемые выводы и их интерпритация:

```
#Пример 1: SSD
NAME    ROTA    TYPE    MODEL              SIZE
sda        0    disk    Samsung_SSD_860    500G   # ROTA=0 (SSD)

#Пример 2: HDD
NAME    ROTA    TYPE    MODEL              SIZE
sda        1    disk    WDC_WD10EZEX       1T     # ROTA=1 (HDD)

#Пример 3: Виртуальный диск (VMware)
NAME    ROTA    TYPE    MODEL              SIZE
sda        0    disk    Virtual_disk       100G   # ROTA=0 (считаем как SSD)

#Пример 4: NVMe
NAME    ROTA    TYPE    MODEL              SIZE
nvme0n1    0    disk    INTEL_SSD          500G   # NVMe (всегда SSD)
```

Далее настроим планировщие I/O:

Создадим udev правило и отредактируем его:

```
nano /etc/udev/rules.d/60-scheduler.rules
```

Для SSD (ROTA=0) вставим следующие значения:

```
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="0",
ATTR{queue/scheduler}="mq-deadline"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="0",
ATTR{queue/nr_requests}="1024"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="0",
ATTR{queue/max_sectors_kb}="2048"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="0",
ATTR{queue/read_ahead_kb}="128"
```

Для HDD (ROTA=1):

```
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/scheduler}="mq-deadline"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/nr_requests}="256"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/max_sectors_kb}="512"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/read_ahead_kb}="256"
```

Для NVMe (ROTA=0):

```
ACTION=="add|change", KERNEL=="nvme[0-9]*", ATTR{queue/scheduler}="none"
ACTION=="add|change", KERNEL=="nvme[0-9]*", ATTR{queue/nr_requests}="1024"
```

И применим правила:

```
udevadm control --reload-rules
udevadm trigger --name-match=sda #изменить имя диска, если отличается
```

Далее настроим параметры ядра (sysctl):

Создадим файл настроек и отредактируем его:

```
nano /etc/sysctl.d/99-database-tuning.conf
```

Для SSD / NVMe / Виртуальных дисков (ROTA=0):

```
vm.dirty_ratio = 10
vm.dirty_background_ratio = 5
vm.dirty_expire_centisecs = 3000
vm.dirty_writeback_centisecs = 500

fs.aio-max-nr = 2097152

vm.swappiness = 1
vm.vfs_cache_pressure = 50

net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
```

Для HDD (ROTA=1):

```
vm.dirty_ratio = 30
vm.dirty_background_ratio = 10
vm.dirty_expire_centisecs = 6000
vm.dirty_writeback_centisecs = 3000

fs.aio-max-nr = 2097152

vm.swappiness = 1
vm.vfs_cache_pressure = 50

net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
```

И применем настройки:

```
sysctl -p /etc/sysctl.d/99-database-tuning.conf
```

Далее настроим опции файловой системы:

Отредактируем fstab (предварительно сделайте резервную копию):

```
nano /etc/fstab
```

Найдём строку, как пример для ext4:

```
UUID=3fe51788-bb69-467d-89a9-a146c1df7fdd /                       ext4    defaults        1 1
```

Заменим на:

```
UUID=3fe51788-bb69-467d-89a9-a146c1df7fdd /                       ext4
rw,noatime,nodiratime,data=ordered,errors=remount-ro       1 1
```

Для xfs заменим на: **noatime,nodiratime,logbufs=8,logbsize=256k**

Далее перемонтируем корневую систему:

```
mount -o remount /
```

или перезагрузим систему.

Также рекомендую обновить systemd после изменения /etc/fstab:

```
systemctl daemon-reload
```

Далее настроим TRIM (только для SSD):

```
# Включить fstrim.timer:
systemctl enable --now fstrim.timer

# Проверить статус:
systemctl status fstrim.timer
```

Далее перепроверим конфигуарцию postgresql.conf:

```
nano /etc/postgresql/<ВЕРСИЯ>/main/postgresql.conf

# Переопределите следующие параметры:
listen_addresses = '*'
port = 5432
max_connections = 512
shared_buffers = 8GB                  # 25% от ОЗУ, минимум 3 ГБ
effective_cache_size = 24GB           # 75% от ОЗУ
temp_buffers = 24MB
work_mem = 64MB
maintenance_work_mem = 1GB
```

```
max_stack_depth = 7MB                 # Для Linux: ulimit -s минус 1 МБ
effective_io_concurrency = 200        # 200 для SSD или 2 для HDD
max_parallel_workers_per_gather = 0
wal_buffers = 64MB
max_wal_size = 4GB
min_wal_size = 1GB
random_page_cost = 1.1                # 1.1 для SSD или 4.0 для HDD
log_hostname = 1
standard_conforming_strings = on      # Обязательно должно быть 'on'
```

Повторно можете выполнить команду fio и сравнить с первоначальными результатами. Должны увидеть рост пропускной способности, уменьшение задержек, увеличение числа операций в единицу времени.

