УТВЕРЖДАЮ

Генеральный директор
ООО «РДП.РУ»

_____ Никифоров Д. А.

«___» _____ 2021 г.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
EcoDPIOS-cEMS
Версия 3.2

Руководство оператора

ЛИСТ УТВЕРЖДЕНИЯ

RU.РДПТ.00012-32 34 01-ЛУ

СОГЛАСОВАНО

Исполнительный директор
ООО «РДП.РУ»

_____ Никулин С. В.

«___» _____ 2021 г.

2021

Подпись и дата	
Инв. № дубл.	
Взам. инв. №	
Подпись и дата	
Инв. № подл.	

УТВЕРЖДЁН
RU.РДПТ.00012-32 34 01-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
EcoDPIOS-cEMS
Версия 3.2

Руководство оператора

RU.РДПТ.00012-32 34 01

Листов 117

2021

Инв. № подл.	Подпись и дата	Взам. инв. №	Инв. № дубл.	Подпись и дата

АННОТАЦИЯ

В настоящем руководстве описан порядок установки и первичной настройки специализированного встраиваемого программного обеспечения EcoDPIOS-cEMS версии 3.2.

Некоторые команды и значения параметров могут отличаться в более поздних или более ранних версиях программного обеспечения. Для получения информации об актуальной версии программного обеспечения и документации обратитесь на сайт компании РДП.РУ или в службу технической поддержки. В случае обнаружения неисправности, которую не удается устранить с помощью процессов, описанных в настоящем руководстве, следует обратиться в техподдержку РДП.РУ, процедура взаимодействия, сроки ответа, форма заявки описаны в регламенте техподдержки РДП.РУ.

Указания, сопровождающиеся словами «ВНИМАНИЕ» или «ВАЖНО», обязательны для выполнения. Невыполнение данных указаний может вызвать нарушение работы оборудования и/или встроенного программного обеспечения.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение EcoDPIOS-cEMS версии 3.2 (ПО EcoDPIOS-cEMS v3.2) представляет собой специализированное встраиваемое ПО с микросервисной архитектурой для построения распределённой системы сбора и анализа статистических данных, предназначенной для использования в составе автоматизированных систем управления и мониторинга, осуществляющих функции коммутации и маршрутизации пакетов информации, глубокого анализа трафика.

Построенная на базе ПО EcoDPIOS-cEMS v3.2 система сбора и анализа статистических данных делится на две подсистемы:

- подсистема формирования предварительных списков фильтрации (ПФПС) – первый эшелон;
- подсистема централизованной обработки списков фильтрации (ПЦОС) – второй эшелон.

ПО EcoDPIOS-cEMS v3.2 является обязательной составляющей для обеспечения фильтрации трафика на программно-аппаратных комплексах (ПАК) EcoFilter и EcoBalancer.

1.1. Функциональность ПФПС

ПФПС выполняет следующие функции:

- получение журналов блокировок и журналов трафика от устройств фильтрации;
- разбор содержимого журналов блокировок и журналов трафика;
- хранение разобранной информации в локальной базе данных;
- формирование предварительных чёрных списков фильтрации;
- отправка сформированных списков в ПЦОС;
- предоставление API для мониторинга и сбора метрик.

В состав ПФПС входят следующие сервисы:

- сервис приёма первичных данных. Принимает "сырые" данные от оборудования фильтрации первого эшелона (журналы блокировок, журналы абонентских сессий) и производит синтаксический разбор полученных данных;
- сервис хранения первичных данных. Обеспечивает выполнение аналитических запросов в режиме реального времени на структурированных больших данных. Для работы с неагрегированными данными и выполнения всех необходимых вычислений используется специализированная колоночная аналитическая СУБД Яндекс ClickHouse с открытым исходным кодом;
- сервис генерации записей чёрного списка. Выполняет серии сложных многоуровневых запросов к сервису хранения первичных данных для выявления определённых поведенческих закономерностей, характерных для целевых приложений, и на основании полученных результатов создаёт предварительные чёрные списки IPv4- и IPv6-адресов;

1.2. Функциональность ПЦОС

ПЦОС выполняет следующие функции:

- получение сформированных предварительных чёрных и белых списков фильтрации;
- хранение предварительных списков в буферной базе данных;
- формирование основных белых и чёрных списков фильтрации;
- хранение основных белых и чёрных списков фильтрации;
- формирование серых списков фильтрации;
- отправка сформированных списков на оборудование балансировки;
- обеспечение доступа оборудованию фильтрации к чёрным спискам;
- предоставление API для мониторинга и сбора метрик.

В состав ПЦОС входят следующие сервисы:

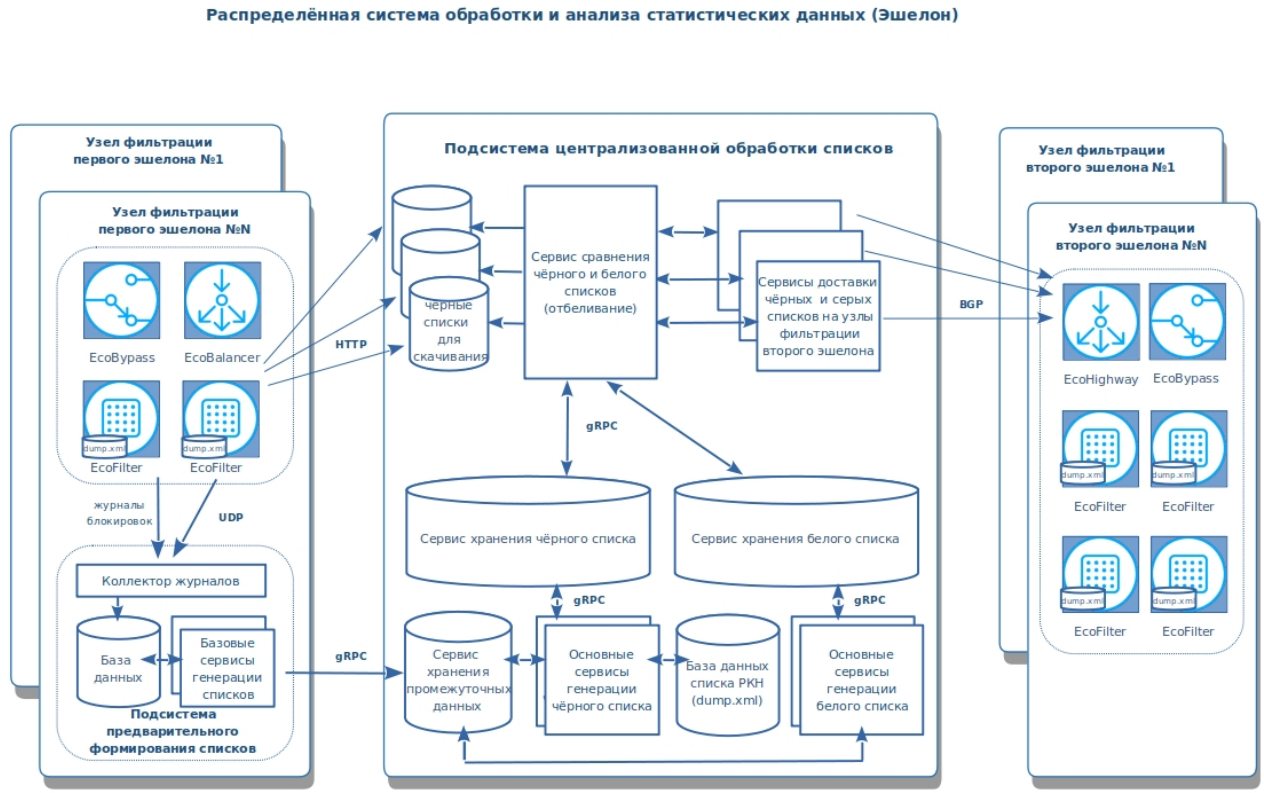
- сервис хранения промежуточных данных. Буферизирует и хранит данные из чёрных и белых списков, получаемых от ПФПС. Обеспечивает выполнение аналитических запросов в режиме реального времени на структурированных больших данных. Для хранения записей используется специализированная колоночная аналитическая СУБД Яндекс ClickHouse с открытым исходным кодом;
- сервис генерации записей чёрных списков. Формирует основные чёрные списки IPv4- и IPv6-адресов по результатам выполнения запросов к базе данных в составе сервиса хранения промежуточных данных, к базе данных списка РКН и получения готовых размеченных данных;
- сервис генерации записей белых списков. Формирует основные белые списки IPv4- и IPv6-адресов по результатам выполнения запросов к базе данных в составе сервиса хранения промежуточных данных, на основании статически предопределённого белого списка и с учётом записей, динамически добавляемых через API с указанием IP-адреса, порта, маски подсети и времени жизни;
- сервис хранения записей чёрных списков. Обеспечивает хранение основных чёрных списков IPv4- и IPv6-адресов. Для хранения данных используется документоориентированная NoSQL СУБД MongoDB с открытым исходным кодом, не требующая описания схемы таблиц, с форматом хранения данных в формате JSON.
- сервис хранения записей белых списков. Обеспечивает хранение основных белых списков IPv4- и IPv6-адресов. Для хранения данных используется документоориентированная NoSQL СУБД MongoDB с открытым исходным кодом, не требующая описания схемы таблиц, с форматом хранения данных в формате JSON.
- сервис сравнения чёрных и белых списков. Формирует группу итоговых чёрных и серых списков фильтрации по результатам сравнения основных чёрных и белых списков IPv4- и IPv6-адресов, а также позволяет загружать

категорированные чёрные списки в оборудование фильтрации первого эшелона;

- сервис управления списками фильтрации. Преобразует формат представления группы итоговых чёрных и серых списков фильтрации в формат сообщений протокола BGP;
- сервис выгрузки списков фильтрации. Обеспечивает отправку группы итоговых чёрных и серых списков IPv4- и IPv6-адресов, а также группы TCP-портов по протоколу BGP с использованием расширения BGP FlowSpec по направлению к оборудованию балансировки трафика второго эшелона;
- сервис визуализации. Обеспечивает графическое представление работы сервисов в составе системы сбора и анализа статистических данных.

1.3. Взаимодействие компонентов системы

На рисунке ниже показано взаимодействие основных компонентов системы сбора и анализа статистических данных.



2. УСЛОВИЯ ПРИМЕНЕНИЯ

Для правильной работы ПО EcoDPIOS-cEMS v3.2 должно быть установлено на серверную платформу, отвечающую приведённым ниже требованиям (из расчёта обеспечения производительности 500 Гбит/с).

Процессор	Intel® Xeon® Gold 6212U 2.4 ГГц (24 ядра)
Оперативная память	192 ГБ DDR4 2666 МГц ECC Registered
Накопитель	1 ТБ NVMe SSD
Порты служебные	не менее 1 порта 1GbE RJ45 (Management/IPMI, Console) не менее 1 порта 10G SFP+ (LOG)
Порты данных	2 порта 10GbE SFP+ (XL710)
Питание	2 блока питания постоянного или переменного тока, работающие по схеме с резервированием

3. УСТАНОВКА ПФПС

3.1. Подготовка к установке

Ниже перечислены программные инструменты, которые требуются для установки ПФПС, и настройки, которые необходимо предварительно выполнить:

- Docker версии не ниже 19.03 с отключенным docker-proxy;
- Docker Compose версии не ниже 3.7;
- для работы контейнеров может потребоваться настройка SELinux и FirewallD;
- выполнить в утилите sysctl указанные ниже настройки. Рекомендуется сохранить эти настройки в файле sysctl.conf для восстановления после перезагрузки.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
echo never > /sys/kernel/mm/transparent_hugepage/defrag
cpupower frequency-set --governor performance
```

- смонтировать раздел для хранения БД с опциями **noatime** и **nodiratime**;
- на устройствах EcoFilter в разделе конфигурации **clickstream** задать **log_format binary**. Пример настроек:

```
system.clickstream# show
enable
log_interface default
server_ip_and_port 10.10.9.43:30576
source_port 30576
mtu 1500
log_format binary
```

- во избежание фрагментации сетевых пакетов необходимо на интерфейсе, принимающем журналы, задать значение MTU 9216. Данное значение MTU необходимо задать как на логирующем интерфейсе оборудования фильтрации трафика, так и на всём промежуточном сетевом оборудовании.

3.2. Процедура установки

1. Распаковать архив **operator-docker-compose.tar** и перейти в директорию **operator-docker-compose**.
2. Загрузить образы контейнеров, отправив для каждого образа команду `docker load -i <имя образа>.tar` или команду для загрузки сразу всех образов `ls *.tar | xargs -n 1 docker load -i`.
3. Задать специфичные параметры в файле **docker-compose.yml**. Для всех параметров, которые необходимо задать, в файле **docker-compose.yml** даны описания (например, параметры SITE_NAME, ACLLISTHOST, ACLLISTPORT).
4. Поместить файлы **mem.xml** и **config.yml** в директорию, в которой находится файл **docker-compose.yml**.
5. Запустить контейнеры командой `docker-compose up -d`.
6. Проверить, что все сервисы были успешно запущены. Для этого отправить команды:

   `docker-compose ps` проверка статуса сервисов;

   `docker-compose logs -f` проверка журналов системных событий.

Процедура полной проверки работы ПФПС подробно описана в документе "Контрольные карты".

3.3. Настройка сервиса формирования предварительных чёрных списков

За формирование предварительных чёрных списков отвечает сервис **acl-creator**. Для правильной работы данного сервиса требуется задать все необходимые параметры в файле конфигурации **acl.creator.config.yml**.

Ниже показана структура файла **acl.creator.config.yml** и дано описание всех содержащихся в нём параметров.

```
clickhouse:
  host: clickhouse
  port: 9000

acllist:
  ipv4:
    host:
    port:
    secure:

  ipv6:
    host:
    port:
    secure:

timeouts:
  loop:
  error:
  maxError:

logLevel:

senders:

siteName:

defaultLifetime:

sql:
  - name:
    lifetime:
    params:
      - name:
        values:
```

```
  - name:
    lifetime:
    params:
      - name:
        values:
      - name:
        value:
```

Блок **clickhouse** содержит параметры подключения к СУБД ClickHouse, работающей в составе ПФПС. Изменений не требует.

```
clickhouse:
 host: clickhouse
 port: 9000
```

Блок **acllist** содержит параметры подключения к расположенным в ПЦОС кэширующим серверам acl-list, на которые сервис **acl-creator** будет передавать предварительные чёрные списки IPv4- и IPv6-адресов. В этом блоке следует задать IP-адреса и порты кэширующих серверов. При необходимости можно включить шифрование соединения (параметр **secure = true**; по умолчанию **false**). Пример настройки:

```
acllist:
  ipv4:
    host: 192.168.202.1
    port: 30643
    secure: false

  ipv6:
    host: 192.168.202.1
    port: 30663
    secure: false
```

В блоке **timeouts** задаются задержки в секундах между итерациями по формированию списков фильтрации:

- **loop** – задержка между успешными итерациями (периодичность обновления списков);
- **error** – задержка между итерациями, если на предыдущей возникла ошибка;
- **maxError** – максимальная задержка между итерациями в случае последовательного возникновения какой-либо ошибки.

```
timeouts:
  loop: 30s
  error: 60s
  maxError: 300s
```

Переменная **logLevel** определяет детализацию логирования событий в работе сервиса acl-creator. Ниже перечислены уровни детализации журнала событий от самого низкого к самому высокому:

- **panic** – экстренные предупреждения, требующие немедленных действий оператора;
- **fatal** – критическое состояние;
- **error** – ошибки в работе;
- **warning** – предупреждения, не требующие немедленных действий оператора;
- **info** – информационные сообщения;
- **debug** – информация для отладки кода;
- **trace** – информация для трассировки кода;

По умолчанию задан уровень детализации **info**. При выборе того или иного уровня детализации будут регистрироваться все события, соответствующие выбранному уровню, и события всех уровней ниже.

Переменная **senders** задаёт количество параллельных потоков gRPC, используемых для передачи записей на серверы acl-list.

Переменная **siteName** задаёт имя площадки, которое будет использовано при формировании записей, отправляемых серверам acl-list.

Переменная **defaultLifetime** определяет время жизни записи, используемое по умолчанию, если иное значение не задано в блоке **sql**.

Блок **sql** содержит параметры запросов. Особое внимание при изменениях следует уделить значениям переменных. Если переменная может приминать массив значений, то задаётся параметр **values**. Если переменная может принимать одно значение, то задаётся параметр **value**. Ниже дан пример настроек в блоке **sql** с описаниями переменных.

```
sql:
  - # Имя запроса
    name: gray
    # Время жизни записи для данного запроса - 30 минут
    lifetime: 30m
    # Параметры запроса
    params:
      - # Имя параметра
        name: protocols
        # Значения, передаваемые переменной.
        # Данный параметр принимает массив значений,
        # который представляет список протоколов
        values:
          - 'QUIC_MIGRATION'
          - 'INSTAGRAM'
          - 'OPERAVPN'
          - 'QUIC'
          - 'YOUTUBE'

  - # Имя запроса
    name: black
    # Время жизни записи для данного запроса – 30 минут
    lifetime: 30m
    # Параметры запроса
    params:
        # Имя параметра
      - name: protocols
        # Значения, передаваемые переменной.
        # Данный параметр принимает массив значений,
        # который представляет список протоколов
        values:
          - IPSEC
          - L2TP
          - OPENVPN_TCP
          - OPENVPN_UDP
          - PPTP
          - UTP
          - VIBER
          - VIBER_VOICE
          - VYPRVPN
          - WHATSAPP
          - WHATSAPP_VOICE
        # Имя параметра
      - name: threshold
        # Данный параметр принимает одно значение,
        # поэтому задан параметр value
        value: 60
```

4. УСТАНОВКА И НАСТРОЙКА ПЦОС

4.1. Требования к системе

Для установки ПЦОС потребуются следующие программные инструменты:

- Kubernetes версии не ниже 1.12;
- Helm версии не ниже 3;
- поддержка PV (Persistent Volume) и/или PVC (Persistent Volume Claim)
- поддержка режима ReadWrite для томов.

4.2. Описание сервисов и параметров

В таблице ниже дано описание сервисов в составе ПЦОС.

Название сервиса	**Описание**
acl-list-cache-black-v4	Сервис кэширующего чёрного списка IPv4
acl-list-cache-black-v4-db	БД кэширующего чёрного списка IPv4
acl-list-cache-black-v6	Сервис кэширующего чёрного списка IPv6
acl-list-cache-black-v6-db	БД кэширующего чёрного списка IPv6
acl-list-cache-white-v4	Сервис кэширующего белого списка IPv4
acl-list-cache-white-v4-db	БД кэширующего белого списка IPv4
acl-list-cache-white-v6	Сервис кэширующего белого списка IPv6
acl-list-cache-white-v6-db	БД кэширующего белого списка IPv6
acl-creator-from-cache-black-v4	Сервис формирования чёрного списка по данным кэширующего acl-list для IPv4
acl-creator-from-cache-black-v6	Сервис формирования чёрного списка по данным кэширующего acl-list для IPv6
acl-creator-from-cache-white-v4	Сервис формирования белого списка по данным кэширующего acl-list для IPv4
acl-creator-from-cache-white-v6	Сервис формирования белого списка по данным кэширующего acl-list для IPv6
acl-list-black-v4	Сервис чёрного списка IPv4
acl-list-black-v4-db	БД чёрного списка IPv4
acl-list-black-v6	Сервис чёрного списка IPv6
acl-list-black-v6-db	БД чёрного списка IPv6

Название сервиса	Описание
acl-list-white-v4	Сервис белого списка IPv4
acl-list-white-v4-db	БД белого списка IPv4
acl-list-white-v6	Сервис белого списка IPv6
acl-list-white-v6-db	БД белого списка IPv6
acl-list-ismon-v4	сервис синего списка IPv4
acl-list-ismon-v4-db	БД синего списка IPv4
acl-list-ismon-v6	сервис синего списка IPv6
acl-list-ismon-v6-db	БД синего списка IPv6
acl-differ-v4	Сервис для получения отбелённого чёрного списка для IPv4
acl-differ-v6	Сервис для получения отбелённого чёрного списка для IPv6
acl-inverter-v4	Сервис для получения белого списка исключений для IPv4
acl-inverter-v6	Сервис для получения белого списка исключений для IPv6
acl-manager-PROROCOL-v4	Сервис управления записями IPv4 в VRF с именем PROROCOL
acl-manager-PROROCOL-v6	Сервис управления записями IPv6 в VRF с именем PROROCOL
envoy	web-grpc gateway для acl-list UI
gobgp	goBGP
rkn-creator	Сервис обновления информации из единого реестра РКН
acl-differ-web	Сервис для скачивания списков с устройств EcoFilter

В таблице ниже описаны параметры сервисов в составе ПЦОС.

Параметр	Описание	Значение по умолчанию
acl-creator-from-cache-black-v4.config.sourceAclList.host	Доменное имя хоста, на котором доступен кэширующий чёрный список IPv4	acllist.local
acl-creator-from-cache-black-v4.config.sourceAclList.port	TCP-порт, на котором доступен кэширующий чёрный список IPv4	8181
acl-creator-from-cache-black-v4.config.acllist.host	Доменное имя хоста, на котором доступен чёрный список IPv4	acllist.local
acl-creator-from-cache-black-v4.config.acllist.port	TCP-порт, на котором доступен чёрный список IPv4	8080
acl-creator-from-cache-black-v6.config.sourceAclList.host	Доменное имя хоста, на котором доступен кэширующий чёрный список IPv6	acllist.local

Параметр	Описание	Значение по умолчанию
acl-creator-from-cache-black-v6.config.sourceAclList.port	TCP-порт, на котором доступен кэширующий чёрный список IPv6	8181
acl-creator-from-cache-black-v6.config.acllist.host	Доменное имя хоста, на котором доступен чёрный список IPv6	acllist.local
acl-creator-from-cache-black-v6.config.acllist.port	TCP-порт, на котором доступен чёрный список IPv6	8080
acl-creator-from-cache-white-v4.config.sourceAclList.host	Доменное имя хоста, на котором доступен кэширующий белый список IPv4	acllist.local
acl-creator-from-cache-white-v4.config.sourceAclList.port	TCP-порт, на котором доступен кэширующий белый список IPv4	8181
acl-creator-from-cache-white-v4.config.acllist.host	Доменное имя хоста, на котором доступен белый список IPv4	acllist.local
acl-creator-from-cache-white-v4.config.acllist.port	TCP-порт, на котором доступен белый список IPv4	8080
acl-creator-from-cache-white-v6.config.sourceAclList.host	Доменное имя хоста, на котором доступен кэширующий белый список IPv6	acllist.local
acl-creator-from-cache-white-v6.config.sourceAclList.port	TCP-порт, на котором доступен кэширующий белый список IPv6	8181
acl-creator-from-cache-white-v6.config.acllist.host	Доменное имя хоста, на котором доступен белый список IPv6	acllist.local
acl-creator-from-cache-white-v6.config.acllist.port	TCP-порт, на котором доступен белый список IPv6	8080
rkn-creator.config.dump.username	Имя пользователя для скачивания списка РКН	undefined
rkn-creator.config.dump.password	Пароль пользователя для скачивания списка РКН	undefined
rkn-creator.config.acllist4.host	Доменное имя хоста, на котором доступен чёрный список РКН IPv4	acl-list-v4.local

Параметр	Описание	Значение по умолчанию
rkn-creator.config.acllist4.port	TCP-порт, на котором доступен чёрный список РКН IPv4	30004
rkn-creator.config.acllist6.host	Доменное имя хоста, на котором доступен чёрный список РКН IPv6	acl-list-v6.local
rkn-creator.config.acllist6.port	TCP-порт, на котором доступен чёрный список РКН IPv6	30006
acl-differ-v4.web.ingress.hosts	Настройки Ingress-контроллера для доступа к спискам IPv4 с тегами 'tls' и 'hrandom', поступившим от EcoFilter	[]
acl-differ-v4.web.ingress.annotations	Аннотации для Ingress-контроллера для доступа к спискам IPv4 с тегами 'tls' и 'hrandom', поступившим от EcoFilter	{}
acl-differ-v4.whiteacllist.grpc	Доменное имя и порт для доступа к белому списку IPv4 в формате 'name:port'	list.main.white.ipv4:30641
acl-differ-v4.blackacllist.grpc	Доменное имя и порт для доступа к чёрному списку IPv4 в формате 'name:port'	list.main.black.ipv4:30642
acl-differ-v6.web.ingress.hosts	Настройки Ingress-контроллера для доступа к спискам IPv6 с тегами 'tls' и 'hrandom', поступившим от EcoFilter	[]
acl-differ-v6.web.ingress.annotations	Аннотации для Ingress-контроллера для доступа к спискам IPv6 с тегами 'tls' и 'hrandom', поступившим от EcoFilter	{}
acl-differ-v6.whiteacllist.grpc	Доменное имя и порт для доступа к белому списку IPv6 в формате 'name:port'	list.main.white.ipv6:30661
acl-differ-v6.blackacllist.grpc	Доменное имя и порт для доступа к чёрному списку IPv6 в формате 'name:port'	list.main.black.ipv6:30662

Параметр	Описание	Значение по умолчанию
acl-inverter-v4.web.ingress.hosts	Настройки Ingress-контроллера для доступа к белым спискам IPv4 с EcoFilter	[]
acl-inverter-v4.web.ingress.annotations	Аннотации для Ingress-контроллера для доступа к белым спискам IPv4 с EcoFilter	{}
acl-inverter-v4.whiteacllist.grpc	Доменное имя и порт любого списка IPv4 в формате name:port (по факту не используется)	list.main.white.ipv4:30641
acl-inverter-v4.blackacllist.grpc	Доменное имя и порт белого списка IPv4 (для инвертирования) в формате name:port	list.main.white.ipv4:30641
acl-inverter-v6.web.ingress.hosts	Настройки Ingress-контроллера для доступа к белым спискам IPv6 с EcoFilter	[]
acl-inverter-v6.web.ingress.annotations	Аннотации для Ingress-контроллера для доступа к белым спискам IPv6 с EcoFilter	{}
acl-inverter-v6.whiteacllist.grpc	Доменное имя и порт любого списка IPv6 в формате name:port (по факту не используется)	list.main.white.ipv6:30661
acl-inverter-v6.blackacllist.grpc	Доменное имя и порт белого списка IPv6 (для инвертирования) в формате name:port	list.main.white.ipv6:30661
gobgp.bgp.routerID	Идентификатор маршрутизатора для экземпляра goBGP	nil
gobgp.bgp.neighbors	Список соседей экземпляра goBGP	[]
gobgp.nodeSelector	Узел, за которым будет закреплён экземпляр goBGP	{}
acl-manager-PROROCOL-v4.config.loopTimeout	Периодичность формирования списка (в секундах)	30

Параметр	Описание	Значение по умолчанию
acl-manager-PROROCOL-v4.config.prefix.addr	Доменное имя и порт acl-differ-v4 в формате name:port	acl-differ-v4:30940
acl-manager-PROROCOL-v4.config.goBGP.addr	Доменное имя и порт gRPC gobgp в формате name:port	bgp.host:50051
acl-manager-PROROCOL-v6.config.loopTimeout	Периодичность формирования списка (в секундах)	30
acl-manager-PROROCOL-v6.config.prefix.addr	Доменное имя и порт acl-differ-v6 в формате name:port	acl-differ-v6:30960
acl-manager-PROROCOL-v6.config.goBGP.addr	Доменное имя и порт gRPC gobgp в формате name:port	bgp.host:50051
acl-list-black-v4.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-black-v4.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-black-v4.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-black-v4.mongodb.auth.username	Имя пользователя для доступа к БД чёрного списка IPv4	пустое значение
acl-list-black-v4.mongodb.auth.password	Пароль для доступа к БД чёрного списка IPv4	пустое значение
acl-list-black-v4.mongodb.auth.rootPassword	Пароль для доступа к БД чёрного списка IPv4 с правами 'root'	пустое значение
acl-list-black-v4.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true
acl-list-black-v4.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-black-v4.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""

Параметр	Описание	Значение по умолчанию
acl-list-black-v4.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-black-v4.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД чёрного списка IPv4	пустое значение
acl-list-black-v4.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД чёрного списка IPv4	пустое значение
acl-list-black-v4.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение
acl-list-black-v4.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние чёрного списка	[]
acl-list-black-v4.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-black-v4.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/black/v4
acl-list-black-v6.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-black-v6.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-black-v6.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-black-v6.mongodb.auth.username	Имя пользователя для доступа к БД чёрного списка IPv6	пустое значение
acl-list-black-v6.mongodb.auth.password	Пароль для доступа к БД чёрного списка IPv6	пустое значение
acl-list-black-v6.mongodb.auth.rootPassword	Пароль для доступа к БД чёрного списка IPv6 с правами 'root'	пустое значение
acl-list-black-v6.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true

Параметр	Описание	Значение по умолчанию
acl-list-black-v6.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-black-v6.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-black-v6.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-black-v6.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД чёрного списка IPv6	пустое значение
acl-list-black-v6.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД чёрного списка IPv6	пустое значение
acl-list-black-v6.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение
acl-list-black-v6.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние чёрного списка	[]
acl-list-black-v6.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-black-v6.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/black/v6
acl-list-white-v4.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-white-v4.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-white-v4.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-white-v4.mongodb.auth.username	Имя пользователя для доступа к БД белого списка IPv4	пустое значение

Параметр	Описание	Значение по умолчанию
acl-list-white-v4.mongodb.auth.password	Пароль для доступа к БД белого списка IPv4	пустое значение
acl-list-white-v4.mongodb.auth.rootPassword	Пароль для доступа к БД белого списка IPv4 с правами 'root'	пустое значение
acl-list-white-v4.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true
acl-list-white-v4.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-white-v4.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-white-v4.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-white-v4.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД белого списка IPv4	пустое значение
acl-list-white-v4.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД белого списка IPv4	пустое значение
acl-list-white-v4.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение
acl-list-white-v4.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние белого списка	[]
acl-list-white-v4.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-white-v4.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/white/v4
acl-list-white-v6.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-white-v6.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}

Параметр	Описание	Значение по умолчанию
acl-list-white-v6.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-white-v6.mongodb.auth.username	Имя пользователя для доступа к БД белого списка IPv6	пустое значение
acl-list-white-v6.mongodb.auth.password	Пароль для доступа к БД белого списка IPv6	пустое значение
acl-list-white-v6.mongodb.auth.rootPassword	Пароль для доступа к БД белого списка IPv6 с правами 'root'	пустое значение
acl-list-white-v6.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true
acl-list-white-v6.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-white-v6.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-white-v6.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-white-v6.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД белого списка IPv6	пустое значение
acl-list-white-v6.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД белого списка IPv6	пустое значение
acl-list-white-v6.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение
acl-list-white-v6.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние белого списка	[]
acl-list-white-v6.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-white-v6.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/white/v4

Параметр	Описание	Значение по умолчанию
acl-list-ismon-v4.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-ismon-v4.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-ismon-v4.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-ismon-v4.mongodb.auth.username	Имя пользователя для доступа к БД синего списка IPv4	пустое значение
acl-list-ismon-v4.mongodb.auth.password	Пароль для доступа к БД синего списка IPv4	пустое значение
acl-list-ismon-v4.mongodb.auth.rootPassword	Пароль для доступа к БД синего списка IPv4 с правами 'root'	пустое значение
acl-list-ismon-v4.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true
acl-list-ismon-v4.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-ismon-v4.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-ismon-v4.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-ismon-v4.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД синего списка IPv4	пустое значение
acl-list-ismon-v4.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД синего списка IPv4	пустое значение
acl-list-ismon-v4.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение

Параметр	Описание	Значение по умолчанию
acl-list-ismon-v4.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние синего списка	[]
acl-list-ismon-v4.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-ismon-v4.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/ismon/v4
acl-list-ismon-v6.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-ismon-v6.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-ismon-v6.mongodb.enabled	true – запускать базу mongodb false – использовать внешнюю базу	true
acl-list-ismon-v6.mongodb.auth.username	Имя пользователя для доступа к БД синего списка IPv6	пустое значение
acl-list-ismon-v6.mongodb.auth.password	Пароль для доступа к БД синего списка IPv6	пустое значение
acl-list-ismon-v6.mongodb.auth.rootPassword	Пароль для доступа к БД синего списка IPv6 с правами 'root'	пустое значение
acl-list-ismon-v6.mongodb.persistence.enabled	Использовать PVC для постоянного хранилища	true
acl-list-ismon-v6.mongodb.persistence.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-ismon-v6.mongodb.persistence.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-ismon-v6.mongodb.persistence.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi

Параметр	Описание	Значение по умолчанию
acl-list-ismon-v6.externalDatabase.mongodbUsername	Имя пользователя для доступа к внешней БД синего списка IPv6	пустое значение
acl-list-ismon-v6.externalDatabase.mongodbPassword	Пароль для доступа к внешней БД синего списка IPv6	пустое значение
acl-list-ismon-v6.externalDatabase.mongodbHost	Адрес внешней БД	пустое значение
acl-list-ismon-v6.ui.ingress.hosts	Настройки Ingress-контроллера для доступа к UI, позволяющему просматривать текущее состояние синего списка	[]
acl-list-ismon-v6.ui.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-list-ismon-v6.ui.grpc.proxy	Ссылка на прокси-сервер web-grpc	http://cluster.domain.com:30273/acl/ismon/v4
acl-list-cache-black-v4.clickhouse.enabled	true – запускать базу clickhouse false – использовать внешнюю базу	true
acl-list-cache-black-v4.clickhouse.storage.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-cache-black-v4.clickhouse.storage.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-cache-black-v4.clickhouse.storage.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-cache-black-v4.externalDatabase.clickhouseHost	Адрес БД при использовании внешней БД	пустое значение
acl-list-cache-black-v4.externalDatabase.clickhouseDatabase	Имя БД при использовании внешней БД	acllist
acl-list-cache-black-v4.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]

Параметр	Описание	Значение по умолчанию
acl-list-cache-black-v4.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-cache-black-v6.clickhouse.enabled	true – запускать базу clickhouse false – использовать внешнюю базу	true
acl-list-cache-black-v6.clickhouse.storage.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-cache-black-v6.clickhouse.storage.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-cache-black-v6.clickhouse.storage.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-cache-black-v6.externalDatabase.clickhouseHost	Адрес БД при использовании внешней БД	пустое значение
acl-list-cache-black-v6.externalDatabase.clickhouseDatabase	Имя БД при использовании внешней БД	acllist
acl-list-cache-black-v6.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-cache-black-v6.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-cache-white-v4.clickhouse.enabled	true – запускать базу clickhouse false – использовать внешнюю базу	true
acl-list-cache-white-v4.clickhouse.storage.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-cache-white-v4.clickhouse.storage.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-cache-white-v4.clickhouse.storage.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi

Параметр	Описание	Значение по умолчанию
acl-list-cache-white-v4.externalDatabase.clickhouseHost	Адрес БД при использовании внешней БД	пустое значение
acl-list-cache-white-v4.externalDatabase.clickhouseDatabase	Имя БД при использовании внешней БД	acllist
acl-list-cache-white-v4.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-cache-white-v4.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
acl-list-cache-white-v6.clickhouse.enabled	true – запускать базу clickhouse false – использовать внешнюю базу	true
acl-list-cache-white-v6.clickhouse.storage.class	Имя класса для динамического выделения тома для постоянного хранилища	пустое значение
acl-list-cache-white-v6.clickhouse.storage.existingClaim	Имя существующего тома для постоянного хранилища	""
acl-list-cache-white-v6.clickhouse.storage.size	Размер динамически выделяемого тома для постоянного хранилища	0Gi
acl-list-cache-white-v6.externalDatabase.clickhouseHost	Адрес БД при использовании внешней БД	пустое значение
acl-list-cache-white-v6.externalDatabase.clickhouseDatabase	Имя БД при использовании внешней БД	acllist
acl-list-cache-white-v6.ingress.hosts	Настройки Ingress-контроллера для доступа по gRPC	[]
acl-list-cache-white-v6.ingress.annotations	Аннотации для Ingress-контроллера для доступа по gRPC	{}
envoy.templates.envoy.yaml	Конфигурация сервиса envoy	nil
acl-differ-web.ingress.hosts	Настройки Ingress-контроллера для скачивания списков с EcoFilter	[]

Параметр	Описание	Значение по умолчанию
acl-differ-web.ingress.annotations	Настройка аннотаций для Ingress-контроллера	{}
acl-differ-web.config	Конфигурация сервиса для настройки запросов и источников данных	nil
acl-differ-web.config.cacheLifetime	Периодичность формирования списка в формате "2h45m". Возможные единицы измерения "ns", "us", "ms", "s", "m", "h"	1m

4.3. Конфигурация envoy.templates.envoy.yaml

Для того чтобы UI мог взаимодействовать с сервисами **acl-list**, требуется указать действительные хосты и порты, на которых развёрнуты данные сервисы.

Конфигурация задаётся согласно шаблону ниже, в котором необходимо раскомментировать и переопределить **ip.address.of.cluster**.

```
envoy:
  templates:
    envoy.yaml: |-
      admin:
        access_log_path: /dev/stdout
        address:
          socket_address:
            address: 0.0.0.0
            port_value: {{ .Values.ports.admin.containerPort }}

      static_resources:
        listeners:
        - name: listener_0
          address:
            socket_address:
              address: 0.0.0.0
              port_value: {{ .Values.ports.n0.containerPort }}
          filter_chains:
          - filters:
            - name: envoy.http_connection_manager
              config:
                codec_type: auto
                stat_prefix: ingress_http
                route_config:
                  name: local_route
                  virtual_hosts:
```

```
                  - name: local_service
                    domains: ["*"]
                    routes:
                    - match: { prefix: "/acl/black/v4/" }
                      route:
                        cluster: acl_list_black_v4
                        max_grpc_timeout: 0s
                        prefix_rewrite: "/"
                    - match: { prefix: "/acl/white/v4/" }
                      route:
                        cluster: acl_list_white_v4
                        max_grpc_timeout: 0s
                        prefix_rewrite: "/"
                    - match: { prefix: "/acl/black/v6/" }
                      route:
                        cluster: acl_list_black_v6
                        max_grpc_timeout: 0s
                        prefix_rewrite: "/"
                    - match: { prefix: "/acl/white/v6/" }
                      route:
                        cluster: acl_list_white_v6
                        max_grpc_timeout: 0s
                        prefix_rewrite: "/"
                    cors:
                      allow_origin:
                      - "*"
                      allow_methods: GET, PUT, DELETE, POST, OPTIONS
                      allow_headers: keep-alive,user-agent,cache-
control,content-type,content-transfer-encoding,custom-header-1,x-accept-content-
transfer-encoding,x-accept-response-streaming,x-user-agent,x-grpc-web,grpc-
timeout
                      max_age: "1728000"
                      expose_headers: custom-header-1,grpc-status,grpc-message
                http_filters:
                - name: envoy.grpc_web
                - name: envoy.cors
                - name: envoy.router
        clusters:
        - name: acl_list_black_v4
          connect_timeout: 0.25s
          type: logical_dns
          http2_protocol_options: {}
          lb_policy: round_robin
          hosts: [{ socket_address: { address: ip.address.of.cluster,
port_value: 30642 }}]
        - name: acl_list_white_v4
          connect_timeout: 0.25s
          type: logical_dns
          http2_protocol_options: {}
          lb_policy: round_robin
          hosts: [{ socket_address: { address: ip.address.of.cluster,
port_value: 30641 }}]
        - name: acl_list_black_v6
          connect_timeout: 0.25s
          type: logical_dns
          http2_protocol_options: {}
          lb_policy: round_robin
          hosts: [{ socket_address: { address: ip.address.of.cluster,
port_value: 30662 }}]
```

```
        - name: acl_list_white_v6
          connect_timeout: 0.25s
          type: logical_dns
          http2_protocol_options: {}
          lb_policy: round_robin
          hosts: [{ socket_address: { address: ip.address.of.cluster,
port_value: 30661 }}]
```

4.4. Настройка MongoDB

Со всеми настройками MongoDB можно ознакомиться в официальном репозитории по ссылке https://github.com/bitnami/charts/tree/master/bitnami/mongodb/

Минимально необходимые настройки – параметры **mongodbUsername**, **mongodbPassword**, **mongodbRootPassword**, а также параметры хранилища **persistence**.

Параметры **nameOverride**, **mongodbDatabase** и **metrics** предопределены в данном чарте для всех экземпляров.

4.5. Настройка acl-list.ui

Для единого чёрного или белого списка должны быть определены секции **ingress**. Ниже приведён пример секции настроек. Вместо **cluster.domain.ru** следует указать действительное доменное имя для доступа к UI. Необходимо также указать **NodePort** для **envoy**. В примере указан порт 30273. На практике после развёртывания **envoy** следует обновить конфигурацию и указать действительное значение. Кроме того, требуется задать пути к чёрному и белому спискам. В путях должны быть указаны тип списка (black или white) и версия IP (v4 или v6).

```
ui:
  ingress:
    hosts:
      - host: cluster.domain.ru
        paths: ['/acl/black/v4/(.*)']
    annotations:
      kubernetes.io/ingress.class: 'nginx'
      nginx.ingress.kubernetes.io/enable-rewrite-log: 'true'
      nginx.ingress.kubernetes.io/rewrite-target: '/$1'

  grpc:
    proxy: http://cluster.domain.ru:30273/acl/black/v4
```

4.6. Настройка gobgp.bgp.neighbors

В секции конфигурации **gobgp.bgp.neighbors** указывается список соседей для экземпляра goBGP. Ниже приведён пример заполнения данной секции.

```
neighbors:
  - name: server1
    ip: 127.0.0.2
    as: 64500
  - name: server2
    ip: 127.0.0.3
    as: 64500
```

4.7. Настройка сервиса acl-differ-web для формирования необходимых итоговых списков фильтрации

Настройка формирования итоговых списков фильтрации, передаваемых на оборудование второго эшелона, производится в файле конфигурации **config.yaml** для сервиса **acl-differ-web**. Ниже представлен шаблон конфигурации и даны пояснения по его заполнению.