????????? ???????? ????????????? Kerberos

Процесс настройки доменной аутентификации описан в официальной справке. Обратите внимание, что keytab выписывается на console.<smp.domain>.

Если после этого возникли проблемы с авторизацией или всплывает окно для ввода учетных данных, то:

1) Проверка keytab файла

Установите MIT Kerberos for Windows: https://web.mit.edu/kerberos/dist/kfw/

И выполните:

```
"C:\Program Files (x86)\MIT\Kerberos\bin\klist.exe" -k -t -e C:\<путь>\<имя>.keytab
```

Правильный вывод:

KVNO	Timestamp	Principal
3	01/01/70 03:00:00	HTTP/console.xdr.sales.lab@SALES.LAB (AES-256 CTS...)

Частые ошибки это двойные бэкслеши **HTTP\\console...**, либо неправильное шифрование **etype: arcfour-hmac**, либо kvno указан иной. В этом случае, то пересоздайте keytab, указав верные данные, включая учетную запись, к который прикреплен SPN.

2) Настройка браузера для SSO

Быстрая локальная проверка - это запуск браузера с параметрами:

```
chrome.exe --auth-server-allowlist="*.<smp.domain>"
msedge.exe --auth-server-allowlist="*.<smp.domain>"
```

Закрепить результат можно через Group Policy Management Editor (gpme.msc) перейдите в **Computer Configuration → Preferences → Windows Settings → Registry** и добавьте элемент со следующими значениями:

Параметр	Значение для Chrome	Значение для Edge
Action	Update	Update
Hive	HKEY_LOCAL_MACHINE	HKEY_LOCAL_MACHINE
Key Path	SOFTWARE\Policies\Google\Chrome	SOFTWARE\Policies\Microsoft\Edge
Value name	AuthServerAllowlist	AuthServerAllowlist
Value type	REG_SZ	REG_SZ
Value data	*.<smp.domain>	*.<smp.domain>

Также можно добавить изменение в локальный реестр:

```
reg add "HKLM\SOFTWARE\Policies\Google\Chrome" /v AuthServerAllowlist /t REG_SZ /d
"*.<smp.domain>" /f
reg add "HKLM\SOFTWARE\Policies\Microsoft\Edge" /v AuthServerAllowlist /t REG_SZ /d
"*.<smp.domain>" /f
```

Для корпоративных браузеров можно использовать ADMX шаблоны.

При изменении политики на клиентском ПК необходимо выполнить:

```
gpupdate /force
```

3) Legacy способ с настройкой системной зоны

В локальной политике или групповой политике перейти в **Computer Configuration → Administrative Templates → Windows Components → Internet Explorer → Internet Control Panel → Security Page** и для **Site to Zone Assignment List** добавить содержание ***.<smp.domain>** со значением **1**.

При изменении политики на клиентском ПК необходимо выполнить:

```
gpupdate /force
```

Способ устаревший и уже не применяется для браузера Mozilla Firefox!

??????????? ? ???????
???????

Диагностика и решение проблем

KEDR Expert on-premise: ????? ?????????? ?? ?????????? ? ?????????? ?????????