```
cacheLifetime:

sources:
  black: []

  white: {}

queries:
  name:
    source:
    params:
      black:
        include: []
        exclude: []
      white:
        include: []
        exclude: []
```

Переменная **cacheLifetime** задаёт периодичность формирования списков. Допустимые единицы измерения: ns, us, ms, s, m, h. Можно задавать в смешанном формате. Например, 1h30m.

В блоке **sources** необходимо указать источники для загрузки чёрного (**black**) и белого (**white**) списков в формате **<IP-адрес или доменное имя сервера:номер порта>**. Параметры **black** и **white** могут принимать массив значений, т. е. можно указать несколько серверов.

В блоке **queries** необходимо указать, какие списки следует загружать и какие записи должны быть включены в указанные списки. Описание параметров:

- **name** – вместо слова **name** необходимо указать имя загружаемого списка. Допустимые имена указаны в таблице ниже в столбцах "Список html" и "Отбеливающий список html". Для загрузки нескольких списков требуется создать в блоке **queries** отдельную секцию **name** для каждого списка и задать в ней все необходимые параметры (см. пример конфигурации ниже).
- **source** – источник для загрузки. Допустимые значения: **black** и **white**, т. е. серверы, указанные в блоке **sources**;
- **include** и **exclude** – эти параметры определяют, какие записи должны быть, соответственно, включены в загружаемый список и исключены из него. В этих параметрах необходимо указать имена тегов из таблицы ниже (столбцы "Чёрные/серые теги" и "Белые теги"). Параметры могут принимать массив значений.

Чёрные / серые теги	Белые теги	Сигнатура на фильтре	Список bgp	Список html	Отбеливающий список html	Protocol ID
	static	-	-	-	static_wh	
hrandom	signature, pattern, hrandom_static	telegram_ex	telegram_bb	telegram_bh	telegram_wh	0
tls	tls_static	block_cnt, tls_packet	faketls_bb	faketls_bh	faketls_wh	13
whatsapp	whatsapp_static	whatsapp_ex	whatsapp_bb	whatsapp_bh	whatsapp_wh	1
viber	viber_static	viber_ex	viber_bb	viber_bh	viber_wh	2
whatsapp_voice	whatsapp_voice_static	whatsapp_voice_ex	whatsapp_voice_bb	whatsapp_voice_bh	whatsapp_voice_wh	4
viber_voice	viber_voice_static	viber_voice_ex	viber_voice_bb	viber_voice_bh	viber_voice_wh	3
utp	utp_static	utp_ex	utp_bb	utp_bh	utp_wh	11
ipsec	ipsec_static	ipsec_ex	ipsec_bb	ipsec_bh	ipsec_wh	6
l2tp	l2tp_static	l2tp_ex	l2tp_bb	l2tp_bh	l2tp_wh	7
pptp	pptp_static	pptp_ex	pptp_bb	pptp_bh	pptp_wh	8
openvpn_udp	openvpn_udp_static	openvpn_udp_ex	openvpn_udp_bb	openvpn_udp_bh	openvpn_udp_wh	9
openvpn_tcp	openvpn_tcp_static	openvpn_tcp_ex	openvpn_tcp_bb	openvpn_tcp_bh	openvpn_tcp_wh	12
vyprvpn	vyprvpn_static	vyprvpn_ex	vyprvpn_bb	vyprvpn_bh	vyprvpn_wh	5
operavpn	operavpn_static	operavpn_ex	operavpn_gb	-	operavpn_wh	14
youtube	youtube_static	youtube_ex	youtube_gb	-	youtube_wh	15

Чёрные / серые теги	Белые теги	Сигнатура на фильтре	Список bgp	Список html	Отбеливающий список html	Protocol ID
facebook	facebook_static	facebook_ex	facebook_gb	-	facebook_wh	16
instagram	instagram_static	instagram_ex	instagram_gb	-	instagram_wh	17
quic	quic_static	quic_list ( )	quic_gb	-	quic_wh	10
		-	rkn_port_gb	-	-	65000
rkn		-	rkn_ip_bb	-	-	65001
ismon		-	ismon_cb	-	-	65002

Обозначения

ex – сигнатура на фильтре

bb – чёрный/серый список для балансировщика

gb – серый список для балансировщика

cb – список клиентских подсетей для фильтрации на балансировщике

bh – чёрный список для фильтра

wh – отбеливающий список для фильтра

Формулы для наполнения списков

[Список bgp] = [все чёрные теги] - [все белые теги] - static

[Список html] = [все чёрные теги] - [все белые теги] - static

[Отбеливающий html] = [все белые теги]

Списки bgp загружаются в балансировщик, и уже на нём оператор выбирает, какие списки будут задействованы.

Списки html загружаются в фильтр. При загрузке указывается, для каких протоколов необходимо загрузить списки.

Ниже приведён пример конфигурации сервиса acl-differ-web для загрузки нескольких списков фильтрации:

```
cacheLifetime: 1m

sources:
  black:
    - server1.myblacklist.ru:30940
    - server2.myblacklist.ru:30960

  white:
    - server3.mywhitelist.ru:30941
    - server4.mywhitelist.ru:30961

queries:
  telegram_bh:
    source: black
    params:
      black:
        include:
          - hrandom
        exclude:
          - rkn
```

```
    white:
      include:
        - hrandom_static
        - static
      exclude: []

telegram_wh:
  source: white
  params:
    black:
      include:
        - hrandom_static
        - signature
        - pattern
      exclude: []
    white:
      include:
        - nooneofrealtags
      exclude: []

whatsapp_bh:
  source: black
  params:
    black:
      include:
        - whatsapp
      exclude:
        - rkn
    white:
      include:
        - whatsapp_static
        - static
      exclude: []

whatsapp_wh:
  source: white
  params:
    black:
      include:
        - whatsapp_static
      exclude: []
    white:
      include:
        - nooneofrealtags
      exclude: []
```

4.8. Процедура установки

1. Распаковать архив **core-services-release.tar** и перейти в каталог **core-services-release**.
2. Загрузить образы контейнеров, отправив для каждого образа команду `docker load -i <имя образа>.tar` или команду для загрузки сразу всех образов `ls *.tar | xargs -n 1 docker load -i`.
3. Задать все необходимые параметры в файле **values.yml** (см. раздел 4.2). Пример минимально необходимой конфигурации приведён ниже.
4. Настроить сервис **acl-differ-web** (см. раздел 4.7).
5. При использовании NodePort заменить внутренние доменные имена **cluster.local** для сервисов **acl-list**.
6. Задать параметры хранилища для MongoDB и Clickhouse.
7. Указать имена пользователей и пароли для MongoDB и Clickhouse
8. Задать логин и пароль учётной записи для загрузки Единого реестра запрещённых ресурсов РКН.
9. Установить программное обеспечение командой `helm install --namespace asbi --create-namespace echelon-core . -f values.yaml`.
10. Проверить, что все сервисы были успешно запущены. Для этого отправить команды просмотра статуса сервисов и логов:

    ```
    helm status -n asbi echelon-core
    ```

    ```
    kubectl -n asbi get deployments
    ```

    ```
    kubectl -n asbi get daemonsets
    ```

    ```
    kubectl -n asbi get statefulsets
    ```

11. По документу "Контрольная карта" проверить, что все компоненты системы EcoDPIOS-cEMS успешно запущены.

Минимально необходимая конфигурация

Ниже приведён пример минимально необходимой конфигурации в файле **values.yaml**.

```
acl-creator-from-cache-black-v4:
  config:
    acllist:
      host: list.main.black.ipv4.cluster.example.org
    sourceAclList:
      host: list.cache.black.ipv4.cluster.example.org
acl-creator-from-cache-black-v6:
  config:
    acllist:
      host: list.main.black.ipv6.cluster.example.org
    sourceAclList:
      host: list.cache.black.ipv6.cluster.example.org
acl-creator-from-cache-white-v4:
  config:
    acllist:
      host: list.main.white.ipv4.cluster.example.org
    sourceAclList:
      host: list.cache.white.ipv4.cluster.example.org
acl-creator-from-cache-white-v6:
  config:
    acllist:
      host: list.main.white.ipv6.cluster.example.org
    sourceAclList:
      host: list.cache.white.ipv6.cluster.example.org
acl-differ-v4:
  blackacllist:
    grpc: list.main.black.ipv4.cluster.example.org:30642
  web:
    ingress:
      hosts:
        - host: differ.v6.echelon.external.example.org
          paths:
            - /
  whiteacllist:
    grpc: list.main.white.ipv4.cluster.example.org:30641
acl-differ-v6:
  blackacllist:
    grpc: list.main.black.ipv6.cluster.example.org:30662
  web:
    hosts:
      - host: differ.v4.echelon.external.example.org
        paths:
          - /
  whiteacllist:
    grpc: list.main.white.ipv6.cluster.example.org:30661
acl-list-black-v4:
  mongodb:
    mongodbPassword: password
    mongodbRootPassword: rootpass
    mongodbUsername: admin
    persistence:
      storageClass: 'default'
      size: 3Gi
  ui:
    grpc:
      proxy: http://echelon.external.example.org:30273/acl/black/v4
    ingress:
      hosts:
        - host: acl.list.black.v4.echelon.external.example.org
```

```
          paths:
            - /acl/black/v4/(.*)
acl-list-black-v6:
  mongodb:
    mongodbPassword: password
    mongodbRootPassword: rootpass
    mongodbUsername: admin
    persistence:
      storageClass: 'default'
      size: 3Gi
  ui:
    grpc:
      proxy: http://echelon.external.example.org:30273/acl/black/v6
    ingress:
      hosts:
        - host: acl.list.black.v6.echelon.external.example.org
          paths:
            - /acl/black/v6/(.*)
acl-list-cache-black-v4:
  clickhouse:
    storage:
      class: 'default'
      size: 3Gi
acl-list-cache-black-v6:
  clickhouse:
    storage:
      class: 'default'
      size: 3Gi
acl-list-cache-white-v4:
  clickhouse:
    storage:
      class: 'default'
      size: 3Gi
acl-list-cache-white-v6:
  clickhouse:
    storage:
      class: 'default'
      size: 3Gi
acl-list-white-v4:
  mongodb:
    mongodbPassword: password
    mongodbRootPassword: rootpass
    mongodbUsername: admin
    persistence:
      storageClass: 'default'
      size: 3Gi
  ui:
    grpc:
      proxy: http://echelon.external.example.org:30273/acl/white/v4
    ingress:
      hosts:
        - host: acl.list.white.v4.echelon.external.example.org
          paths:
            - /acl/white/v4/(.*)
acl-list-white-v6:
  mongodb:
    mongodbPassword: password
    mongodbRootPassword: rootpass
    mongodbUsername: admin
```

```
    persistence:
      storageClass: 'default'
      size: 3Gi
  ui:
    grpc:
      proxy: http://echelon.external.example.org:30273/acl/white/v6
    ingress:
      hosts:
        - host: acl.list.white.v6.echelon.external.example.org
          paths:
            - /acl/white/v6/(.*)
rkn-creator:
  config:
    acllist4:
      host: list.main.black.ipv4.cluster.example.org
    acllist6:
      host: list.main.black.ipv6.cluster.example.org
```

4.9. Шифрование соединения между ПФПС и ПЦОС

По умолчанию все запросы и ответы между сервисами ПФПС и ПЦОС передаются по незашифрованным соединениям. Однако в системе EcoDPIOS-cEMS предусмотрена возможность TLS-шифрования. Включение данной возможности производится четырьмя простыми действиями:

1. В конфигурации сервиса **acl-creator** необходимо в блоке **acllist** для ipv4 и ipv6 присвоить параметру **secure** значение **true**.
2. В файле **values.yaml** для сервиса **acl-list** присвоить параметру **tls** значение **true**.
3. В файле **values.yaml** для сервиса **acl-list-cache** присвоить параметру **tls** значение **true**.
4. Перезапустить вышеуказанные сервисы, чтобы новые настройки вступили в силу.

Сертификаты TLS загружаются в контейнеры соответствующих сервисов.

5. ОПИСАНИЕ API

В системе EcoDPIOS-cEMS реализован gRPC API, который позволяет работать с записями ACL (добавление, обновление, удаление), выполнять поиск записей по заданным критериям, просматривать историю добавления записей и журналы отладки. Разделы данной главы содержат отдельные описания работы с API для каждой из вышеперечисленных задач. Следует помнить, что синтаксис запросов и ответов зависит от используемого gRPC-клиента (см. справочные материалы к gRPC-клиенту), поэтому даны только общие описания запросов, ответов и их параметров в виде таблиц. Для удобства навигации по разделам предусмотрены перекрёстные ссылки.

5.1. API для работы с записями ACL

API для добавления, обновления, удаления и поиска записей ACL реализуется через файл **protobuf/acl_list.proto**. В таблицах ниже дано описание доступных методов и формат запросов и ответов.

Метод	**Запрос**	**Ответ**	**Действие**
InsertOrUpdate	AclItem	AclListResult	Добавление или обновление одной записи
InsertOrUpdateStream	AclItem (потоковый)	AclListResult	Добавление или обновление нескольких записей
Delete	AclItem	AclListResult	Удаление записи
List	AclListSearchParams	AclItemWithTags (потоковый)	Поиск записей по заданным критериям
DeleteExpiredItems	DeleteExpiredItemsParams	AclListResult	Удаление устаревших записей

AclItem

Запись в ACL.

Поле	Тип данных	Метка	Описание
IP	NetIP		IP-адрес
Port	uint32		TCP-порт, который необходимо фильтровать. '0' означает все порты
Tag	AclItemTag		Информация о теге
site	SiteInfo		Информация о площадке

AclItemTag

Тег, связанный с записью.

Поле	Тип данных	Метка	Описание
Name	string		Имя тега
Description	string		Дополнительная информация
ExpiredAt	google.protobuf.Timestamp		Временная метка, после которой данный тег станет неактуальным

AclItemWithTags

ACL-запись со всеми тегами.

Поле	Тип данных	Метка	Описание
IP	NetIP		IP-адрес
Port	uint32		TCP-порт, который необходимо фильтровать. '0' означает все порты
Tags	AclItemTag	repeated	Все теги, связанные с указанным IP-адресом и портом

AclListResult

Результат запроса на добавление, обновление или удаление записей.

Поле	Тип данных	Метка	Описание
code	AclListResult.ResultCodes		Код результата
message	string		Сообщение об ошибке

AclListSearchParams

Критерии поиска записей в ACL.

Поле	Тип данных	Метка	Описание
NoExpired	bool		Выводить только актуальные записи или все записи. True – выводить только актуальные записи, т. е. хотя бы у одного тега значение ExpiredAt должно быть больше, чем текущее время. False – выводить все записи.
IP	string		IP-адрес или его часть. Поиск выполняется по подстроке от начала к концу. Если указать 1.2.3. (с точкой в конце), то будут найдены все адреса, начинающиеся с 1.2.3. (например, 1.2.3.4, 1.2.3.40 и т. п.). Если указать 1.2.3 (без точки в конце), то, например, будет найден как адрес 1.2.3.4, так и адрес 1.2.30.4. Если IP-адрес не указан, то будут выведены все записи, для которых выполняются остальные критерии запроса

Поле	Тип данных	Метка	Описание
Port	uint32		TCP-порт. 0 означает все порты. При значении отличном от 0 поиск будет выполнен по конкретному значению
Pagination	AclListSearchParams.PaginationParams		Параметры постраничного вывода
IncludeTags	string	repeated	Запись должна содержать указанные теги. Если NoExpired = true, то хотя бы один из указанных тегов должен быть актуален
ExcludeTags	string	repeated	Запись не должна содержать указанные теги независимо от значения NoExpired

AclListSearchParams.PaginationParams

Параметры постраничного вывода. Основное назначение – для UI.

Поле	Тип данных	Метка	Описание
PageNo	uint32		Номер страницы
PerPage	uint32		Количество элементов на одной странице

DeleteExpiredItemsParams

Параметры удаления устаревших записей.

Поле	Тип данных	Метка	Описание
ItemAgeOverSpecifiedHours	uint32		Записи должны быть старше указанного количества часов

NetIP

Представление IP-адреса для формирования записи.

Поле	Тип данных	Метка	Описание
IP	string		IP-адрес в формате IPv4 или IPv6
Mask	uint32		Маска, соответствующая адресу

SiteInfo

Информация о площадке, с которой поступила запись.

Поле	Тип данных	Метка	Описание
Name	string		Имя площадки

AclListResult.ResultCodes

Коды результатов запроса на добавление, обновление или удаление записей.

Результат	Код	Пояснение
OK	0	Успешно
FAIL	1	Ошибка

5.2. API для просмотра истории добавления записей в ACL

API для просмотра истории добавления записей в ACL реализуется через файл **history/history.proto**. В таблицах ниже дано описание доступных методов и формат запросов и ответов.

Метод	Запрос	Ответ	Действие
IPHistory	IPHistoryParams	IPHistoryRecord (потоковый)	Вывод истории добавления записей по заданным параметрам
SiteTagStats	SiteTagStatsParams	SiteTagStatRecord (потоковый)	Вывод статистики добавления записей в разрезе тегов и площадок

IPHistoryParams

Параметры запроса истории добавления записей

Поле	Тип данных	Метка	Описание
ip	string		IP-адрес или его часть. Поиск выполняется по подстроке от начала к концу. Если указать 1.2.3. (с точкой в конце), то будут найдены все адреса, начинающиеся с 1.2.3. (например, 1.2.3.4, 1.2.3.40 и т. п.). Если указать 1.2.3 (без точки в конце), то, например, будет найден как адрес 1.2.3.4, так и адрес 1.2.30.4
timerange	TimeRange		Временной диапазон. Если не указан, то будут выведены результаты за всё время с момента начала работы системы

IPHistoryRecord

Содержимое ответа на запрос истории добавления записей.

Поле	Тип данных	Метка	Описание
ip_with_mask_and_port	string		IP-адрес в формате [ip/mask]:port
tags_history	TagHistoryRecord	repeated	Записи о тегах, связанных с данным IP-адресом

SiteTagStatRecord

Статистика добавления записей в разрезе тегов и площадок.

Поле	Тип данных	Метка	Описание
insert_time	google.protobuf.Timestamp		Время добавления записи
site_name	string		Название площадки
tag_name	string		Имя тега
count	uint32		Количество добавленных записей

SiteTagStatsParams

Параметры выборки статистики добавления записей в разрезе тегов и площадок.

Поле	Тип данных	Метка	Описание
timerange	TimeRange		Временной диапазон. Если не указан, то будут выведены результаты за всё время с момента начала работы системы

TagHistoryRecord

Информация о теге.

Поле	Тип данных	Метка	Описание
site_name	string		Название площадки
tag_name	string		Имя тега у записи
expired_tag_at	google.protobuf.Timestamp		Время, до которого запись актуальна
insert_time	google.protobuf.Timestamp		Время добавления записи

TimeRange

Диапазон времени.

Поле	Тип данных	Метка	Описание
from	google.protobuf.Timestamp		Начало диапазона
to	google.protobuf.Timestamp		Конец диапазона

5.3. API для запроса данных из журналов сессий и блокировок

API для работы с журналами сессий и блокировок реализуется через файл **protobuf/log_proxy.proto**. В таблицах ниже дано описание доступных методов и формат запросов и ответов.

Метод	Запрос	Ответ	Действие
ListAccounting	RequestParams	AccountingResult (потоковый)	Вывод информации об установленных сессиях
ListClickStream	RequestParams	ClickStreamResult (потоковый)	Вывод информации о сессиях HTTP/HTTPS
ListProtocol	RequestParams	ProtocolResult (потоковый)	Вывод информации о распознанных протоколах
ListShortList	RequestParams	ShortListResult (потоковый)	Вывод журнала блокировок по единому реестру РКН и по IP-адресу/URL
ListDebugLog	RequestParams	DebugLogResult (потоковый)	Вывод журнала блокировок по сигнатурам (с тегом **random**)

AccountingResult

Информация об установленных сессиях.

Поле	Тип данных	Метка	Описание
local_ip	string		IP-адрес клиента в формате IPv4 или IPv6
local_port	uint32		Порт клиента
remote_ip	string		IP-адрес удалённого ресурса в формате IPv4 или IPv6
remote_port	uint32		Порт удалённого ресурса
protocol	L4ProtocolCode		Протокол транспортного уровня: TCP или UDP
direction	DirectionCode		Направление сессии: входящая – сессия инициирована из сети Интернет (WAN) в сторону абонента (LAN); исходящая – сессия инициирована абонентом (LAN) в сторону сети Интернет (WAN)
start_time	google.protobuf.Timestamp		Время начала сессии
end_time	google.protobuf.Timestamp		Время завершения сессии
out_bytes	uint32		Количество исходящих байтов
in_bytes	uint32		Количество входящих байтов
out_packets	uint32		Количество исходящих пакетов

Поле	Тип данных	Метка	Описание
in_packets	uint32		Количество входящих пакетов
filter_id	string		Уникальный идентификатор фильтра

ClickStreamResult

Информация о сессиях HTTP/HTTPS.

Поле	Тип данных	Метка	Описание
start_time	google.protobuf.Timestamp		Время запроса
content	string		Для HTTP: содержимое HTTP запроса
sni	string		Для HTTPS: доменное имя
filter_id	string		Уникальный идентификатор фильтра

DebugLogResult

Информация о блокировках по сигнатурам.

Поле	Тип данных	Метка	Описание
protocol	L4ProtocolCode		Протокол транспортного уровня: TCP или UDP
local_ip	string		IP-адрес клиента в формате IPv4 или IPv6
local_port	uint32		Порт клиента
remote_ip	string		IP-адрес удалённого ресурса в формате IPv4 или IPv6
remote_port	uint32		Порт удалённого ресурса
dpi_protocol_code	uint32		Тип распознанного протокола в соответствии со встроенной сигнатурой (список ID приведён в документации на EcoFilter)

Поле	Тип данных	Метка	Описание
direction	DirectionCode		Направление пакета: входящий – пакет направлен из сети Интернет (WAN) в сторону абонента (LAN); исходящий – пакет направлен от абонента (LAN) в сторону сети Интернет (WAN)
content	bytes		Содержимое распознанного Ethernet-пакета (до 256 байт)
pkt_time	google.protobuf.Timestamp		Время записи информации о пакете в БД
filter_id	string		Уникальный идентификатор фильтра

ProtocolResult

Информация о распознанных протоколах

Поле	Тип данных	Метка	Описание
local_ip	string		IP-адрес клиента в формате IPv4 или IPv6
local_port	uint32		Порт клиента
remote_ip	string		IP-адрес удалённого ресурса в формате IPv4 или IPv6
remote_port	uint32		Порт удалённого ресурса
start_time	google.protobuf.Timestamp		Время начала сессии
end_time	google.protobuf.Timestamp		Время завершения сессии
dpi_protocol_code	uint32	repeated	Список распознанных протоколов (список ID приведён в документации на EcoFilter)
filter_id	string		Уникальный идентификатор фильтра

RequestParams

Параметры запроса.

Поле	Тип данных	Метка	Описание
start_from	google.protobuf.Timestamp		Запрос записей журнала, начиная с указанного момента времени
sampling_rate	uint32		Коэффициент сэмплирования журналов
follow	bool		Ожидать появления новых записей и отправлять их клиенту

ShortListResult

Журнал блокировок по Единому реестру запрещённых ресурсов Роскомнадзора.

Поле	Тип данных	Метка	Описание
protocol	DPIListProtocolCode		Блокированный протокол: HTTP, HTTPS или IP
url	string		URL для протокола HTTP
sni	string		SNI для протокола HTTPS
remote_ip	string		IP-адрес удалённого ресурса в формате IPv4 или IPv6
remote_port	uint32		Порт удалённого ресурса
local_ip	string		IP-адрес клиента в формате IPv4 или IPv6
local_port	uint32		Порт клиента
dpilist	uint32		Идентификатор списка фильтрации, согласно которому была произведена блокировка
start_time	google.protobuf.Timestamp		Время запроса
filter_id	string		Уникальный идентификатор фильтра

DPIListProtocolCode

Коды заблокированных протоколов.

Обозначение	Код	Описание
DPI_LIST_PROTOCOL_CODE_UNKNOWN	0	Не определён
DPI_LIST_PROTOCOL_CODE_HTTP	1	HTTP
DPI_LIST_PROTOCOL_CODE_HTTPS	2	HTTPS
DPI_LIST_PROTOCOL_CODE_IP	3	IP

DirectionCode

Коды направлений сессий и пакетов.

Обозначение	Код	Описание
DIRECTION_CODE_UNKNOWN	0	Не определено
DIRECTION_CODE_EGRESS	1	Исходящее (LAN → WAN)
DIRECTION_CODE_INGRESS	2	Входящее (WAN → LAN)

L4ProtocolCode

Коды протоколов транспортного уровня.

Обозначение	Код	Описание
L4_PROTOCOL_CODE_UNKNOWN	0	
L4_PROTOCOL_CODE_TCP	1	TCP
L4_PROTOCOL_CODE_UDP	2	UDP

5.4. Описание скалярных типов данных

В таблице ниже дано описание скалярных типов данных, передаваемых в запросах и ответах для различных языков реализации API.

Тип данных protobuf	Примечания	C++	Java	Python	Go	C#	PHP	Ruby
double		double	double	float	float64	double	float	Float
float		float	float	float	float32	float	float	Float
int32	Использует кодировку переменной длины. Неэффективен для кодирования отрицательных чисел. Если в вашем случае поле может содержать как положительные, так и отрицательные значения, то следует использовать **sint32**.	int32	int	int	int32	int	integer	Bignum или Fixnum (в зависимости от конкретных требований)
int64	Использует кодировку переменной длины. Неэффективен для кодирования отрицательных чисел. Если в вашем случае поле может содержать как положительные, так и отрицательные значения, то следует использовать **sint64**.	int64	long	int/long	int64	long	integer/string	Bignum
uint32	Использует кодировку переменной длины.	uint32	int	int/long	uint32	uint	integer	Bignum или Fixnum (в зависимости от конкретных требований)
uint64	Использует кодировку переменной длины.	uint64	long	int/long	uint64	ulong	integer/string	Bignum или Fixnum (в зависимости от конкретных требований)
sint32	Знаковое целое число. Использует кодировку переменной длины. По сравнению с int32 более эффективен для кодирования отрицательных чисел.	int32	int	int	int32	int	integer	Bignum или Fixnum (в зависимости от конкретных требований)
sint64	Знаковое целое число. Использует кодировку переменной длины. По сравнению с int64 более эффективен для кодирования отрицательных чисел.	int64	long	int/long	int64	long	integer/string	Bignum

Тип данных protobuf	Примечания	C++	Java	Python	Go	C#	PHP	Ruby
fixed32	Префикс длины – всегда 4 байта. Более эффективен, чем uint32, если значения обычно больше 2^28.	uint32	int	int	uint32	uint	integer	Bignum или Fixnum (в зависимости от конкретных требований)
fixed64	Префикс длины – всегда 8 байт. Более эффективен, чем uint64, если значения обычно больше 2^56.	uint64	long	int/long	uint64	ulong	integer/string	Bignum
sfixed32	Префикс длины – всегда 4 байта.	int32	int	int	int32	int	integer	Bignum или Fixnum (в зависимости от конкретных требований)
sfixed64	Префикс длины – всегда 8 байт.	int64	long	int/long	int64	long	integer/string	Bignum
bool		bool	boolean	boolean	bool	bool	boolean	TrueClass/FalseClass
string	Строка должна содержать текст в кодировке UTF-8 или 7-bit ASCII.	string	String	str/unicode	string	string	string	String (UTF-8)
bytes	Может содержать любую произвольную последовательность байтов.	string	ByteString	str	[]byte	ByteString	string	String (ASCII-8BIT)

6. НАСТРОЙКА СРЕДСТВ МОНИТОРИНГА

В системе EcoDPIOS-cEMS предусмотрена возможность комплексного мониторинга работы как системы в целом, так и её отдельных сервисов. Данная возможность реализуется с помощью связки приложений Grafana и Prometheus, которые способны в режиме реального времени предоставлять в графическом и текстовом виде подробную информацию обо всех аспектах работы EcoDPIOS-cEMS.

В следующих разделах данной главы описана процедура настройки взаимодействия приложений Grafana и Prometheus с основными сервисами EcoDPIOS-cEMS.

6.1. Настройка взаимодействия с сервисами ПФПС

6.1.1. acl-creator

Данный сервис предоставляет статистику создания предварительных списков. Шаблон настроек – файл **acl-creator.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-creator** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-creator
  ip_type: v4
  rule_type: white
  log_type: random
  playground: test
```