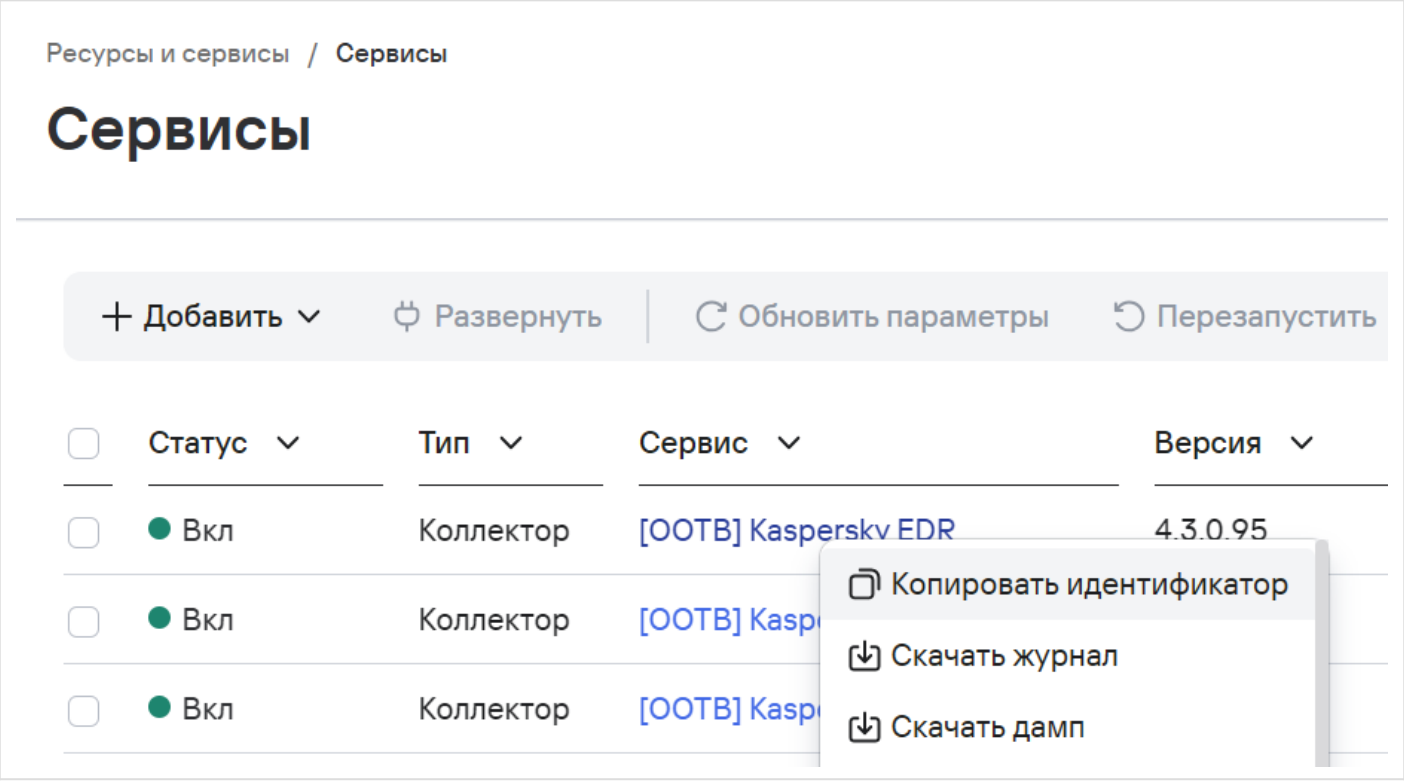
Версия продукта: KEDR Expert on-premise 8.0

Актуальный список ограничений: support.kaspersky.ru/kedr-expert-on-premise/8.0/303912

1. Проверка основных компонентов KUMA (вне Kubernetes кластере):

```
systemctl status kuma-collector-<id>.service
systemctl status kuma-correlator-<id>.service
systemctl status kuma-storage-<id>.service
```

ID компонента можно скопировать из консоли OSMP (Ресурсы и сервисы - Активные сервисы)



Работа с логами через системное журналирование:

```
journalctl -xe
journalctl -xe | grep "<component-name-with-id>"
journalctl -u "<component-name-with-id>" -e
```

Работа с логами коллекторов, корреляторов и хранилищ:

```
less /opt/kaspersky/kuma/collector/<collector_id>/log/collector
less /opt/kaspersky/kuma/correlator/<correlator_id>/log/correlator
less /opt/kaspersky/kuma/storage/<storage_id>/log/storage
tail -n 100 /opt/Kaspersky/kuma/storage/<storage_id>/log/storage
systemctl list-units | grep kuma-<name service>
```

Проверить открыт ли удаленный порт:

```
telnet <host-ip> <port>
nc -uzv <host-ip> <port>
nmap <host-ip> -p <port>
```

Проверить открытые порты:

```
firewall-cmd --list-ports
```

Открыть порт на firewall:

```
firewall-cmd --add-port=7210/tcp --permanent
firewall-cmd --reload
```

Проверка входящих подключений через tcpdump:

```
tcpdump -i any port 5144 -A
```

2. Полезные команды для работы с Kubernetes

Для Kubernetes k0s Cluster

Task	Command
View cluster info	k0s kubectl cluster-info
View nodes	k0s kubectl get nodes
View node details	k0s kubectl describe node <node-name>
Check k0s status	k0s status