- **rule_type** – может принимать значения **white** или **black**;
- **ip_type** – может принимать значения **v4** или **v6**;
- **log_type** – берётся из названия контейнера. Например, **tls** для hub.rdp.ru/acl-creator-black-tls-v4:v2.3.3;
- **playground** – название площадки, указанное в **docker-compose.yaml**.

6.1.2. drop-log-reader

Данный сервис предоставляет статистику приёма логов.

Шаблоны настроек – файлы **drop-log-reader.json** (для каждого сервиса) и **drop-log-reader-playground-aggregation.json** (общий по всем сервисам), которые находятся в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **drop-log-reader** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: drop-log-reader
  playground: test
```

- **playground** – название площадки, указанное в **docker-compose.yaml**;
- **service** – возможные значения: drop-log-reader, accounting-log-reader, clickstream-log-reader, shortlist-log-reader, proto-log-reader.

6.1.3. log-proxy-reader

Данный сервис предоставляет статистику передачи логов в ЦСУО.

Шаблон настроек – файл **log-proxy-reader.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **log-proxy-reader** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: log-proxy-reader
  playground: test
```

- **playground** – название площадки, указанное в **docker-compose.yaml**.

6.1.4. clickhouse

Данный сервис предоставляет статистику приёма логов.

Шаблон настроек – файл **clickhouse-self-monitoring.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **clickhouse** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: clickhouse
  type: drop-log-reader
  playground: test
```

- **playground** – название площадки, указанное в **docker-compose.yaml**.

6.2. Настройка взаимодействия с сервисами ПЦОС

6.2.1. rkn-creator

Сервис обновления информации из реестра РКН.

Шаблон настроек – файл **rkn-creator.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **rkn-creator** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels
  service: rkn-creator
  rule_type: black
  description: 'get ipv4 and ipv6 from dump.xml'
  playground: test
```

- **rule_type** – может принимать значения **white** или **black**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.2. acl-list

Сервис для работы со списками.

Шаблон настроек – файл **acl-list.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-list** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-list
  rule_type: black
  ip_type: v4
  db: mongodb
  playground: test
```

- **rule_type** – может принимать значения **white**, **black** или **ismon**;
- **ip_type** – может принимать значения **v4** или **v6**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.3. acl-differ

Сервис для получения финального списка.

Шаблон настроек – файл **acl-differ.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-differ** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-differ
  ip_type: v4
  playground: test
```

- **ip_type** – может принимать значения **v4** или **v6**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.4. acl-list-db

База данных со списком.

Шаблон настроек – файл **mongodb.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-list-db** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-list-db
  ip_type: v4
  rule_type: black
  playground: test
```

- **rule_type** – может принимать значения **white** или **black**;
- **ip_type** – может принимать значения **v4** или **v6**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.5. acl-manager

Сервис управления записями в VRF.

Шаблон настроек – файл **acl-manager.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-manager** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-manager
  ip_type: v4
  type: block
  playground: test
```

- **ip_type** – может принимать значения **v4** или **v6**;

- **type** – берётся из названия deployment. Например, **block** для **scos-prod-acl-manager-rknip-v6.scos-prod**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.6. acl-creator-from-cache

Формирование списка по данным кэширующего acl-list.

Шаблон настроек – файл **acl-creator.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-creator-from-cache** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-creator
  rule_type: black
  ip_type: v4
  tag: cache
  playground: test
```

- **rule_type** – может принимать значения **white** или **black**;
- **ip_type** – может принимать значения **v4** или **v6**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.7. acl-list-cache

Сервис кэширующего списка.

Шаблон настроек – файл **acl-list-cache.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **acl-list-cache** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: acl-list
  rule_type: black
  ip_type: v4
  db: clickhouse
  playground: test
```

- **rule_type** – может принимать значения **white** или **black**;
- **ip_type** – может принимать значения **v4** или **v6**;
- **playground** – название площадки или идентификатор ЦОД.

6.2.8. gobgp

Сервис передачи списков на балансировщики второго эшелона.

Шаблон настроек – файл **gobgp.json** в архиве **grafana.tar.gz**.

Для взаимодействия с сервисом **gobgp** необходимо, чтобы приложение Prometheus проставило следующие теги:

```
labels:
  service: gobgp
  playground: test
```

- **playground** – название площадки или идентификатор ЦОД.

7. УСТРАНЕНИЕ ТИПОВЫХ ПРОБЛЕМ

В этой главе рассмотрены типовые проблемы, которые могут возникнуть в процессе работы системы EcoDPIOS-cEMS, и даны указания по выявлению и устранению их причин.

7.1. В коллектор не поступают записи журналов блокировок от оборудования фильтрации

За приём и обработку записей журналов блокировок от оборудования EcoFilter в коллекторе журналов отвечает сервис приёма первичных данных.

7.1.1. Описание проблемы

После настройки оборудования EcoFilter и запуска сервиса приёма первичных данных значения счётчиков для метрик сервиса не отвечают критериям исправной работы, а именно:

- значения метрик сервиса не приходят по запросу;

```
operator@operator-pc# curl 192.168.91.50:2113/metrics
curl: (7) Failed to connect to 192.168.91.50 port 2112: В соединении отказано
```

- значения метрик сервиса равны нулю или не изменяются в течение продолжительного времени.

```
operator@operator-pc# curl -s 192.168.91.50:2112/metrics | grep -E
"reader_received_packets|writer_inserted_packets"
# HELP reader_received_packets The total number of received UDP packets from
EcoFilters
# TYPE reader_received_packets counter
reader_received_packets 0
# HELP writer_inserted_packets The total number of inserted packets in DB
# TYPE writer_inserted_packets counter
writer_inserted_packets{query="insert_histogram_log"} 0
writer_inserted_packets{query="insert_incorrect_log"} 0
writer_inserted_packets{query="insert_random_log"} 0
```

Описание метрик

- **reader_received_packets** – метрика, счётчик которой показывает количество записей журналов блокировок, поступивших в коллектор от оборудования EcoFilter. Счётчик является накопительным.

При правильной настройке оборудования EcoFilter и правильной работе сервиса приёма первичных данных значение счётчика должно быть больше 0.

- **writer_inserted_packets** – набор метрик, счетчики которых показывают количество записей, которые были распознаны коллектором и успешно отправлены в сервис хранения первичных данных. Для каждого типа распознанных записей предусмотрен отдельный счётчик метрики. Счётчики являются накопительными.

При правильной работе сервисов приёма и хранения первичных данных значения счётчиков должны быть больше 0.

7.1.2. Возможные причины неисправной работы

Причинами возникновения вышеописанной проблемы могут быть:

- неправильная настройка раздела **debug_logger** в конфигурации EcoFilter;
- проблема с сетевой связностью между EcoFilter и сервисом коллектора;
- несоответствующая версия прошивки EcoFilter;
- нерабочее состояние сервисов коллектора журналов блокировок и базы данных;
- неправильная конфигурация сервисов коллектора журналов;
- несоответствующая версия образа для контейнера с сервисом коллектора журналов.

7.1.3. Порядок выявления и устранения причин

1. Проверить настройки **debug_logger** в конфигурации EcoFilter. Убедиться, что отправка сообщений включена и указаны правильные значения параметра **protocols**, а также IP-адрес и порт коллектора. При необходимости внести изменения в конфигурацию оборудования EcoFilter.
2. Проверить актуальность версии прошивки. При необходимости обновить программное обеспечение оборудования EcoFilter.

```
2:# show version
EcoNAT generic v2.1 (C) Ecotelecom [RDP.RU Ltd.] 2013-2019. All rights reserved.
Firmware version: 3.1.4.0.40.bp
S/N: 1C87764018C3
```

3. Проверить журнал событий для контейнера с сервисом коллектора журналов командой **`docker logs имя_сервиса`**

```
operator@operator-pc# docker logs collector
time="2020-04-27T16:18:49Z" level=info msg="Check for migrations" func="drop-
log-reader/db.(*ClickHouse).Migrate" file="/go/src/drop-log-
reader/db/clickhouse.go:89"
time="2020-04-27T16:18:49Z" level=info msg="No migrations. Current version 13 is
up to date" func="drop-log-reader/db.(*ClickHouse).Migrate" file="/go/src/drop-
log-reader/db/clickhouse.go:100"
server listening [::]:555
```

или командой **`docker-compose logs имя_сервиса`**

```
docker-compose logs collector
Attaching to spfs_collector_1
collector_1         | time="2020-04-27T16:18:49Z" level=info msg="Check for
migrations" func="drop-log-reader/db.(*ClickHouse).Migrate" file="/go/src/drop-
log-reader/db/clickhouse.go:89"
collector_1         | time="2020-04-27T16:18:49Z" level=info msg="No migrations.
Current version 13 is up to date" func="drop-log-
reader/db.(*ClickHouse).Migrate" file="/go/src/drop-log-
reader/db/clickhouse.go:100"
collector_1         | server listening [::]:555
```

В журнале событий контейнера с сервисом коллектора журналов не должно быть систематических ошибок.

Для получения дополнительной информации из журналов при неправильном поведении сервиса использовать команду **`docker logs имя_сервиса --details`**.

```
goroutine 1 [running]:
github.com/jmoiron/sqlx.MustConnect(...)
      /go/pkg/mod/github.com/jmoiron/sqlx@v1.2.0/sqlx.go:650
collector/db.(*ClickHouse).Open(0xc00013c400)
      /go/src/collector/db/clickhouse.go:57 +0x149
main.(*App).prepareDB(0xc0001182a0)
      /go/src/collector/main.go:92 +0x43
main.(*App).start(0xc0001182a0)
      /go/src/collector/main.go:68 +0x2f
main.main()
      /go/src/collector/main.go:46 +0x109
panic: dial tcp: lookup clickhouse-server on 127.0.0.11:53: no such host
```

4. Проверить, что сервисы коллектора журналов и базы данных находятся в рабочем состоянии (STATUS -> Up). Для этого следует отправить команды **`docker ps`** и **`docker-compose ps`** (данную команду необходимо отправлять из директории, в которой находится файл **docker-compose.yaml**).

```
operator@operator-pc# docker ps
CONTAINER ID IMAGE        COMMAND          CREATED        STATUS      PORTS               NAMES
51250b4a6e30 clickhouse   "/entrypoint.sh" 9 minute ago   Up 9 minute 123/tcp,9000/tcp,   clickhouse
                                                                       9009/tcp
08c984509f21 collector:1.0 "/collector"    13 minutes ago Up 9 minute 192.168.91.50:30555 collector
                                                                       ->555/udp
                                                                       192.168.91.50:2112
                                                                       ->2112/tcp
operator@operator-pc# docker-compose ps
         Name                   Command     State   Ports
------------------------------------------------------------------------------
spfs clickhouse 1         /entrypoint.sh    Up      8123/tcp, 9000/tcp, 9009/tcp
spfs collector 1          /collector        UP      192.168.91.50:2112->2112/tcp,
                                                    192.168.91.50:3055>555/udp
```

5. Проверить параметры конфигурации сервиса коллектора журнала блокировок, применяемые при запуске сервиса, с помощью команды **`cat docker-compose.yaml`** (данную команду необходимо отправлять из директории, в которой находится файл **docker-compose.yaml**).

```
operator@operator-pc# cat docker-compose.yml
version: '3.7'
services:

  collector:
    image: collector:1.0
    environment:
      - LISTENIP=0.0.0.0
      - LISTENPORT=555
      - PARSERSCOUNT=4
      - CLICKHOUSEHOST=clickhouse
      - CLICKHOUSEPORT=9000
  clickhouse:
    image: clickhouse
```

Текущую конфигурацию в работающем контейнере с сервисом коллектора можно проверить командой **`docker inspect -f "{{json .Config.Env}}" имя_сервиса | jq`**.

```
operator@operator-pc# docker inspect -f "{{json .Config.Env}}" collector | jq
[
  "LISTENIP=0.0.0.0",
  "LISTENPORT=555",
  "PARSERSCOUNT=4",
  "CLICKHOUSEHOST=clickhouse-server",
  "CLICKHOUSEPORT=9000",
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
]
```

В настройках сервиса коллектора журнала блокировок должны быть указаны правильные значения следующих параметров:

- **LISTENIP** и **LISTENPORT**. Эти параметры указывают, на каком внутреннем IP-адресе и внутреннем UDP-порту сервис принимает данные от EcoFilter. Достаточно значений по умолчанию: 0.0.0.0 и 555.

- **CLICKHOUSEHOST** и **CLICKHOUSEPORT**. Эти параметры указывают сервису коллектора журнала блокировок DNS-имя и порт сервиса базы данных. DNS-имя должно в точности совпадать с именем сервиса базы данных, указанным в файле **docker-compose.yaml**. В противном случае сервис коллектора журнала блокировок не сможет правильно работать.

  При необходимости внести соответствующие правки в конфигурационный файл **docker-compose.yaml** и исправить поведение сервиса командой **`docker-compose up -d`**.

6. Проверить командами **`docker ps`** и **`docker-compose ps`**, что сервис коллектора журнала блокировок имеет правильную конфигурацию: IP-адрес и порт в разделе **ports** соответствуют настройкам раздела **debug_logger** в конфигурации оборудования EcoFilter:

```
operator@operator-pc# docker ps
CONTAINER ID IMAGE        COMMAND          CREATED        STATUS      PORTS                 NAMES
51250b4a6e30 clickhouse   "/entrypoint.sh" 9 minute ago   Up 9 minute 123/tcp,9000/tcp,   clickhouse
                                                                        9009/tcp
08c984509f21 collector:1.0 "/collector"    13 minutes ago Up 9 minute 192.168.91.50:30555 collector
                                                                        ->555/udp
                                                                       192.168.91.50:2112
```

```
                                                                    ->2112/tcp
operator@operator-pc# docker-compose ps
         Name                   Command     State   Ports
------------------------------------------------------------------------------
spfs clickhouse 1         /entrypoint.sh    Up      8123/tcp, 9000/tcp, 9009/tcp
spfs collector 1          /collector        UP      192.168.91.50:2112->2112/tcp,
                                                    192.168.91.50:3055>555/udp
```

При необходимости внести соответствующие правки в конфигурационный файл **docker-compose.yaml** и обновить развёрнутый сервис командой **`docker-compose up -d`**.

7. Проверить, что сервис коллектора журнала блокировок проброшен на внешний порт сервера. Для этого следует отправить команду **`netstat -tulpn | grep -E "30555|2112"`**.

```
operator@operator-pc# netstat -tulpn  | grep -E  "30555|2112"
tcp    0    0 192.168.91.50:2112    0.0.0.0:*      LISTEN      25129/docker-proxy
udp    0    0 192.168.91.50:30555   0.0.0.0:*                  25116/docker-proxy
```

Проброс портов необходим для того, чтобы трафик, посылаемый EcoFilter, после получения сервером на UDP-порт (30555) мог быть перенаправлен в сервис коллектора журнала блокировок. Кроме основного порта для приёма трафика логов может быть проброшен TCP-порт (2112), используемый для доступа к метрикам сервиса коллектора журнала блокировок в целях диагностики. Для обслуживания этих портов должна использоваться служба **docker-proxy**.

8. Проверить версию образа сервиса командой **`docker inspect -f "{{json .Config.Image}}" имя_сервиса | jq`**.

```
docker inspect -f "{{json .Config.Image}}" collector | jq
"collector:1.0"
```

или командой **`docker ps | grep имя_сервиса`**

```
operator@operator-pc# docker ps | grep collector
CONTAINER ID IMAGE         COMMAND         CREATED        STATUS      PORTS                NAMES
51250b4a6e30 clickhouse    "/entrypoint.sh" 9 minute ago   Up 9 minute 123/tcp, 9000/tcp,   clickhouse
                                                                        9009/tcp
08c984509f21 collector:1.0 "/collector"    13 minutes ago Up 9 minute 192.168.91.50:30555 collector
                                                                        ->555/udp
                                                                       192.168.91.50:2112
                                                                        ->2112/tcp
```

При необходимости следует скачать новый образ из репозитория (команды **`docker pull`** или **`docker-compose pull`**), внести соответствующие

правки в конфигурационный файл **docker-compose.yaml** и обновить развёрнутые сервисы командой **`docker-compose up -d`**.

9. Проверить сетевую связность между сервисом коллектора журнала блокировок и оборудованием EcoFilter. Например, командой **`docker exec -it имя_сервиса sh`** зайти в оболочку сервиса коллектора журнала логов и проверить доступность управляющего интерфейса EcoFilter с помощью утилиты **ping**.

```
/srv # ping 10.86.4.101 -c 4
PING 10.86.4.101 (10.86.4.101): 56 data bytes
64 bytes from 10.86.4.101: seq=0 ttl=62 time=0.375 ms
64 bytes from 10.86.4.101: seq=1 ttl=62 time=0.438 ms
64 bytes from 10.86.4.101: seq=2 ttl=62 time=0.467 ms
64 bytes from 10.86.4.101: seq=3 ttl=62 time=0.486 ms

--- 10.86.4.101 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.375/0.441/0.486 ms
```

Устранить проблему сетевой связности при её наличии.

10. Проверить получение трафика логов от оборудования EcoFilter на сервер на системном уровне и на уровне сервиса коллектора журнала блокировок с использованием общедоступных системных утилит **tcpdump**, **iftop**, **nload**. Подключение к оболочке сервиса выполняется также с помощью команды **`docker exec -it имя_сервиса sh`**.

```
/ srv # tcpdump -ni eth0 udp port 30555 -c 10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
12:22:25.217505 IP 10.86.4.192.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217510 IP 10.86.4.192.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217515 IP 10.86.4.192.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217543 IP 10.86.4.192.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217525 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217530 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217534 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217539 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217543 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
12:22:25.217548 IP 10.86.4.193.1088 > 192.168.91.50.30555: UDP, length 78
10 packets captured
11 packets received by filter
0 packets dropped by kernel
```

При необходимости откорректировать работу межсетевого экрана на системном уровне.

7.2. Не работают базовые сервисы генерации списка и сервисы промежуточного хранения данных

Базовые сервисы генерации списка ПФПС отвечают за формирование записей из данных, полученных сервисом коллектора журнала блокировок. Сгенерированные записи должны быть отправлены в сервис хранения промежуточных данных, работающий в составе ПЦОС.

В свою очередь, задачей сервиса хранения промежуточных данных в ПЦОС является хранение записей, полученных от базовых сервисов генерации списков в составе ПФПС.

7.2.1. Описание проблемы

После правильной настройки и запуска оборудования EcoFilter, сервисов коллектора журнала блокировок и базы данных, показатели метрик базовых сервисов генерации списка или сервисов хранения промежуточных данных не соответствуют критериям успешной работы, а именно:

- значения счётчиков для метрик базовых сервисов генерации списка не приходят по запросу:

```
operator@operator-pc# curl 192.168.91.50:2113/metrics
curl: (7) Failed to connect to 192.168.91.50 port 2113: В соединении отказано
```

- значения счётчиков для метрик базовых сервисов генерации списка в течение продолжительного времени остаются равными нулю, однако при этом записи журнала от EcoFilter поступают в сервис коллектора журнала блокировок.

```
operator@operator-pc# operator@operator-pc# curl -s 192.168.91.50:2113/metrics|grep -E
"acl creator items sended via grpc|acl creator items from source"
# HELP acl creator items from source Number of records returned from source in last query
# TYPE acl creator items from source gauge
acl creator items from source 0
# HELP acl_creator_items_sended_via_grpc Number of records sended via grpc
# TYPE acl_creator_items_sended_via_grpc gauge
acl_creator_items_sended_via_grpc 0
```

- значения счётчиков для метрик сервисов хранения промежуточных данных не приходят по запросу:

```
operator@operator-pc# kubectl port-forward scos-acl-list-cache-black-v4 --namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep acl list items in last list
Handling connection for 5000
E0429 15:45:55.670279 24686 portforward.go:400] an error occurred forwarding 5000 -> 2112: error
forwarding port 2112 to pod d5f6b9c8c1e8f64646886f04bf696ab770ce45b4c7b7f117f4068d0c44a90ac5, uid :
exit status 1: 2020/04/29 12:45:55 socat[27411] E connect(5, AF=2 127.0.0.1:2112, 16): Connection
refused
```

- значения счётчиков для метрик сервисов хранения промежуточных данных в течение длительного промежутка времени остаются равными нулю, однако при этом:
  - записи журнала от EcoFilter поступают в сервис коллектора журнала блокировок,
  - базовые сервисы генерации списка получают записи из сервиса базы данных.

```
operator@operator-pc# kubectl port-forward scos-acl-list-cache-black-v4 --namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep acl_list_items_in_last_list

# HELP acl list items in last list Count items returned in last list Query
# TYPE acl list items in last list gauge
acl list items in last list{exclude="",include="",ip="",no expired="true",pagination page no="",pagi
nation_per_page="",port="0"} 0
```

Описание метрик

- **acl_creator_items_from_source** (для базового сервиса генерации списка)

  Метрика, счётчик которой показывает количество записей, попавших в базовый сервис генерации списка из сервиса базы данных. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при выполнении проверок базового сервиса генерации списка необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе базового сервиса генерации списка значение счётчика данной метрики в момент поступления записей из базы данных должно быть больше 0.

- **acl_creator_items_sended_via_grpc** (для базового сервиса генерации списка)

  Метрика, счётчик которой показывает количество записей, отправленых из базового сервиса генерации списка в сервис хранения промежуточных данных. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при выполнении проверок базового сервиса генерации списка необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе базового сервиса генерации списка и сервиса хранения промежуточных данных значение счётчика данной метрики в момент отправки записей в сервис хранения промежуточных данных должно быть больше 0.

- **acl_list_items_in_last_list** (для сервиса хранения промежуточных данных)

  Метрика, счётчик которой показывает количество записей, находящихся в сервисе хранения промежуточных данных. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при выполнении проверок сервиса хранения промежуточных данных необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса хранения промежуточных данных и базовых сервисов генерации списка значение счётчика данной метрики в момент поступления записей в сервис хранения промежуточных данных должно быть больше 0.

Базовые сервисы генерации списка и сервисы хранения промежуточных данных состоят из N экземпляров. Каждый экземпляр базового сервиса генерации списка генерирует определенные типы записей, а каждый экземпляр сервиса хранения промежуточных данных хранит определенные типы записей. При обнаружении проблемы в работе этих сервисов необходимо выяснить, какие именно экземпляры работают неправильно.

7.2.2. Возможные причины неисправной работы

- экземпляр базового сервиса генерации списка находится в нерабочем состоянии, и данные не передаются в сервис хранения промежуточных данных;
- экземпляр сервиса хранения промежуточных данных находится в нерабочем состоянии, и базовый сервис генерации списка не может передать ему на хранение сгенерированные записи;
- неправильная конфигурация экземпляра базового сервиса генерации списка или экземпляра сервиса хранения промежуточных данных;
- нарушение сетевой связности между сервисом базы данных и экземпляром базового сервиса генерации списка;
- нарушение сетевой связности между базовым сервисом генерации списка и сервисом хранения промежуточных данных;
- несоответствующая версия образа для контейнера с экземпляром базового сервиса генерации списка или экземпляром сервиса хранения промежуточных данных.

7.2.3. Порядок выявления и устранения причин

1. Проверить журнал событий для контейнера с экземпляром базового сервиса генерации списка командой **`docker logs имя_сервиса`**

```
operator@operator-pc# docker logs spfs-gen-list-random
time="2020-04-28T14:39:01Z" level=info msg="config reload server started"
func=main.startConfigReloadServer file="/go/src/acl-creator/main.go:70"
time="2020-04-28T14:39:01Z" level=info msg="Start processing" func="acl-
creator/creator.(*Creator).StartMainLoop" file="/go/src/acl-
creator/creator/creator.go:117"
time="2020-04-28T14:39:01Z" level=error msg="rpc error: code = Unavailable desc
= connection error: desc = \"transport: Error while dialing dial tcp
192.168.160.4:8080: connect: connection refused\"" func="acl-
creator/creator.(*Creator).sendItemsToACLList" file="/go/src/acl-
creator/creator/creator.go:183"
```

или командой **`docker-compose logs имя_сервиса`**

```
operator@operator-pc# docker-compose logs spfs-gen-list-random
Attaching to spfs-gen-list-random
```

```
gen-list-random_1   | time="2020-04-28T14:39:01Z" level=info msg="config reload
server started" func=main.startConfigReloadServer file="/go/src/acl-
creator/main.go:70"
gen-list-random_1   | time="2020-04-28T14:39:01Z" level=info msg="Start
processing" func="acl-creator/creator.(*Creator).StartMainLoop"
file="/go/src/acl-creator/creator/creator.go:117"
gen-list-random_1   | time="2020-04-28T14:39:01Z" level=error msg="rpc error:
code = Unavailable desc = connection error: desc = \"transport: Error while
dialing dial tcp 192.168.160.4:8080: connect: connection refused\"" func="acl-
creator/creator.(*Creator).sendItemsToACLList" file="/go/src/acl-
creator/creator/creator.go:183"
```

В журнале событий контейнера с экземпляром базового сервиса генерации списка не должно быть систематических ошибок за последнее время.

Для получения дополнительной информации из журналов при неправильной работе сервиса использовать команду **`docker logs имя_сервиса --details`**.

```
operator@operator-pc# docker logs spfs-gen-list-random --details
time="2020-04-28T14:39:01Z" level=info msg="config reload server started"
func=main.startConfigReloadServer file="/go/src/acl-creator/main.go:70"
 time="2020-04-28T14:39:01Z" level=info msg="Start processing" func="acl-
creator/creator.(*Creator).StartMainLoop" file="/go/src/acl-
creator/creator/creator.go:117"
 time="2020-04-28T14:39:01Z" level=error msg="rpc error: code = Unavailable desc
= connection error: desc = \"transport: Error while dialing dial tcp
192.168.160.4:8080: connect: connection refused\"" func="acl-
creator/creator.(*Creator).sendItemsToACLList" file="/go/src/acl-
creator/creator/creator.go:183
```

2. Проверить журнал событий для контейнера с экземпляром сервиса хранения промежуточных данных командой **`kubectl logs имя_экземпляра_сервиса -n scos`**.

```
operator@operator-pc# kubectl logs scos-acl-list-cache-black-v4 -n scos
time="2020-04-23T14:13:25Z" level=info msg=Connected func=main.getDBConnection
file="/go/src/acl-list/main.go:87"
time="2020-04-23T14:13:25Z" level=info msg="Check for migrations" func="acl-
list/clickhouse.(*Connection).Migrate" file="/go/src/acl-
list/clickhouse/clickhouse.go:140"
time="2020-04-23T14:13:25Z" level=info msg="Migrate from 2 to 3 (steps 1)"
func="acl-list/clickhouse.(*Connection).Migrate" file="/go/src/acl-
list/clickhouse/clickhouse.go:143"
time="2020-04-23T14:13:27Z" level=info msg=Migrated func=main.getDBConnection
file="/go/src/acl-list/main.go:90"
time="2020-04-23T14:13:27Z" level=error msg="not applicable method
CreateIndexes" func="acl-list/clickhouse.(*Writer).CreateIndexes"
file="/go/src/acl-list/clickhouse/writer.go:223"
time="2020-04-23T14:13:27Z" level=info msg="Indexes are created"
func=main.getDBConnection file="/go/src/acl-list/main.go:93"
time="2020-04-23T14:13:27Z" level=info msg="Start processing loop"
func=main.getDBConnection file="/go/src/acl-list/main.go:96"
```

В журнале событий контейнера с экземпляром сервиса промежуточного хранения данных не должно быть систематических ошибок за последнее время.

3. Проверить, что экземпляры базовых сервисов генерации списка находятся в рабочем состоянии (STATUS – Up). Для этого следует отправить команды **`docker ps`** и **`docker-compose ps`** (данную команду необходимо отправлять из директории, в которой находится файл **docker-compose.yaml**).

```
operator@operator-pc# docker ps
CONTAINER ID IMAGE       COMMAND      CREATED       STATUS       PORTS         NAMES
fc4e761bddba acl-creator "/gen-list"  22 hours ago  Up 22 hours  0.0.0.0:2113  spfs-gen-list-tls
                                                                 ->2112/tcp
28d007a1d861 acl-creator "/gen-list"  22 hours ago  Up 22 hours  0.0.0.0:2114  spfs-gen-list-random
                                                                 ->2112/tcp
50e693e88808 acl-creator "/gen-list"  22 hours ago  Up 22 hours  0.0.0.0:2115  spfs-gen-list-hrandom
                                                                 ->2112/tcp

operator@operator-pc# docker-compose ps
         Name              Command            State         Ports
-------------------------------------------------------------------------------
spfs-gen-list-hrandom          /gen-list          Up            0.0.0.0:2115->2112/tcp
spfs-gen-list-random           /gen-list          Up            0.0.0.0:2114->2112/tcp
spfs-gen-list-tls              /gen-list          Up            0.0.0.0:2113->2112/tcp
```

4. Проверить, что экземпляр сервиса хранения промежуточных данных и его базы данных находятся рабочем состоянии (статус – Running). Для этого следует отправить команду **`kubectl get pods --namespace=scos | grep имя_экземпляра_сервиса`**.

```
operator@operator-pc# kubectl get pods --namespace=scos | grep list-cache
NAME                              READY   STATUS    RESTARTS   AGE
scos-acl-list-cache-black-v4      1/1     Running   3          2d22h
scos-acl-list-cache-db-o          2/2     Running   0          2d22h
```

5. Проверить значения параметров конфигурации экземпляра базового сервиса генерации списка, применяемые при запуске экземпляра сервиса. Для этого необходимо отправить команду **`cat docker-compose.yaml`**. Данную команду необходимо отправлять из директории, в которой находится файл **docker-compose.yaml**.

```
operator@operator-pc# cat docker-compose.yml
version: '3.7'

services:

  gen-list-random:
    image: hub.scos.ru/acl-creator:latest
    environment:
```

```
      - ITEM_TAG=random
      - MASK_LEN=32
      - CLICKHOUSEHOST=clickhouse
      - CLICKHOUSEPORT=9000
      - ACLLISTHOST=list-cache
      - ACLLISTPORT=8080
    ports:
      - "2114:2112/tcp"
```

Сравнить полученную конфигурацию с текущей конфигурацией запущенного контейнера экземпляра базового сервиса генерации списка с помощью команды **`docker inspect -f "{{json .Config.Env}}" имя_сервиса | jq`** (для выполнения команды необходимо установить утилиту для обработки json из командной строки jq)

```
operator@operator-pc# docker inspect -f "{{json .Config.Env}}" gen-list-random | jq
[
  "ITEM_TAG=random",
  "MASK_LEN=32",
  "CLICKHOUSEHOST=clickhouse",
  "CLICKHOUSEPORT=9000",
  "ACLLISTHOST=192.168.100.10",
  "ACLLISTPORT=30643",
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
]
```

В параметрах экземпляра сервиса коллектора журнала блокировок должны быть заданы правильные значения следующих параметров:

- **ITEM_TAG**. Этот параметр определяет тип записи, которую генерирует базовый сервис генерации списка. Каждый экземпляр сервиса должен генерировать записи для чёрного или белого списка. Экземпляры, генерирующие записи для черного списка, могут быть трёх типов: **random**, **hrandom** и **tls**;
- **MASK_LEN**. Этот параметр определяет длину генерируемого IP-адреса. Для IPv4 значение должно быть 32, для IPv6 – 128;
- **CLICKHOUSEHOST** и **CLICKHOUSEPORT**. Эти параметры указывают экземпляру базового сервиса генерации списка DNS-имя и номер порта сервиса базы данных. DNS-имя и номер порта должны совпадать с именем сервиса базы данных, указанным в файле **docker-compose.yaml**. При неверных значениях этих параметров сервис будет работать неправильно.

- **ACLLISTHOST** и **ACLLISTPORT**. Эти параметры указывают экземпляру базового сервиса генерации списка IP-адрес (или DNS-имя) и порт экземпляра сервиса хранения данных. IP-адрес (или DNS-имя) и номер порта должны совпадать со значениями, присвоенными экземпляру сервиса хранения промежуточных данных. При неверных значениях этих параметров сервис будет работать неправильно.

При необходимости следует внести необходимые правки в конфигурационный файл **docker-compose.yaml** и обновить конфигурацию сервиса командой **`docker-compose up --d`**.

6. Проверить текущие параметры конфигурации экземпляра сервиса хранения промежуточных данных, применяемые при запуске экземпляра сервиса. Для этого следует отправить команду **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].env"`**

```
operator@operator-pc# kubectl get pod scos-acl-list-cache-black-v4 --
namespace=scos -o json | jq ".spec.containers[].env"
[
  {
    "name": "DBUSERNAME",
    "value": "CLICKHOUSEUSER"
  },
  {
    "name": "DBPASSWORD",
    "value": "CLICKHOUSEPASS"
  },
  {
    "name": "DBDATABASE",
    "value": "acllist"
  },
  {
    "name": "IP_TYPE",
    "value": "IPv4"
  },
  {
    "name": "DB_TYPE",
    "value": "CLICKHOUSE"
  },
  {
    "name": "DBHOST",
    "value": "scos-list-cache-db-native"
  }
]
```

В выводе параметров экземпляра сервиса хранения промежуточных данных должны быть указаны правильные значения следующих параметров:

- **DB_TYPE**. Этот параметр определяет тип базы данных, используемый для экземпляра сервиса хранения промежуточных данных. Должно быть задано значение **CLICKHOUSE**.
- **DBHOST**. Этот параметр определяет DNS-имя сервиса базы данных. DNS-имя сервиса базы данных можно узнать с помощью команды **`kubectl get services --namespace=scos | grep list-cache`**.

```
operator@operator-pc# kubectl get services --namespace=scos | grep list-cache
NAME                         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)           AGE
scos-list-cache-v4           NodePort    10.43.217.12    <none>        30643:30643/TCP   8d
scos-list-cache-db-http      ClusterIP   10.43.197.255   <none>        8123/TCP          8d
scos-list-cache-db-metrics   ClusterIP   10.43.86.109    <none>        9116/TCP          8d
scos-list-cache-db-native    ClusterIP   10.43.245.44    <none>        9000/TCP          8d
scos-list-cache-metrics      ClusterIP   10.43.126.189   <none>        2112/TCP          8d
```

Указанное имя должно совпадать с именем, указанным в выводе команды выше (с префиксом *-db-native). При неверном значении этого параметра сервис хранения промежуточных данных будет работать неправильно.

При необходимости следует задать правильные значения параметров экземпляра сервиса хранения промежуточных данных в файле с информацией о чарте (**chart.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

7. Проверить сетевую связность между экземпляром базового сервиса генерации списка и сервисом базы данных. Например, командой **`docker exec -it имя_сервиса sh`** зайти в оболочку экземпляра базового сервиса генерации списка и проверить доступность сервиса базы данных с помощью утилиты **ping**.

```
operator@operator-pc# docker exec -it spfs-gen-list-random sh
/srv # ping -c4 spfs-clickhouse
PING spfs-clickhouse (192.168.160.6): 56 data bytes
64 bytes from 192.168.160.6: seq=0 ttl=64 time=0.044 ms
64 bytes from 192.168.160.6: seq=1 ttl=64 time=0.052 ms
64 bytes from 192.168.160.6: seq=2 ttl=64 time=0.051 ms
64 bytes from 192.168.160.6: seq=3 ttl=64 time=0.050 ms

--- spfs-clickhouse ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
```

```
round-trip min/avg/max = 0.044/0.049/0.052 ms
```

Проверить доступность сервиса базы данных по прослушивающему порту, указанному в файле **docker-compose.yaml**, с помощью общедоступных системных инструментов: **netcat**, **nmap**.

```
/srv # nmap spfs-clickhouse -p 9000
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 15:50 UTC
Nmap scan report for spfs-clickhouse (192.168.160.6)
Host is up (0.000071s latency).
rDNS record for 192.168.160.6: spfs-clickhouse.spfs_default

PORT     STATE SERVICE
9000/tcp open  cslistener
MAC Address: 02:42:C0:A8:A0:06 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 0.68 seconds
```

При отсутствии доступа к сервису базы данных по целевому порту необходимо устранить нарушение сетевой связности.

8. Проверить сетевую связность между базовым сервисом генерации списков и сервисом промежуточного хранения данных. Для получения данных из базового сервиса генерации списка сервис промежуточного хранения данных должен иметь тип сетевой службы **NodePort**.

Информацию об используемом типе службы можно с помощью команды **`kubectl get services --namespace=scos | grep list-cache | grep NodePort`**.

```
operator@operator-pc# kubectl get services --namespace=scos | grep list-cache |
grep NodePort
NAME                TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
scos-list-cache-v4   NodePort   10.43.217.12   <none>       30643:30643/TCP  8d
```

Эта же команда предоставляет информацию о номере TCP-порта для сервиса хранения промежуточных данных. Тип службы **NodePort** позволяет сервису промежуточных данных получать требуемые данные (предварительные списки) из-за пределов кластера kubernetes.

Следующий этап проверки сетевой связности – выявление IP-адреса экземпляра сервиса хранения промежуточных данных, который будет равен любому IP-адресу узла кластера kubernetes. Узнать IP-адреса всех узлов кластера можно с

помощью команды **kubectl get nodes -o jsonpath='{ $.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'**.

```
operator@operator-pc# kubectl get nodes -o jsonpath='{
$.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'
192.168.100.10 192.168.100.11 192.168.100.12 192.168.100.13%
```

После получения информации о том, какой IP-адрес и TCP-порт используются экземпляром сервиса хранения промежуточных данных, выполняется проверка сетевой доступности со стороны экземпляра базового сервиса генерации списка с помощью общедоступных системных инструментов: **netcat**, **nmap**.

```
operator@operator-pc# docker exec -it spfs_gen-list-random_1 sh
/srv # nmap 192.168.100.10 -p 30643
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 16:52 UTC
Nmap scan report for rancher.ttraf.ru (192.168.100.10)
Host is up (0.00046s latency).

PORT      STATE SERVICE
30643/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

При отсутствии доступа к экземпляру сервиса хранения промежуточных данных по целевому порту следует устранить нарушение сетевой связности.

9. Проверить версию образа работающего экземпляра сервиса хранения промежуточных данных с помощью команды **kubectl get pod *имя_экземпляра_сервиса* --namespace=scos -o json | jq ".spec.containers[].image"**.