Для Kubernetes k0s Pods

Task	Command
List all pods	k0s kubectl get pods -A

List all pods changes in real-time	k0s kubectl get pods -A --watch
List pods in a namespace	k0s kubectl get pods -n <namespace>
Pod details	k0s kubectl describe pod <pod-name> -n <namespace>
Delete pod	k0s kubectl delete pod <pod-name> -n <namespace>
Exec into a pod	k0s kubectl exec -it <pod-name> -n <namespace> -- /bin/sh

Для Kubernetes k0s Namespace

Task	Command
List namespaces	k0s kubectl get namespaces -A
Get all resources in namespace	k0s kubectl get all -n <namespace>
Delete namespace	k0s kubectl delete namespace <name>

Для Kubernetes k0s Secrets & ConfigMaps

Task	Command
List all secrets	k0s kubectl get secrets -A
List secrets in a namespace	k0s kubectl get secret -n <namespace>
View secret (base64 encoded)	k0s kubectl get secret <name> -n <namespace> -o yaml
Decode secret	k0s kubectl get secret -o jsonpath=”{.data.}” -n <namespace>
List configmaps	k0s kubectl get configmap
View configmap	k0s kubectl describe configmap <name>

Для Kubernetes k0s Logs

Task	Command
View logs of a pod	k0s kubectl logs <pod-name> -n <namespace>
Get logs of a pod in a real time	k0s kubectl logs -f <pod-name> -n <namespace>
Save a specific log to the file	k0s kubectl logs -f <pod-name> -n <namespace> > log.txt
Stream logs	k0s kubectl logs -f <pod-name> -n <namespace>
View logs of previous run	k0s kubectl logs --previous <pod-name> -n <namespace>

Для Kubernetes k0s Deployments & Services

Task	Command
List all deployments	k0s kubectl get deployments -A
List deployments in a namespace	k0s kubectl get deployments -n <namespace>
List services	k0s kubectl get svc
View deployment status	k0s kubectl rollout status deployment/<name>
Restart deployment	k0s kubectl rollout restart deployment <name>

Для Kubernetes k0s PVC

Task	Command
List all PV	K0s kubectl get pvc -A
List PVCs in a namespace	k0s kubectl get pvc -n <namespace>
PVC details	k0s kubectl describe pvc <pvc-name> -n <namespace>
Edit a PVC in a namespace	k0s kubectl edit pvc -n <namespace>
List all PV	K0s kubectl get pv -A
List PVCs in a namespace	k0s kubectl get pv -n <namespace>
PVC details	k0s kubectl describe pv <pvc-name> -n <namespace>

3. Получение диагностической информации о KEDR (XDR) компонентах

Утилита KDT позволяет получать диагностическую информацию о компонентах KEDR (XDR), устранять неполадки самостоятельно или с помощью службы технической поддержки

Kaspersky.
1) После установки вы можете выполнить следующую команду, чтобы просмотреть список всех установленных компонентов:

```
./kdt state
```

2) Отобразится список установленных компонентов. Правильно установленные компоненты имеют статус "Succeeded". Если установка компонента завершилась неудачно, этот компонент имеет статус "Failed".
3) Чтобы просмотреть полный журнал установки некорректно установленного компонента, выполните следующую команду:

```
./kdt state -l <имя_компонента>
```

4) Чтобы получить диагностическую информацию о компонентах и веб-плагинах управления на хосте администратора, где расположена утилита KDT, выполните следующую команду и укажите необходимый флаг:

```
./kdt logs get <флаг>
```

a. Где <флаги> - это параметры команды, которая позволяет настроить результат ведения журнала.
b. Вы можете указать следующие параметры ведения журнала:

- Период регистрации, например, все журналы за последние 12 часов:

```
./kdt logs get --to-archive --last=12h
```

- Вы можете получить диагностическую информацию за период от 2 минут до 7 дней. Если период регистрации не указан, вы получите его за максимальное время.
- Путь к целевому файлу и каталог для сохранения диагностической информации:

```
./kdt logs get -D ./path_to_folder/ --last=12h
```

5) Чтобы просмотреть доступные флаги, выполните одну из следующих команд:

```
./kdt logs get -h
./kdt logs get --help
```

6) Логи также можно загрузить непосредственно из pod'ов. Для этого выполните команды (не забудьте изменить namespace):

```
k0s kubectl get pods -A
k0s kubectl get pods -n irp
```