```
operator@operator-pc# kubectl get pod scos-acl-list-cache-black-v4 --
namespace=scos -o json | jq ".spec.containers[].image"
"acl-list:v2.6.0"
```

Проверить в файле с информацией о чарте (**chart.yaml**) соответствие версии образа для экземпляра сервиса промежуточного хранения данных, при необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **--upgrade**.

10. Проверить версию образа экземпляра сервиса командой **docker inspect -f "{{json .Config.Image}}" *имя_сервиса* | jq**.

```
docker inspect -f "{{json .Config.Image}}" spfs-gen-list-random | jq
"acl-creator:latest"
или docker ps | grep имя_сервиса
operator@operator-pc# docker ps | grep spfs-gen-list-random
CONTAINER ID IMAGE       COMMAND       CREATED       STATUS       PORTS          NAMES
28d007a1d861 acl-creator "/gen-list"   22 hours ago  Up 22 hours  0.0.0.0:2114   spfs-gen-list-random
                                                                  ->2112/tcp
```

При необходимости следует скачать новый образ из репозитория (команда **`docker pull`** или **`docker-compose pull`**), внести соответствующие правки в конфигурационный файл **docker-compose.yaml** и обновить развёрнутые сервисы командой **`docker-compose up -d`**.

7.3. Не работают сервисы создания основных списков или сервисы хранения основных списков

Сервисы создания основных списков отвечают за формирование основных списков из данных, хранящихся в сервисе хранения промежуточных данных. В свою очередь, сервисы хранения основных списков обеспечивают доступ к основным спискам для возможности их дальнейшей обработки другими сервисами в составе ПЦОС.

7.3.1. Описание проблемы

После правильной настройки оборудования фильтрации и начала отправки журналов пользовательских блокировок, настройки и запуска сервисов коллектора журнала блокировок и базы данных базовые сервисы генерации списков и сервисы хранения промежуточных данных работают правильно, но значения счётчиков для сервисов создания основных списков или сервисов хранения основных списков не соответствуют критериям исправной работы, а именно:

- значения счётчиков для метрик сервисов создания основных списков не приходят по запросу:

```
operator@operator-pc# kubectl port-forward scos-acl-creator-from-cache-black-v4
--namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_creator_items_sended_via_grpc|acl_creator_items_from_source"
```

```
Handling connection for 5000
E0429 15:45:55.670279 24686 portforward.go:400] an error occurred forwarding
5000 -> 2112: error forwarding port 2112 to pod
d5f6b9c8c1e8f64646886f04bf696ab770ce45b4c7b7f117f4068d0c44a90ac5, uid : exit
status 1: 2020/04/29 12:45:55 socat[27411] E connect(5, AF=2 127.0.0.1:2112,
16): Connection refused
```

- значения счётчиков для метрик сервисов создания основных списков в течение продолжительного времени равны нулю, однако при этом записи от EcoFilter успешно поступают в сервис коллектора журнала блокировок, базовые сервисы генерации списков успешно получают записи от сервиса базы данных и успешно отправляют сгенерированные записи в сервис хранения промежуточных данных.

```
operator@operator-pc# kubectl port-forward scos-acl-creator-from-cache-black-v4
--namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_creator_items_sended_via_grpc|acl_creator_items_from_source"

# HELP acl_creator_items_from_source Number of records returned from source in
last query
# TYPE acl_creator_items_from_source gauge
acl_creator_items_from_source 0
# HELP acl_creator_items_sended_via_grpc Number of records sended via grpc
# TYPE acl_creator_items_sended_via_grpc gauge
acl_creator_items_sended_via_grpc 0
```

- значения счётчиков для метрик сервисов хранения основных списков не приходят по запросу:

```
operator@operator-pc# kubectl port-forward scos-acl-list-black-v4 --
namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep
acl_list_items_in_last_list
Handling connection for 5000
E0429 15:45:55.670279 24686 portforward.go:400] an error occurred forwarding
5000 -> 2112: error forwarding port 2112 to pod
d5f6b9c8c1e8f64646886f04bf696ab770ce45b4c7b7f117f4068d0c44a90ac5, uid : exit
status 1: 2020/04/29 12:45:55 socat[27411] E connect(5, AF=2 127.0.0.1:2112,
16): Connection refused
```

- значения счётчиков для метрик сервисов хранения основных списков в течение продолжительного времени остаются равными нулю, однако при этом записи журналов EcoFilter поступают в сервис коллектора журнала блокировок, базовые сервисы генерации списков получают записи из сервиса базы данных и передают сгенерированные записи в сервис хранения промежуточных данных, а сервисы создания основных списков формируют записи для сервисов хранения основных списков.

```
operator@operator-pc# kubectl port-forward scos-acl-list-black-v4 --
namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep
acl_list_items_in_last_list

acl_list_items_in_last_list{exclude="",include="rkn,static",ip="",no_expired="tr
ue",pagination_page_no="",pagination_per_page="",port="0"} 0
acl_list_items_in_last_list{exclude="rkn,static",include="tls,hrandom",ip="",no_
expired="true",pagination_page_no="",pagination_per_page="",port="0"} 0
acl_list_items_in_last_list{exclude="static,rknip",include="tls",ip="",no_expire
d="true",pagination_page_no="",pagination_per_page="",port="0"} 0
acl_list_items_in_last_list{exclude="static,rknip,tls",include="hrandom",ip="",n
o_expired="true",pagination_page_no="",pagination_per_page="",port="0"} 0
```

Описание метрик

- **acl_creator_items_from_source** (для сервиса создания основных списков)

  Метрика, счётчик которой показывает количество записей, поступивших в сервис создания основных списков из сервиса хранения промежуточных списков. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при выполнении проверок сервиса создания основных списков необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе базового сервиса генерации списка значение счётчика данной метрики в момент получения записей от сервиса хранения промежуточных списков должно быть больше 0.

- **acl_creator_items_sended_via_grpc** (для сервиса создания основных списков)

  Метрика, счётчик которой показывает количество записей, отправленных сервисом создания основных списков в сервис хранения основных списков. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при выполнении проверок сервиса создания основных списков необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса создания основных списков и сервиса хранения основных списков значение счётчика данной метрики в момент отправки записей в сервис хранения основных списков должно быть больше 0.

- **acl_list_items_in_last_list** (для сервиса хранения основных данных)

  Метрика, счётчик которой показывает количество записей в сервисе хранения основных данных.

  При правильной работе сервиса хранения основных данных и сервиса создания основных списков значение счётчика данной метрики должно быть больше 0.

Сервисы создания основных списков и сервисы хранения основных списков состоят из N экземпляров. Каждый экземпляр сервиса создания основных списков генерирует определённые типы записей, а каждый экземпляр сервиса хранения основных данных хранит определённые типы записей (чёрный список для IPv4, белый список для IPv4, чёрный список для IPv6 и белый список для IPv6). При обнаружении проблемы в работе этих сервисов необходимо выяснить, какие именно экземпляры работают неправильно.

7.3.2. Возможные причины неисправной работы

- экземпляр сервиса создания основных списков находится в нерабочем состоянии, и данные не передаются в сервис хранения основных списков;
- экземпляр сервиса хранения основных списков находится в нерабочем состоянии, и сервис создания основных списков не может передать ему на хранение сгенерированные записи;
- неправильная конфигурация экземпляра сервиса создания основных списков или экземпляра сервиса хранения основных списков;
- нарушение сетевой связности между сервисом хранения промежуточных данных и экземпляром сервиса создания основных списков;
- нарушение сетевой связности между экземпляром сервиса создания основных списков и сервисом хранения основных списков;
- несоответствующая версия образа для контейнера с экземпляром сервиса создания основных списков или экземпляром сервиса хранения основных списков.

7.3.3. Порядок выявления и устранения причин

1. Проверить журнал событий для контейнера с экземпляром сервиса создания основных списков командой **`kubectl logs имя_экземпляра_сервиса -n scos`**.

```
operator@operator-pc# kubectl logs  scos-acl-creator-from-cache-black-v4 --
namespace=scos
time="2020-04-17T15:30:19Z" level=info msg="no config reload server"
func=main.startConfigReloadServer file="/go/src/acl-creator/main.go:72"
time="2020-04-17T15:30:19Z" level=info msg="Start processing" func="acl-
creator/creator.(*Creator).StartMainLoop" file="/go/src/acl-
creator/creator/creator.go:117"
time="2020-04-20T09:50:18Z" level=error msg="rpc error: code = Unavailable desc
= connection error: desc = \"transport: Error while dialing dial tcp
10.210.9.250:30643: connect: connection refused\"" func="acl-
creator/acllist.(*Source).GetItemsChannel.func1" file="/go/src/acl-
creator/acllist/datasource.go:100"
```

В журнале событий контейнера с экземпляром сервиса создания основных списков за последнее время не должно быть систематических ошибок.

2. Проверить журнал событий для контейнера с экземпляром сервиса хранения основных списков командой **kubectl logs имя_экземпляра_сервиса -n scos -c имя основного_контейнера_для_экземпляра_сервиса**.

```
operator@operator-pc# kubectl logs scos-acl-list-black-v4 --namespace=scos -c
acl-list-black-v4
time="2020-04-23T14:13:25Z" level=info msg=Connected func=main.getDBConnection
file="/go/src/acl-list/main.go:87"
time="2020-04-23T14:13:25Z" level=info msg="No migrations" func="acl-
list/mongodb.(*Connection).Migrate" file="/go/src/acl-
list/mongodb/mongodb.go:110"
time="2020-04-23T14:13:25Z" level=info msg=Migrated func=main.getDBConnection
file="/go/src/acl-list/main.go:90"
time="2020-04-23T14:13:25Z" level=error msg="index created: tags_name_idx"
func="acl-list/mongodb.(*Connection).createTagNamesIndex" file="/go/src/acl-
list/mongodb/mongodb.go:598"
time="2020-04-23T14:13:25Z" level=info msg="index created: tags_expiredat_idx"
func="acl-list/mongodb.(*Connection).createTagExpiredIndex" file="/go/src/acl-
list/mongodb/mongodb.go:621"
time="2020-04-23T14:13:25Z" level=error msg="index creation failed:
(IndexKeySpecsConflict) Index must have unique name.The existing index: { v: 2,
key: { tags.name: 1, tags.expiredat: 1 }, name: \"tags_name_expiredat_idx\", ns:
\"blacklist.acllist\" } has the same name as the requested index: { v: 2, key: {
tags.expiredat: 1, tags.name: 1 }, name: \"tags_name_expiredat_idx\", ns:
\"blacklist.acllist\" }" func="acl-
list/mongodb.(*Connection).createTagNameAndExpiredIndex" file="/go/src/acl-
list/mongodb/mongodb.go:643"
time="2020-04-23T14:13:25Z" level=error msg="index creation failed:
(IndexKeySpecsConflict) Index must have unique name.The existing index: { v: 2,
unique: true, key: { mask: 1, port: 1, ip: 1 }, name: \"ip_mask_port_idx\", ns:
\"blacklist.acllist\" } has the same name as the requested index: { v: 2,
unique: true, key: { ip: 1, mask: 1, port: 1 }, name: \"ip_mask_port_idx\", ns:
\"blacklist.acllist\" }" func="acl-
list/mongodb.(*Connection).createFindOneItemIndex" file="/go/src/acl-
list/mongodb/mongodb.go:668"
time="2020-04-23T14:13:25Z" level=info msg="Indexes are created"
func=main.getDBConnection file="/go/src/acl-list/main.go:93"
time="2020-04-23T14:13:25Z" level=info msg="Start processing loop"
func=main.getDBConnection file="/go/src/acl-list/main.go:96"
time="2020-04-23T14:13:25Z" level=info msg="Start processing items inserting"
func="acl-list/mongodb.(*Connection).processInsertItems" file="/go/src/acl-
list/mongodb/mongodb.go:136"
```

В журнале событий контейнера с экземпляром сервиса хранения основных списков за последнее время не должно быть систематических ошибок.

3. Проверить, что экземпляр сервиса создания основных списков находится в рабочем состоянии (STATUS – Running). Для этого следует отправить команду **kubectl get pods -n scos | grep имя_экземпляра_сервиса**.

```
operator@operator-pc# kubectl get pods -n scos |grep scos-acl-creator-from-cache-black-v4
NAME                                  READY   STATUS    RESTARTS   AGE
scos-acl-creator-from-cache-black-v4   1/1     Running   0          12d
```

4. Проверить, что экземпляр сервиса хранения основных списков и экземпляр его базы данных находятся рабочем состоянии (STATUS – Running). Для этого следует отправить команду **kubectl get pods --namespace=scos | grep имя_экземпляра_сервиса**.

```
operator@operator-pc# kubectl get pods --namespace=scos | grep scos-acl-list-black-v4

NAME                                READY   STATUS    RESTARTS   AGE
scos-acl-list-black-v4     2/2     Running   0          6d4h
scos-acl-list-black-v4-db  2/2     Running   0          8d
```

5. Проверить текущие параметры конфигурации экземпляра сервиса создания основных списков с помощью команды **kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].env".**

```
operator@operator-pc# kubectl get pod scos-acl-creator-from-cache-black-v4 --
namespace=scos -o json | jq ".spec.containers[].env"
[
  {
    "name": "ACLLISTHOST",
    "value": "server.scos.ru"
  },
  {
    "name": "ACLLISTPORT",
    "value": "30642"
  },
  {
    "name": "SOURCE_TYPE",
    "value": "ACL_LIST"
  },
  {
    "name": "ACL_LIST_SOURCE_HOST",
    "value": "server.scos.ru "
  },
  {
    "name": "ACL_LIST_SOURCE_PORT",
    "value": "30643"
  }
]
```

В конфигурации экземпляра сервиса создания основных списков должны быть указаны правильные значения следующих параметров:

- **ACLLISTHOST** и **ACLLISTPORT**. Эти параметры определяют сетевую связность с сервисом хранения основных списков. Значение **ACLLISTHOST** должно содержать Правильное DNS-имя или IP-адрес сервиса хранения основных списков, а значение **ACLLISTPORT** – правильный номер порта сервиса хранения основных списков. Правильные значения этих параметров можно узнать следующим способом:

  Шаг 1. Узнать номер порта и тип сетевой службы для экземпляра сервиса хранения основных списков с помощью команды **`kubectl get services --namespace=scos | grep scos-acl-list-black-v4`**.

```
operator@operator-pc# kubectl get services --namespace=scos | grep scos-acl-list-black-v4
NAME                       TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)             AGE
scos-acl-list-black-v4     NodePort    10.43.117.56  <none>        30642:30642/TCP     9d
scos-acl-list-black-v4-db ClusterIP   10.43.36.248  <none>        27017/TCP,9216/TCP 9d
```

  Для экземпляра сервиса хранения основных списков используется тип сетевой службы **NodePort** и TCP-порт 30642. Сетевая служба **NodePort** обеспечивает возможность доступа к экземпляру сервиса из-за пределов кластера.

  Шаг 2. Узнать IP-адрес экземпляра сервиса хранения основных списков, который будет равен любому IP-адресу узла кластера. Узнать IP-адреса всех узлов кластера можно командой **`kubectl get nodes -o jsonpath='{ $.items[*].status.addresses[?(@.type=="InternalIP")].address }'`**.

```
operator@operator-pc# kubectl get nodes -o jsonpath='{
$.items[*].status.addresses[?(@.type=="InternalIP")].address }'
192.168.100.10 192.168.100.11 192.168.100.12 192.168.100.13%
```

  Вместо IP-адреса можно использовать DNS-имя одного из узлов кластера.

```
nslookup 192.168.100.10
Server:     192.168.100.2
Address:    192.168.100.2#53
```

```
10.100.168.192.in-addr.arpa   name = server.scos.ru.
```

В приведённом выше примере правильными значениями параметров **ACLLISTHOST** и **ACLLISTPORT** экземпляра сервиса создания основных списков являются, соответственно, **server.scos.ru** (или **192.168.100.10**) и **30642**.

- **ACL_LIST_SOURCE_HOST** и **ACL_LIST_SOURCE_PORT**. Эти параметры определяют сетевую связность с сервисом хранения промежуточных данных. Значение **ACL_LIST_SOURCE_HOST** должно содержать правильное DNS-имя или IP-адрес сервиса хранения промежуточных данных, а значение **ACL_LIST_SOURCE_PORT** – правильный номер порта сервиса хранения промежуточных данных. Правильные значения этих параметров можно узнать следующим способом:

  Шаг 1. Узнать номер порта и тип сетевой службы для экземпляра сервиса хранения промежуточных данных с помощью команды **`kubectl get services --namespace=scos | grep scos-acl-list-cache-black-v4`**.

```
operator@operator-pc# kubectl get services -n scos | grep scos-acl-list-cache-black-v4
NAME                            TYPE      CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
scos-acl-list-cache-black-v4    NodePort  10.43.217.12  <none>     30643:30643/TCP     9d
scos-acl-list-cache-black-v4-db ClusterIP 10.43.245.44  <none>     9000/TCP            9d
```

  Для экземпляра сервиса хранения промежуточных данных используется тип сетевой службы **NodePort** и TCP-порт 30643. Сетевая служба **NodePort** обеспечивает возможность доступа к экземпляру сервиса из-за пределов кластера.

  Шаг 2. Узнать IP-адрес экземпляра сервиса хранения промежуточных данных, который будет равен любому IP-адресу узла кластера. Узнать IP-адреса всех узлов кластера можно командой **`kubectl get nodes -o jsonpath='{ $.items[*].status.addresses[?(@.type=="InternalIP")].address }'`**.

```
operator@operator-pc# kubectl get nodes -o jsonpath='{
$.items[*].status.addresses[?(@.type=="InternalIP")].address }'
192.168.100.10 192.168.100.11 192.168.100.12 192.168.100.13%
```

  Вместо IP-адреса можно использовать DNS-имя одного из узлов кластера.

```
nslookup 192.168.100.10
Server:     192.168.100.2
Address:    192.168.100.2#53

10.100.168.192.in-addr.arpa   name = server.scos.ru.
```

В приведённом выше примере правильными значениями параметров **ACLLISTHOST** и **ACLLISTPORT** экземпляра сервиса создания основных списков являются, соответственно, **server.scos.ru** (или **192.168.100.10**) и **30643**.

- **SOURCE_TYPE.** Значение данного параметра должно быть по умолчанию **ACL_LIST**.

При необходимости следует задать правильные значения параметров экземпляра сервиса создания основных списков в файле со значениями чарта (**values.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

6. Проверить текущие параметры конфигурации экземпляра сервиса хранения основных списков командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].env"`**.

```
operator@operator-pc# kubectl get pod scos-acl-list-black-v4 --namespace=scos -o
json | jq ".spec.containers[].env"
[
  {
    "name": "DBUSERNAME",
    "value": "list"
  },
  {
    "name": "DBPASSWORD",
    "value": "mongodb-password"
  },
  {
    "name": "DBDATABASE",
    "value": "blacklist"
  },
  {
    "name": "IP_TYPE",
    "value": "IPv4"
  },
  {
    "name": "DB_TYPE",
    "value": "MONGODB"
  },
  {
    "name": "DBHOST",
```

```
    "value": "scos-acl-list-black-v4-db"
  }
]
[
  {
    "name": "GRPC_PROXY",
    "value": "http://server.scos.ru:30273/acl/black/v4"
  }
]
```

В конфигурации экземпляра сервиса хранения основных списков должны быть заданы правильные значения следующих параметров:

- **DBUSERNAME.** Параметр определяет имя пользователя для доступа к базе данных экземпляра сервиса хранения основных списков. Значение данного параметра должно соответствовать значению в конфигурации экземпляра базы данных для данного экземпляра сервиса хранения основных списков;

- **DBPASSWORD.** Параметр определяет пароль для доступа к базе данных экземпляра сервиса хранения основных списков. Значение данного параметра должно соответствовать значению в конфигурации экземпляра базы данных для данного экземпляра сервиса хранения основных списков;

- **DBDATABASE.** Параметр определяет имя базы данных экземпляра сервиса хранения основных списков. Значение данного параметра должно соответствовать значению в конфигурации экземпляра базы данных для данного экземпляра сервиса хранения основных списков;

- **IP_TYPE.** Параметр определяет тип адресов, данные о которых хранятся в экземпляре хранения основных списков. Параметр может принимать два значения: IPv4 и IPv6;

- **DB_TYPE.** Параметр определяет тип базы данных (документо-ориентированная СУБД). Значение по умолчанию должно быть **MONGODB**;

- **DBHOST.** Параметр определяет сетевую связность между экземпляром сервиса хранения основных списков и его экземпляром базы данных. Значение данного параметра должно содержать правильное DNS-имя

сервиса базы данных для данного экземпляра сервиса. Правильное значение данного параметра можно узнать с помощью команды **kubectl get services --namespace=scos | grep scos-acl-list-black-v4**.

```
operator@operator-pc# kubectl get services --namespace=scos | grep scos-acl-list-black-
v4-db
NAME                        TYPE       CLUSTER-IP    EXTERNAL-IP   PORT(S)             AGE
scos-acl-list-black-v4-db   ClusterIP  10.43.36.248  <none>        27017/TCP,9216/TCP  9d
```

При необходимости следует задать правильные значения параметров экземпляра сервиса хранения основных списков в файле со значениями чарта (**values.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **--upgrade**.

7. Проверить сетевую связность между сервисом хранения промежуточных данных и сервисом создания основных списков. Для этого необходимо зайти в оболочку экземпляра сервиса создания основных списков с помощью команды **kubectl exec -it scos-acl-creator-from-cache-black-v4 --namespace=scos sh**

```
operator@operator-pc# kubectl exec -ti scos-acl-creator-from-cache-black --
namespace=scos  sh
/srv #
```

и проверить доступность экземпляра сервиса хранения промежуточных данных по прослушивающему порту с помощью общедоступных системных инструментов: **nmap**, **netcat**.