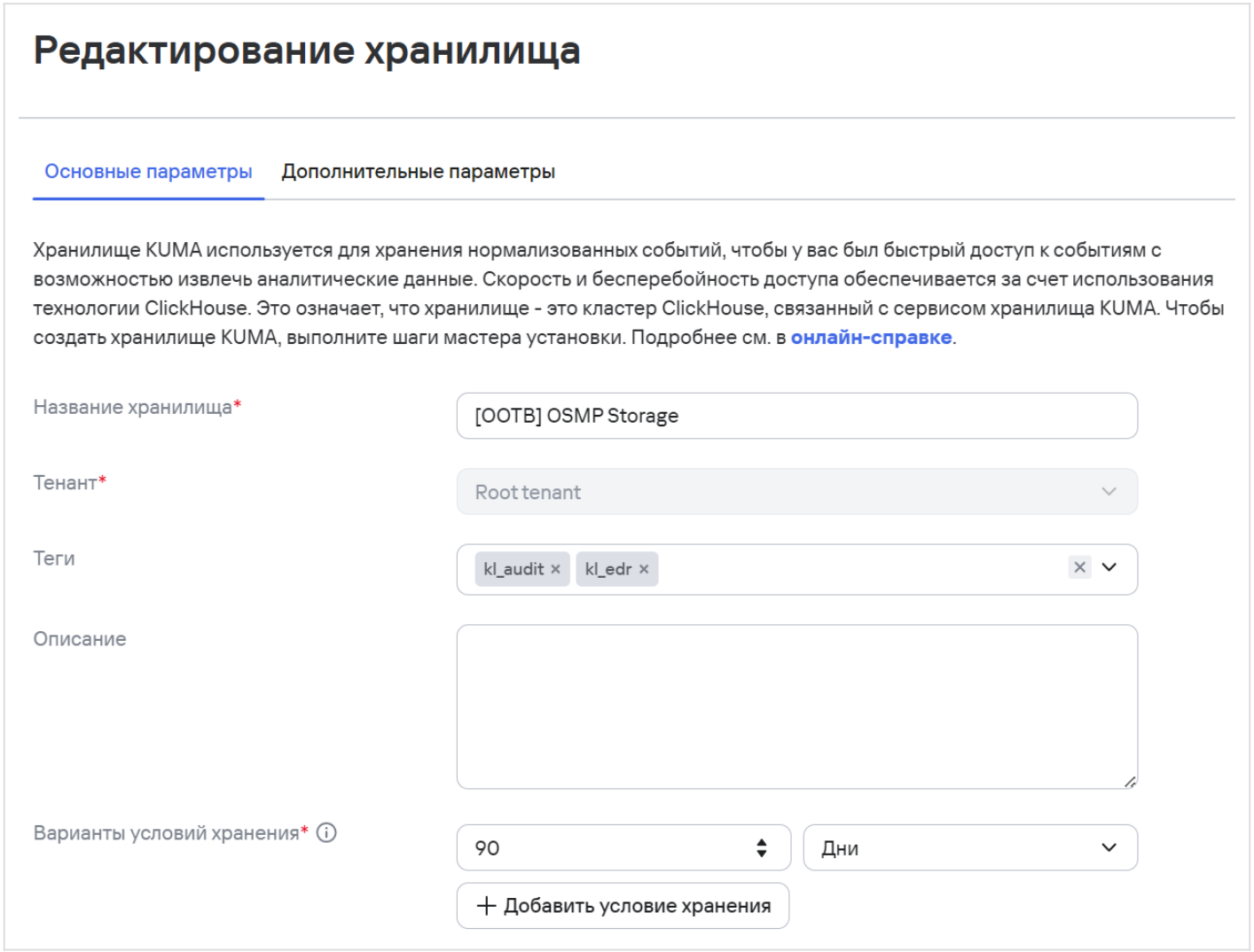
```
k0s kubectl logs interpreter-58c5856f7c-hfd87 -n irp
```

Где имя **interpreter-58c5856f7c-hfd87** должно быть изменено на уникальное имя вашего pod'a

??????? ??????? ? ?????????? (????????????? ?????)

Вариант 1: Автоматическая ротация через политику хранения

- Перейдите в **Ресурсы и сервисы - Активные сервисы**
- Нажмите на имя нужного сервиса хранилища
- Измените условие хранения:



Примечание: Изменения применяются в течение ~1 часа. Устаревшие разделы будут удалены автоматически.

Вариант 2: Ручная очистка разделов

- Перейдите в Ресурсы и сервисы - Активные сервисы

- Нажмите правой кнопкой на имя нужного сервиса хранилища
- Перейдите в просмотр разделов и вручную удалите выбранные разделы (место на диске будет увеличено сразу)