```
/srv # nmap server.scos.ru -p 30643
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
30643/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

При необходимости следует устранить нарушение сетевой связности между экземпляром сервиса создания основных списков и экземпляром сервиса хранения промежуточных данных.

8. Проверить сетевую связность между между экземпляром сервиса создания основных списков и экземпляром сервиса хранения основных

списков. Для этого необходимо зайти в оболочку экземпляра сервиса создания основных списков с помощью команды **kubectl exec -it scos-acl-creator-from-cache-black-v4 --namespace=scos sh**

```
operator@operator-pc# kubectl exec -ti scos-acl-creator-from-cache-black --
namespace=scos  sh
/srv #
```

и проверить доступность экземпляра сервиса хранения основных списков по прослушивающему порту с помощью общедоступных системных инструментов: **nmap**, **netcat**.

```
operator@operator-pc# kubectl exec -ti scos-acl-creator-from-cache-black  --
namespace=scos sh
/srv # nmap server.scos.ru -p 30642
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
30642/tcp open  unknown
Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

При необходимости следует устранить нарушение сетевой связности между экземпляром сервиса создания основных списков и экземпляром сервиса хранения основных списков.

9. Проверить версию образа работающего экземпляра сервиса создания основных списков командой **kubectl get pod *имя_экземпляра_сервиса* --namespace=scos -o json | jq ".spec.containers[].image"**.

```
operator@operator-pc# kubectl get pod scos-acl-creator-from-cache-black-v4--
namespace=scos -o json | jq ".spec.containers[].image"
"hub.scos.ru/acl-creator:v2.1.0"
```

Следует проверить в файле с информацией о чарте (**chart.yaml**) соответствие версии образа для экземпляра сервиса создания основных списков, при необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **--upgrade**.

10. Проверить версию образа работающего экземпляра сервиса хранения основных списков командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].image"`**.

```
operator@operator-pc# kubectl get pod scos-acl-list-black-v4  --namespace=scos -
o json | jq ".spec.containers[].image"
"hub.scos.ru/acl-list:v2.6.0"
```

Следует проверить в файле с информацией о чарте (**chart.yaml**) соответствие версии образа для экземпляра сервиса хранения основных списков, при необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

7.4. Не работают сервисы сравнения списков и сервисы выгрузки списков на узлы фильтрации

Сервисы сравнения списков отвечают за сопоставление записей в основных чёрных и белых списках, передачу результирующих списков сервисам выгрузки и предоставление доступа к чёрным спискам оборудованию первого уровня фильтрации (EcoFilter). Сервисы выгрузки списков обеспечивают отправку актуальных серых и чёрных списков фильтрации на оборудование балансировки трафика в составе второго уровня фильтрации (EcoHighway).

7.4.1. Описание проблемы

После правильной настройки оборудования фильтрации и начала отправки журналов пользовательских блокировок, настройки и запуска сервисов коллектора журналов блокировок и базы данных основные сервисы генерации списков и сервисы хранения промежуточных данных работают правильно, сервисы создания основных списков и сервисы хранениях основных списков также работают правильно, но значения счётчиков метрик сервисов сравнения списков и/или сервисов доставки списков не соответствуют критериям успешной работы, а именно:

- значения метрик сервисов сравнения списков не приходят по запросу:

```
operator@operator-pc# kubectl port-forward scos-acl-differ-v4 --namespace=scos
5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_differ_black_list_len|acl_differ_white_list_len|acl_differ_diffed_list_len"

Handling connection for 5000
E0429 15:45:55.670279 24686 portforward.go:400] an error occurred forwarding
5000 -> 2112: error forwarding port 2112 to pod
d5f6b9c8c1e8f64646886f04bf696ab770ce45b4c7b7f117f4068d0c44a90ac5, uid : exit
status 1: 2020/04/29 12:45:55 socat[27411] E connect(5, AF=2 127.0.0.1:2112,
16): Connection refused
```

- значения метрик сервисов сравнения списков продолжительное время остаются равными нулю, однако при этом:
  - записи журналов от EcoFilter поступают в сервис коллектора журнала блокировок;
  - базовые сервисы генерации списков получают записи от сервиса базы данных и передают сгенерированные записи в сервис хранения промежуточных данных;
  - сервисы создания основных списков отправляют записи в сервисы хранения основных списков.

```
operator@operator-pc# kubectl port-forward scos-acl-differ-v4--namespace=scos
5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_differ_black_list_len|acl_differ_white_list_len|acl_differ_diffed_list_len"

# HELP acl_differ_black_list_len Number of records received from black list.
# TYPE acl_differ_black_list_len gauge
acl_differ_black_list_len{black_exclude="rkn,static",black_include="tls,hrandom"
,white_exclude="",white_include="static"} 0
acl differ black list len{black exclude="rkn,static,tls,hrandom",black include="
random",white_exclude="tag3,tag4",white_include="tag1,tag2"} 0
acl_differ_black_list_len{black_exclude="static,rknip",black_include="tls",white
_exclude="",white_include="static"} 0
acl_differ_black_list_len{black_exclude="static,rknip,tls",black_include="hrando
m",white_exclude="",white_include="static"} 0
# HELP acl_differ_diffed_list_len Number of records sended to destination.
# TYPE acl_differ_diffed_list_len gauge
acl_differ_diffed_list_len{black_exclude="rkn,static",black_include="tls,hrandom
",white_exclude="",white_include="static"} 0
acl_differ_diffed_list_len{black_exclude="rkn,static,tls,hrandom",black_include=
"random",white_exclude="tag3,tag4",white_include="tag1,tag2"} 0
```

```
acl_differ_diffed_list_len{black_exclude="static,rknip",black_include="tls",whit
e_exclude="",white_include="static"} 0
acl_differ_diffed_list_len{black_exclude="static,rknip,tls",black_include="hrand
om",white_exclude="",white_include="static"} 0
# HELP acl_differ_white_list_len Number of records received from white list.
# TYPE acl_differ_white_list_len gauge
acl_differ_white_list_len{black_exclude="rkn,static",black_include="tls,hrandom"
,white_exclude="",white_include="static"} 0
acl_differ_white_list_len{black_exclude="rkn,static,tls,hrandom",black_include="
random",white exclude="tag3,tag4",white include="tag1,tag2"} 0
acl_differ_white_list_len{black_exclude="static,rknip",black_include="tls",white
_exclude="",white_include="static"} 0
acl_differ_white_list_len{black_exclude="static,rknip,tls",black_include="hrando
m",white_exclude="",white_include="static"} 0
```

- значения метрик сервисов доставки списков на узлы фильтрации не приходят по запросу:

```
operator@operator-pc# kubectl port-forward scos-acl-manager-block-v4 --
namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_manager_bgp_received_records|acl_manager_source_received_records|acl_manage
r_bgp_send_add_records|acl_manager_bgp_send_del_records|acl_manager_update_in_pr
ogress"

E0429 15:45:55.670279 24686 portforward.go:400] an error occurred forwarding
5000 -> 2112: error forwarding port 2112 to pod
d5f6b9c8c1e8f64646886f04bf696ab770ce45b4c7b7f117f4068d0c44a90ac5, uid : exit
status 1: 2020/04/29 12:45:55 socat[27411] E connect(5, AF=2 127.0.0.1:2112,
16): Connection refused
```

- значения метрик сервисов доставки списков продолжительное время остаются равными нулю, однако при этом:
  - записи журналов от EcoFilter поступают в сервис коллектора журнала блокировок;
  - базовые сервисы генерации списков получают записи от сервиса базы данных и передают сгенерированные записи в сервис хранения промежуточных данных;
  - сервисы создания основных списков отправляют записи в сервисы хранения основных списков;
  - сервисы сравнения основных списков формируют итоговые списки для сервисов доставки списков.

```
operator@operator-pc# kubectl port-forward scos-acl-manager-block-v4 --
namespace=scos 5000:2112
Forwarding from 127.0.0.1:5000 -> 2112
Forwarding from [::1]:5000 -> 2112

operator@operator-pc# curl -s 127.0.0.1:5000/metrics | grep -E
"acl_manager_bgp_received_records|acl_manager_source_received_records|acl_manage
r_bgp_send_add_records|acl_manager_bgp_send_del_records|acl_manager_update_in_pr
ogress"

# HELP acl_manager_bgp_received_records The total number of received BGP records
# TYPE acl_manager_bgp_received_records gauge
acl_manager_bgp_received_records 0
# HELP acl_manager_bgp_send_add_records The total number of added BGP records
# TYPE acl_manager_bgp_send_add_records gauge
acl_manager_bgp_send_add_records 0
# HELP acl_manager_bgp_send_del_records The total number of deleted BGP records
# TYPE acl_manager_bgp_send_del_records gauge
acl_manager_bgp_send_del_records 0
# HELP acl_manager_source_received_records The total number of received records
from source
# TYPE acl_manager_source_received_records gauge
acl_manager_source_received_records 0
# HELP acl_manager_update_in_progress Current status of update loop
# TYPE acl_manager_update_in_progress gauge
acl_manager_update_in_progress 0
# HELP acl_manager_update_in_progress Current status of update loop
# TYPE acl_manager_update_in_progress gauge
acl_manager_update_in_progress 0
```

Описание метрик

- **acl_differ_black_list_len** (для сервиса сравнения списков)

  Метрика, счётчик которой показывает количество записей, поступивших в сервис сравнения списков от сервиса хранения основных (чёрных) списков. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса сравнения списков необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса хранения основных (чёрных) списков и сервиса доставки списков на узлы фильтрации значение счётчика этой метрики в момент получения записей от сервиса хранения основных списков должно быть больше 0.

- **acl_differ_white_list_len** (для сервиса сравнения списков)

  Метрика, счётчик которой показывает количество записей, поступивших в сервис сравнения списков от сервиса хранения основных (белых) списков. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса сравнения списков необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса хранения основных (белых) списков и сервиса доставки списков на узлы фильтрации значение счётчика этой метрики в момент получения записей от сервиса хранения основных списков должно быть больше 0.

- **acl_differ_diffed_list_len** (для сервиса сравнения списков)

  Метрика, счётчик которой показывает количество записей, успешно обработанных сервисом сравнения списков и успешно отправленных сервису доставки списков на узлы фильтрации. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса сравнения списков необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса хранения основных (чёрных и белых) списков и сервиса доставки списков значение счётчика этой метрики в момент получения записей от сервиса хранения основных списков должно быть больше 0.

- **acl_manager_source_received_records** (для сервиса доставки списков)

  Метрика, счётчик которой показывает количество записей, полученных сервисом доставки списков от сервиса сравнения списков. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок

сервиса доставки списков на узлы фильтрации необходимо отправить несколько запросов этой метрики за короткий интервал времени.

При правильной работе сервиса сравнения списков и сервиса доставки списков значение счётчика этой метрики в момент получения записей от сервиса сравнения списков должно быть больше 0.

- **acl_manager_bgp_received_records** (для сервиса доставки списков)

  Метрика, счётчик которой показывает текущее количество записей, которое получает сервис доставки списков от сервиса goBGP для их последующего обновления. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса доставки списков на узлы фильтрации необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса сравнения списков, сервиса доставки списков и сервиса goBGP значение счётчика этой метрики в момент получения текущего набора записей от сервиса goBGP должно быть больше 0.

- **acl_manager_bgp_send_add_records** (для сервиса доставки списков)

  Метрика, счётчик которой показывает количество добавляемых позиций в текущий набор записей, которые сервис доставки списков отправил сервису goBGP для ретрансляции на узлы фильтрации второго уровня. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса доставки списков на узлы фильтрации необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса сравнения списков, сервиса доставки списков и сервиса goBGP значение счётчика этой метрики в момент

отправки изменений на добавление записей в сервис goBGP должно быть больше 0.

- **acl_manager_bgp_send_del_records** (для сервиса доставки списков)

  Метрика, счётчик которой показывает количество удаляемых позиций из текущего набора записей, которые сервис доставки списков отправил сервису goBGP для ретрансляции этих изменений на узлы фильтрации второго уровня. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса доставки списков на узлы фильтрации необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса сравнения списков, сервиса доставки списков и сервиса goBGP значение счётчика этой метрики в момент отправки изменений на удаление записей в сервис goBGP должно быть больше 0.

- **acl_manager_update_in_progress** (для сервиса доставки списков)

  Метрика, ненулевое значение счётчика которой показывает, что в данный момент сервис доставки списков производит обновление списка. Тип данного счётчика – gauge, и он может принимать в разные моменты времени как нулевое, так и ненулевое значение, поэтому при проведении проверок сервиса доставки списков на узлы фильтрации необходимо отправить несколько запросов этой метрики за короткий интервал времени.

  При правильной работе сервиса сравнения списков, сервиса доставки списков и сервиса goBGP значение счётчика этой метрики в момент обновления списка должно быть равным единице.

Сервисы сравнения списков и сервисы доставки списков состоят из N экземпляров. Каждый экземпляр сервиса сравнения списков генерирует результат сравнения чёрного и белого списка в соответствии с версией протокола IP-адресов

(IPv4 или IPv6). Экземпляры сервиса сравнения списков также предоставляют доступ к чёрным спискам для оборудования первого уровня фильтрации.

Каждый экземпляр сервиса доставки списков запрашивает и получает от сервиса сравнения основных списков набор записей определённого типа и транслирует их в сервис goBGP для распространения на узлах фильтрации второго уровня.

При выявлении проблемы в работе сервисов сравнения списков или сервисов доставки списков необходимо определить, какие именно экземпляры работают неправильно (кроме сервиса goBGP, который работает в единственном экземпляре).

7.4.2. Возможные причины неисправной работы

- экземпляр сервиса сравнения списков находится в нерабочем состоянии, и данные не передаются в сервис доставки списков;
- экземпляр сервиса доставки списков находится в нерабочем состоянии, и сервис goBGP не получает записи для последующей передачи на узлы фильтрации второго уровня;
- неправильная конфигурация экземпляра сервиса сравнения списков, экземпляра сервиса доставки списков или сервиса goBGP;
- нарушение сетевой связности между экземпляром сервиса сравнения списков и экземпляром сервиса доставки списков;
- нарушение сетевой связности между экземпляром сервиса доставки списков и сервисом goBGP;
- несоответствующая версия образа для контейнера с экземпляром сервиса сравнения списков или для контейнера с экземпляром сервиса доставки списков.

7.4.3. Порядок выявления и устранения причин

1. Проверить журнал событий для контейнера с экземпляром сервиса сравнения списков командой **`kubectl logs имя_экземпляра_сервиса -n scos -c имя основного_контейнера_для_экземпляра_сервиса`**.

```
operator@operator-pc# kubectl logs scos-acl-differ-v4  -n scos -c acl-differ-v4
tion refused\"" func="acl-differ/api.(*GRPCServer).List" file="/go/src/acl-
differ/api/grpc.go:139"
time="2020-04-20T16:06:07Z" level=error msg="can't get blacklist, err rpc error:
code = Unavailable desc = all SubConns are in TransientFailure, latest
connection error: connection error: desc = \"transport: Error while dialing dial
tcp 10.210.9.250:30642: connect: connec
tion refused\"" func="acl-differ/api.(*GRPCServer).List" file="/go/src/acl-
differ/api/grpc.go:139"
time="2020-04-20T16:06:12Z" level=error msg="can't get blacklist, err rpc error:
code = Unavailable desc = all SubConns are in TransientFailure, latest
connection error: connection error: desc = \"transport: Error while dialing dial
tcp 10.210.9.250:30642: connect: connec
tion refused\"" func="acl-differ/api.(*GRPCServer).List" file="/go/src/acl-
differ/api/grpc.go:139"
time="2020-04-21T05:00:48Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:01:18Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:01:49Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:02:19Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:02:49Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:03:19Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:03:37Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
time="2020-04-21T05:03:50Z" level=info msg="&{178.47.103.241 ffffffff [{29673
29673}]} &{178.47.96.0 ffffe000 [{1 65535}]} []" func=acl-
differ/differ.SubtractItemLists file="/go/src/acl-differ/differ/differ.go:160"
```

В журнале событий контейнера с экземпляром сервиса сравнения списков за последнее время не должно быть систематически повторяющихся ошибок.

2. Проверить журнал событий для контейнера с экземпляром сервиса доставки списков командой **`kubectl logs имя_экземпляра_сервиса -n scos`**.

```
operator@operator-pc# kubectl logs scos-acl-manager-block-v4 -n scos
time="2020-04-30T22:10:29Z" level=info msg="finish: send del old records"
func="acl-manager/bgp.(*BGP).DelRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:153"
time="2020-04-30T22:10:29Z" level=info msg="start: send new records" func="acl-
manager/bgp.(*BGP).AddRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:89"
time="2020-04-30T22:10:29Z" level=info msg="finish: send new records" func="acl-
manager/bgp.(*BGP).AddRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:112"
time="2020-04-30T22:10:29Z" level=info msg="update: out" func="acl-
manager/manager.(*Manager).update" file="/go/src/acl-
manager/manager/manager.go:184"
time="2020-04-30T22:10:59Z" level=info msg="update: in" func="acl-
manager/manager.(*Manager).update" file="/go/src/acl-
manager/manager/manager.go:167"
time="2020-04-30T22:10:59Z" level=info msg="start processing of lists"
func="acl-manager/manager.(*Manager).processNetworkPrefixFromChannels"
file="/go/src/acl-manager/manager/manager.go:217"
time="2020-04-30T22:10:59Z" level=info msg="acl-differ: connection opened"
func=acl-manager/prefix-getters.Differ.getPrefixesFromACLDifferAndSendToChannel
file="/go/src/acl-manager/prefix-getters/acl differ.go:56"
time="2020-04-30T22:10:59Z" level=info msg="gobgp: connection opened" func=acl-
manager/bgp.getFlowSpecRulesAndSendToChannel file="/go/src/acl-
manager/bgp/bgp.go:416"
time="2020-04-30T22:10:59Z" level=info msg="acl-differ: stream opened" func=acl-
manager/prefix-getters.Differ.getPrefixesFromACLDifferAndSendToChannel
file="/go/src/acl-manager/prefix-getters/acl_differ.go:80"
time="2020-04-30T22:11:00Z" level=info msg="start update of table" func="acl-
manager/manager.(*Manager).sendACLUpdates" file="/go/src/acl-
manager/manager/manager.go:310"
time="2020-04-30T22:11:00Z" level=info msg="start: send del old records"
func="acl-manager/bgp.(*BGP).DelRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:125"
time="2020-04-30T22:11:00Z" level=info msg="finish: send del old records"
func="acl-manager/bgp.(*BGP).DelRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:153"
time="2020-04-30T22:11:00Z" level=info msg="start: send new records" func="acl-
manager/bgp.(*BGP).AddRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:89"
time="2020-04-30T22:11:00Z" level=info msg="finish: send new records" func="acl-
manager/bgp.(*BGP).AddRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:112"
time="2020-04-30T22:11:00Z" level=info msg="update: out" func="acl-
manager/manager.(*Manager).update" file="/go/src/acl-
manager/manager/manager.go:184"
time="2020-04-30T22:11:30Z" level=info msg="update: in" func="acl-
manager/manager.(*Manager).update" file="/go/src/acl-
manager/manager/manager.go:167"
time="2020-04-30T22:11:30Z" level=info msg="start processing of lists"
func="acl-manager/manager.(*Manager).processNetworkPrefixFromChannels"
file="/go/src/acl-manager/manager/manager.go:217"
```

```
time="2020-04-30T22:11:30Z" level=info msg="gobgp: connection opened" func=acl-
manager/bgp.getFlowSpecRulesAndSendToChannel file="/go/src/acl-
manager/bgp/bgp.go:416"
time="2020-04-30T22:11:30Z" level=info msg="acl-differ: connection opened"
func=acl-manager/prefix-getters.Differ.getPrefixesFromACLDifferAndSendToChannel
file="/go/src/acl-manager/prefix-getters/acl_differ.go:56"
time="2020-04-30T22:11:30Z" level=info msg="acl-differ: stream opened" func=acl-
manager/prefix-getters.Differ.getPrefixesFromACLDifferAndSendToChannel
file="/go/src/acl-manager/prefix-getters/acl_differ.go:80"
time="2020-04-30T22:11:30Z" level=info msg="start update of table" func="acl-
manager/manager.(*Manager).sendACLUpdates" file="/go/src/acl-
manager/manager/manager.go:310"
time="2020-04-30T22:11:30Z" level=info msg="start: send del old records"
func="acl-manager/bgp.(*BGP).DelRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:125"
time="2020-04-30T22:11:30Z" level=info msg="finish: send del old records"
func="acl-manager/bgp.(*BGP).DelRoutesFromChannelStream" file="/go/src/acl-
manager/bgp/bgp.go:153"
```

В журнале событий контейнера с экземпляром сервиса доставки списков за последнее время не должно быть систематически повторяющихся ошибок.

3. Проверить логи событий для контейнера с экземпляром сервиса goBGP командой **`kubectl logs имя_экземпляра_сервиса -n scos -c имя_основного_контейнера`**.

```
operator@operator-pc# kubectl logs scos-gobgp -n scos -c gobgp
{"level":"info","msg":"gobgpd started","time":"2020-04-17T18:21:21Z"}
{"Topic":"Config","level":"info","msg":"Finished reading the config
file","time":"2020-04-17T18:21:21Z"}
{"level":"info","msg":"Peer 192.168.250.64 is added","time":"2020-04-
17T18:21:21Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration
for:192.168.250.64","time":"2020-04-17T18:21:21Z"}
{"level":"info","msg":"Peer 10.86.4.52 is added","time":"2020-04-17T18:21:21Z"}
{"Topic":"Peer","level":"info","msg":"Add a peer configuration
for:10.86.4.52","time":"2020-04-17T18:21:21Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"in
fo","msg":"Peer Up","time":"2020-04-17T18:21:26Z"}
{"Key":"10.86.4.52","Reason":"read-
failed","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer
Down","time":"2020-04-21T16:01:41Z"}
{"Key":"10.86.4.52","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info",
"msg":"Peer Up","time":"2020-04-21T16:03:08Z"}
{"Code":6,"Data":null,"Key":"192.168.250.64","Subcode":3,"Topic":"Peer","level":
"warning","msg":"received notification","time":"2020-04-22T12:54:45Z"}
{"Key":"192.168.250.64","Reason":"notification-received code 6(cease) subcode
3(peer
deconfigured)","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg"
:"Peer Down","time":"2020-04-22T12:54:45Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"in
fo","msg":"Peer Up","time":"2020-04-23T19:53:21Z"}
{"Code":6,"Data":null,"Key":"192.168.250.64","Subcode":3,"Topic":"Peer","level":
"warning","msg":"received notification","time":"2020-04-24T10:40:17Z"}
```

{"Key":"192.168.250.64","Reason":"notification-received code 6(cease) subcode 3(peer
deconfigured)","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer Down","time":"2020-04-24T10:40:17Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-04-24T10:58:12Z"}
{"Code":6,"Data":null,"Key":"192.168.250.64","Subcode":3,"Topic":"Peer","level":"warning","msg":"received notification","time":"2020-04-25T07:05:18Z"}
{"Key":"192.168.250.64","Reason":"notification-received code 6(cease) subcode 3(peer
deconfigured)","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer Down","time":"2020-04-25T07:05:18Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-04-25T07:08:02Z"}
{"Code":6,"Data":null,"Key":"192.168.250.64","Subcode":3,"Topic":"Peer","level":"warning","msg":"received notification","time":"2020-04-27T09:59:11Z"}
{"Key":"192.168.250.64","Reason":"notification-received code 6(cease) subcode 3(peer
deconfigured)","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer Down","time":"2020-04-27T09:59:11Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-04-27T10:01:56Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-05-05T10:35:57Z"}
{"Key":"10.86.4.52","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-05-05T10:36:09Z"}
{"Code":6,"Data":null,"Key":"192.168.250.64","Subcode":3,"Topic":"Peer","level":"warning","msg":"received notification","time":"2020-05-06T09:22:36Z"}
{"Key":"192.168.250.64","Reason":"notification-received code 6(cease) subcode 3(peer
deconfigured)","State":"BGP_FSM_ESTABLISHED","Topic":"Peer","level":"info","msg":"Peer Down","time":"2020-05-06T09:22:36Z"}
{"Key":"192.168.250.64","State":"BGP_FSM_OPENCONFIRM","Topic":"Peer","level":"info","msg":"Peer Up","time":"2020-05-06T09:25:19Z"}

В журнале событий контейнера с экземпляром сервиса goBGP за последнее время не должно быть систематически повторяющихся ошибок.

4. Проверить, что экземпляр сервиса сравнения списков находится в рабочем состоянии (STATUS – Running), с помощью команды **`kubectl get pods -n scos | grep имя_экземпляра_сервиса`**.

```
operator@operator-pc# kubectl get pods --namespace=scos | grep acl-differ-v4
NAME                                READY   STATUS    RESTARTS   AGE
scos-acl-differ-v4                  2/2     Running   0          10d
```

5. Проверить, что экземпляр сервиса доставки списков находится в рабочем состоянии (STATUS – Running), с помощью команды **`kubectl get pods -n scos | grep имя_экземпляра_сервиса`**.

```
operator@operator-pc# kubectl get pods --namespace=scos | grep acl-manager-block-v4
```

```
NAME                                  READY   STATUS    RESTARTS   AGE
scos-acl-manager-block-v4             1/1     Running   0          6d22h
```

6. Проверить, что экземпляр сервиса goBGP находится в рабочем состоянии (STATUS – Running), с помощью команды **`kubectl get pods -n scos | grep имя_экземпляра_сервиса`**.

```
operator@operator-pc# kubectl get pods --namespace=scos | grep gobgp
NAME                                  READY   STATUS    RESTARTS   AGE
scos-gobgp                            3/3     Running   0          13d
```

7. Проверить текущие параметры конфигурации экземпляра сервиса сравнения списков командой kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].env"

```
operator@operator-pc# kubectl get pod scos-acl-differ-v4  --namespace=scos -o
json | jq ".spec.containers[].env"
[
  {
    "name": "BLACKLISTRPC",
    "value": "server.scos.ru:30642"
  },
  {
    "name": "WHITELISTRPC",
    "value": " server.scos.ru:30641"
  }
]
[
  {
    "name": "ACL_DIFFER_ADDR",
    "value": "localhost:9090"
  },
  {
    "name": "LISTS",
    "value": "/lists/lists.yaml"
  }
]
```

В конфигурации экземпляра сервиса сравнения списков должны быть заданы правильные значения следующих параметров:

- **BLACKLISTGRPC** и **WHITELISTGRPC**. Данные параметры определяют сетевую связность с сервисами хранения основных списков. Значение **BLACKLISTGRPC** должно содержать правильное DNS-имя (или IP-адрес) и номер порта экземпляра сервиса хранения основного чёрного списка, а значение **WHITELISTGRPC** – правильное DNS-имя (или IP-адрес) и номер

порта экземпляра сервиса хранения основного белого списка. Правильные значения этих параметров можно узнать следующим способом:

Шаг 1. Узнать номер порта и тип сетевой службы для экземпляра сервиса хранения основного черного списка командой **`kubectl get services --namespace=scos | grep scos-acl-list-black-v4`**.

```
operator@operator-pc# kubectl get services --namespace=scos | grep scos-acl-list-black-v4
NAME                        TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)             AGE
scos-acl-list-black-v4      NodePort    10.43.17.16  <none>        30642:30642/TCP     9d
scos-acl-list-black-v4-db   ClusterIP   10.43.36.28  <none>        27017/TCP,9216/TCP  9d
```

Экземпляр сервиса хранения основных списков использует тип сетевой службы **NodePort** и TCP-порт 30642. Сетевая служба типа **NodePort** обеспечивает возможность доступа к экземпляру сервиса из-за пределов кластера.

Шаг.2. Узнать IP-адрес экземпляра сервиса хранения основных списков, который будет равен любому IP-адресу узла кластера. Узнать IP-адреса всех узлов кластера можно с помощью команды **`kubectl get nodes -o jsonpath='{ $.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'`**.

```
operator@operator-pc# kubectl get nodes -o jsonpath='{
$.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'
192.168.100.10 192.168.100.11 192.168.100.12 192.168.100.13
```

Вместо IP-адреса можно использовать DNS-имя одного из узлов кластера.

```
nslookup 192.168.100.10
Server:          192.168.100.2
Address:    192.168.100.2#53

10.100.168.192.in-addr.arpa   name = server.scos.ru.
```

Шаг 3. Повторить шаги 1 и 2 для получения правильного значения для основного сервиса хранения белого списка

В приведённом выше примере правильными значениями параметров **BLACKLISTGRPC** и **WHITELISTGRPC** экземпляра сервиса сравнения списков являются:

- **server.scos.ru** (или **192.168.100.10**) и **30642** для параметра **BLACKLISTGRPC**;
- **server.scos.ru** (или **192.168.100.10**) и **30641** для параметра **WHITELISTGRPC**.

- **ACL_DIFFER_ADDR.** Значение данного параметра должно быть по умолчанию **localhost:9090**.
- **LISTS.** Значение данного параметра должно быть по умолчанию **/lists/lists.yaml**.

При необходимости следует задать правильные значения параметров экземпляра сервиса сравнения списков в файле со значениями чарта (**values.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

8. Проверить текущие параметры конфигурации экземпляра сервиса доставки списков командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[].env"`**.

```
operator@operator-pc# kubectl get pod scos-acl-manager-block-v4 --namespace=scos
-o json | jq ".spec.containers[].env"
[
  {
    "name": "IP_MODE",
    "value": "IPv4"
  },
  {
    "name": "PREFIX_SOURCE",
    "value": "acl-differ"
  },
  {
    "name": "PREFIX_SOURCE_API",
    "value": "server.scos.ru:30940"
  },
  {
    "name": "GOBGP_GRPC_ADDR",
    "value": "server.scos.ru:50051"
  },
  {
    "name": "GOBGP_VRF_NAME",
    "value": "block"
```

```
  },
  {
    "name": "BLACK_INCLUDE_TAGS",
    "value": "tls,hrandom"
  },
  {
    "name": "BLACK_EXCLUDE_TAGS",
    "value": "rkn,static"
  },
  {
    "name": "WHITE_INCLUDE_TAGS",
    "value": "static"
  },
  {
    "name": "WHITE_EXCLUDE_TAGS"
  }
]
```

В конфигурации экземпляра сервиса доставки списков должны быть заданы правильные значения следующих параметров:

- **IP_MODE.** Параметр указывает на версию IP (IPv4 или IPv6) и определяет формат представления записей в списке. Значение данного параметра должно соответствовать значению в конфигурации данного экземпляра сервиса доставки списков;

- **PREFIX_SOURCE.** Значение данного параметра указывает на источник получения записей и должно соответствовать значению в конфигурации данного экземпляра сервиса доставки списков;

- **PREFIX_SOURCE_API.** Значение данного параметра определяет сетевую связность с экземпляром сервиса сравнения списков. Правильное значение этого параметра можно получить следующим способом:

  Шаг 1. Определить номер порта и тип сетевой службы для экземпляра сервиса сравнения списков командой **`kubectl get services --namespace=scos | scos-acl-differ-v4-grpc`**.

```
operator@operator-pc# kubectl get services --namespace=scos  | scos-acl-differ-
v4-grpc
NAME                      TYPE     CLUSTER-IP   EXTERNAL-IP PORT(S)           AGE
scos-acl-differ-v4-grpc   NodePort 10.43.19.39  <none>   30940:30940/TCP      10d
```

Экземпляр сервиса сравнения списков использует сетевую службу типа **NodePort** и TCP-порт 30940. Сетевая служба **NodePort** обеспечивает возможность доступа к экземпляру сервиса из-за пределов кластера.

Шаг 2. Узнать IP-адрес экземпляра сервиса сравнения списков, который будет равен любому IP-адресу узла кластера. Узнать IP-адреса всех узлов кластера можно с помощью команды **`kubectl get nodes -o jsonpath='{ $.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'.`**

```
operator@operator-pc# kubectl get nodes -o jsonpath='{
$.items[*].status.addresses[?(@.type=="InternalIP")].address }{"\n"}'
192.168.100.10 192.168.100.11 192.168.100.12 192.168.100.13%
```

Вместо IP-адреса можно использовать DNS-имя одного из узлов кластера.

```
nslookup 192.168.100.10
Server:          192.168.100.2
Address:    192.168.100.2#53

10.100.168.192.in-addr.arpa   name = server.scos.ru.
```

- **GOBGP_GRPC_ADDR.** Значение данного параметра определяет сетевую связность с экземпляром сервиса goBGP. Правильное значение этого параметра можно узнать следующим способом:

  Шаг 1. Сервис goBGP взаимодействует непосредственно с узлами фильтрации, поэтоу ему требуется прямой сетевой доступ из-за пределов кластера. Данный тип доступа обеспечивается специальнам режимом работы **hostNetwork**, который позволяет сделать привязку сервиса goBGP непосредственно к сетевой службе одного из хостов кластера. Необходимо проверить факт запуска сервиса goBGP в требуемом режиме **hostNetwork** командой **`kubectl get pod scos-gobgp  --namespace=scos -o json | jq ".spec.hostNetwork"`**.

```
operator@operator-pc# kubectl get pod scos-gobgp  --namespace=scos -o json | jq
".spec.hostNetwork"
true
```

Шаг 2. Необходимо определить DNS-имя узла кластера, на которой развернут экземпляр сервиса goBGP, и сеть, которую он использует. Для этого необходимо отправить команду **`kubectl get pod scos-gobgp --namespace=scos -o json | jq ".spec.nodeName"`**.

```
operator@operator-pc# kubectl get pod scos-gobgp --namespace=scos -o json | jq
".spec.nodeName"
"server"
```

Шаг 3. Необходимо определить порт, на который сервис goBGP принимает записи от сервиса доставки списка по протоколу GRPC с помощью команды **`kubectl get pod scos-gobgp --namespace=scos -o json | jq ".spec.containers[0].ports[1]"`**.

```
operator@operator-pc# kubectl get pod scos-gobgp --namespace=scos -o json | jq
".spec.containers[0].ports.[1]"
  {
    "containerPort": 50051,
    "hostPort": 50051,
    "name": "grpc",
    "protocol": "TCP"
  }
```

Таким образом, в данном примере правильным значением проверяемого параметра **GOBGP_GRPC_ADDR** является **server.scos.ru:50051**.

- **GOBGP_VRF_NAME.** Данный параметр определяет название VRF таблицы сервиса goBGP, в которую в итоге будут помещены записи сформированного списка. Значение данного параметра должно соотстветствовать конфигурации данного экземпляра сервиса.

- **BLACK_INCLUDE_TAGS.** Данный параметр определяет типы записей чёрного списка, запрашиваемые данным экземпляром сервиса доставки списков. Значение данного параметра должно соотстветствовать конфигурации данного экземпляра сервиса.

- **BLACK_EXCLUDE_TAGS.** Данный параметр определяет типы записей чёрного списка, не запрашиваемые данным экземпляром сервиса доставки

списков. Значение данного параметра должно соотстветствовать конфигурации данного экземпляра сервиса.

- **WHITE_INCLUDE_TAGS.** Данный параметр определяет типы записей белого списка, запрашиваемые данным экземпляром сервиса доставки списков. Значение данного параметра должно соотстветствовать конфигурации данного экземпляра сервиса.

- **WHITE_EXCLUDE_TAGS.** Данный параметр определяет типы записей белого списка, не запрашиваемые данным экземпляром сервиса доставки списков. Значение данного параметра должно соотстветствовать конфигурации данного экземпляра сервиса.

При необходимости следует задать правильные значения параметров экземпляра сервиса хранения основных списков в файле с информацией о чарте (**values.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

9. Проверить текущий набор параметров конфигурации экземпляра сервиса goBGP с учётом того, что в качестве метода получения настроек используется ConfigMap. Для этого следует отправить команду **`kubectl get configmaps scos-gobgp -n scos -o json | jq ".data"`**.

```
operator@operator-pc# kubectl get configmaps scos-gobgp -n scos -o json | jq
".data"
{
"config.yml": "global:\n  config:\n    port: 179\n    as: 64512\n    router-id:
192.168.100.10\n  apply-policy:\n    config:\n      import-policy-list:\n
- scos_policy\nneighbors:\n  - config:\n      neighbor-address: 192.168.200.10\n
peer-as: 64512\n    afi-safis:\n      - config:\n          afi-safi-name:
\"ipv4-flowspec\"\n      - config:\n          afi-safi-name: \"ipv6-flowspec\"\n
- config:\n          afi-safi-name: \"l3vpn-ipv4-flowspec\"\n      - config:\n
afi-safi-name: \"l3vpn-ipv6-flowspec\"\n  - config:\n      neighbor-address:
192.168.250.64\n      peer-as: 64512\n    afi-safis:\n      - config:\n
afi-safi-name: \"ipv4-flowspec\"\n      - config:\n          afi-safi-name:
\"ipv6-flowspec\"\n      - config:\n afi-safi-name: \"l3vpn-ipv4-flowspec\"\n
- config:\n          afi-safi-name: \"l3vpn-ipv6-flowspec\"\n  - config:\n
neighbor-address: 10.86.4.52\n      peer-as: 64512\n    afi-safis:\n  -
config:\n          afi-safi-name: \"ipv4-flowspec\"\n      - config:\n
afi-safi-name: \"ipv6-flowspec\"\n      - config:\n          afi-safi-name:
\"l3vpn-ipv4-flowspec\"\n      - config:\n          afi-safi-name: \"l3vpn-ipv6-
flowspec\"\nvrfs:\n  - config:\n      name: block\n  id: 1\n      rd: 64512:10\n
both-rt-list:\n        - 64512:10\n  - config:\n      name: redirect\n      id:
```

```
2\n      rd: 64512:20\n      both-rt-list:\n        - 64512:20\n  - config:\n
name: rknip\n      id: 3\n      rd: 64512:30\n      both-rt-list:\n        -
64512:30\n  - config:\n  name: rknport\n      id: 4\n      rd: 64512:40\n
both-rt-list:\n        - 64512:40\ndefined-sets:\n  neighbor-sets:\n    -
neighbor-set-name: ns_scos_self\n      neighbor-info-list:\n        -
192.168.100.10\n    - neighbor-set-name: ns_all_neighbors\n      neighbor-info-
list:\n        - 0.0.0.0/0\npolicy-definitions:\n  - name: scos_policy\n
statements:\n      - name: statement1\n    conditions:\n          match-
neighbor-set:\n    neighbor-set: ns_scos_self\n            match-set-options:
any\n          bgp-conditions:\n    afi-safi-in-list:\n  - ipv4-flowspec\n
- ipv6-flowspec\n              - l3vpn-ipv4-flowspec\n              - l3vpn-
ipv6-flowspec\n        actions:\n          route-disposition: accept-route\n
- name: statement2\n        conditions:\n  match-neighbor-set:\n     neighbor-
set: ns_all_neighbors\n match-set-options: any\n       bgp-conditions:\n
afi-safi-in-list:\n              - ipv4-flowspec\n              - ipv6-
flowspec\n              - l3vpn-ipv4-flowspec\n              - l3vpn-ipv6-
flowspec\n      actions:\n route-disposition: reject-route"
}
```

После проведения проверки при необходимости следует задать правильные значения параметров экземпляра сервиса goBGP в исходном файле (**configmap.yaml**) и обновить сервис с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

10. Проверить сетевую связность между сервисом сравнения списков и сервисом хранения списков. Для этого необходимо зайти в оболочку экземпляра сервиса сравнения списков с помощью команды **`kubectl exec -it scos-acl-differ-v4 --namespace=scos sh`**

```
operator@operator-pc# kubectl exec -it scos-acl-differ-v4  --namespace=scos  sh
/srv #
```

проверить доступность экземпляра сервиса хранения чёрных списков по прослушиваемому порту с помощью общедоступных системных инструментов: **nmap**, **netcat**:

```
/srv # nmap server.scos.ru -p 30642
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
30642/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

и проверить доступность экземпляра сервиса хранения белых списков по прослушиваемому порту с помощью общедоступных системных инструментов (**nmap**, **netcat**):

```
/srv # nmap server.scos.ru -p 30641

Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
30641/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

При необходимости устранить нарушение сетевой связности между экземпляром сервиса сравнения списков и экземплярами сервисов хранения списков.

11. Проверить сетевую связность между сервисом доставки списков и сервисом сравнения списков. Для этого необходимо зайти в оболочку экземпляра сервиса доставки списков на узлы фильтрации с помощью команды **`kubectl exec -it scos-acl-manager-block-v4 --namespace=scos sh`**

```
operator@operator-pc# kubectl exec -it scos-acl-manager-block-v4  --
namespace=scos  sh
/srv #
```

проверить доступность экземпляра сервиса сравнения списков по прослушиваемому порту с помощью общедоступных системных инструментов (**nmap**, **netcat**):

```
/srv # nmap server.scos.ru -p 30940
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
30940/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.73 seconds
```

При необходимости устранить нарушение сетевой связности между экземпляром сервиса доставки списков и сервисом сравнения списков.

12. Проверить сетевую связность между сервисом доставки списков и экземпляром сервиса goBGP. Для этого необходимо зайти в оболочку экземпляра сервиса доставки списков с помощью команды **`kubectl exec -it scos-acl-manager-block-v4 --namespace=scos sh`**

```
operator@operator-pc# kubectl exec -it scos-acl-manager-block-v4  --
namespace=scos  sh
/srv #
```

проверить доступность экземпляра сервиса goBGP по прослушиваемому порту с помощью общедоступных системных инструментов (**nmap**, **netcat**):

```
/srv # nmap server.scos.ru -p 50051
Starting Nmap 7.70 ( https://nmap.org ) at 2020-04-29 20:06 UTC
Nmap scan report for server.scos.ru (192.168.100.10)
Host is up (0.00025s latency).
rDNS record for 192.168.100.2: scos-dns-unbound.scos-dns.svc.cluster.local

PORT      STATE SERVICE
50051/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.70 seconds
```

При необходимости устранить нарушение сетевой связности между экземпляром сервиса доставки списков и экземпляром сервиса goBGP.

13. Проверить версию образа работающего экземпляра сервиса сравнения списков командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[0].image"`**.

```
operator@operator-pc# kubectl get pod scos-acl-differ-v4 --namespace=scos -o
json | jq ".spec.containers[0].image"
"hub.scos.ru/acl-differ:v3.0.1"
```

Проверить в файле с информацией о чарте (**chart.yaml**) соответствие версии образа для экземпляра сервиса сравнения списков, при необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

14. Проверить версию образа работающего экземпляра сервиса доставки списков командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[0].image"`**.

```
operator@operator-pc# kubectl get pod scos-acl-manager-block-v4 --namespace=scos
-o json | jq ".spec.containers[0].image"
"hub.scos.ru/acl-manager:v2.4.2"
```

Проверить в файле с информацией о чарте (**chart.yaml**) соответствие версии образа для экземпляра сервиса доставки списков, при необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.

15. Проверить версию образа работающего экземпляра сервиса goBGP командой **`kubectl get pod имя_экземпляра_сервиса --namespace=scos -o json | jq ".spec.containers[0].image"`**.

```
operator@operator-pc# kubectl get pod scos-gobgp  --namespace=scos -o json | jq
".spec.containers[0].image"
"hub.scos.ru/gobgp:v2.15.0"
"alpine:3.10"
```

Проверить, что в файле с информацией о чарте (**chart.yaml**) указана правильная версия образа для экземпляра сервиса goBGP. При необходимости изменить версию образа и обновить экземпляр сервиса с помощью установщика пакетов **helm**, используя ключевое слово **`--upgrade`**.