servicepipe

Документация по FlowCollector

1.6 / spider-4.7

СГЕНЕРИРОВАНО 26.03.2026, 07:45

flowcollector 1.6 / spider-4.7

Содержание

FlowCollector

FlowCollector — интеллектуальная система анализа потокового трафика, предназначенная для раннего выявления DDoS-атак и сетевых аномалий. Обеспечивает высокоскоростную агрегацию и обработку потоковых данных, выявляя подозрительную активность и автоматически инициируя защитные меры. Используется как самостоятельный детектор или как часть комплексной системы с DosGate.

Архитектура FlowCollector

FlowCollector реализует три ключевых направления работы:

- приём и агрегация потоковых данных (NetFlow v10 (IPFIX), NetFlow v9, sFlow, IMON);
- пороговая детекция трафика с применением векторных фильтров;
- формирование управляющих действий при обнаружении аномалий.

Потоковые записи обрабатываются в режиме реального времени. На основе указанных порогов и векторов производится вычисление нагрузки по IP-адресам, подсетям и направлениям. При превышении порогов система может сформировать маршрут очистки и направить трафик в защитный контур (например, через DosGate), а также зафиксировать событие для отчётности и анализа.

Модуль Data Explorer — хранит ретроспективные данные о трафике и событиях, обеспечивает быстрый поиск и анализ для последующего аудита или отладки порогов.

Возможности системы:

- **Детекция и противодействие DDoS-атакам**
  FlowCollector выявляет аномалии сетевого трафика на ранней стадии. В зависимости от конфигурации, возможны следующие действия: перенаправление трафика на системы фильтрации, генерация BGP FlowSpec или BGP Blackhole.
- **Поддержка потоковых протоколов**
  Поддерживает NetFlow v10 (IPFIX), NetFlow v9, sFlow, IMON и анализ

зеркалированного трафика. Удобно встраивается в инфраструктуру любого уровня.

- **Высокопроизводительная обработка трафика**
  Одна нода обрабатывает до 250 000 flow в секунду при минимальной задержке.
- **Интеграция с системой фильтрации**
  При детектировании аномалии FlowCollector может перенаправлять трафик в DosGate с помощью отправки BGP анонсов для фильтрации DDoS-атаки. Управление DosGate и FlowCollector осуществляется через единый веб-интерфейс.
- **Аналитика и ретроспективный анализ**
  FlowCollector формирует и сохраняет отчёты с подробной информацией по каждой детектированной аномалии. С помощью модуля Data Explorer доступен полный набор метаданных из потоков NetFlow v10 (IPFIX), NetFlow v9, sFlow и IMON для ретроспективного анализа сетевой активности.
- **Техподдержка 24/7/365**
  Клиенты получают круглосуточную поддержку с реакцией до 5 минут по SLA.

Что нового в FlowCollector

Раздел содержит изменения в системе FlowCollector. Обновления веб-интерфейса SP Spider вынесены отдельно.

- История версий SP Spider

Версия 1.6

Дата релиза: 30.09.2025

Новые возможности

FlowSpec-правила
Добавлены предустановленные FlowSpec-правила — механизм распространения фильтрационных политик на маршрутизаторы через BGP Flow Specification.

Фильтр источников по ifIndex для flow-протоколов
Точная фильтрация источников по интерфейсу.

Поддержка NetFlow v9
Приём и обработка потоков NetFlow v9.

Поддержка flow протокола IMON
Добавлена совместимость с протоколом IMON.

Опция disable для источников
Возможность отключить источник трафика на уровне конфигурации.

Улучшения

- Сделан исполняемым по умолчанию скрипт *bin/create_hugepages.sh*.
- Исключена запись *if_rx_octets* в режиме *save-flow*.

- В API **/managed-object/config** добавлен статус для каждого профиля.
- Параметр *rate* перенесён из **packet-counters** в секции **dpdk** и **ipfix**.
- Реализовано автоматическое переподключение к хранилищу потоков ClickHouse.

Версия 1.5

Дата релиза: 30.06.2025

Новые возможности

Автоматическое объединение локальных аномалий
Локальные аномалии могут объединяться в глобальную при выполнении заданных критериев.

Поддержка sFlow
Приём и обработка sFlow.

Авторизация к ClickHouse по логину и паролю
Поддержка user/pass в подключениях.

IPFIX: поля переменного размера
Поддержка variable-size полей.

Накопительная валидация опций в YAML
Ошибки по нескольким опциям собираются и показываются вместе.

Новые системные метрики по потокам
Добавлены группы метрик для `lcore-flow-rx/*`, `lcore-flow-worker/*`, а также агрегаты `flow/*`, `flow-ipfix/*`, `flow-sflow/*` (приём, обработка, передача, дропы).

Исправления

- Пропуск некорректных файлов профилей при старте. Повреждённые файлы не блокируют запуск.
- Корректное завершение детекций при остановке *DetectionService*.

- Удалён неиспользуемый параметр *metrics.interval*.
- Унифицированы значения временных интервалов в конфигурации.
- Удалены устаревшие ветки системных метрик: `ipfix-packets-pool/*`, `ipfix-packets-ring/*`, `lcore-rx-ipfix/*`, `lcore-worker-ipfix/*`.

Версия 1.4

Дата релиза: 11.03.2025

Новые возможности

Поддержка BGP FlowSpec
Реализована генерация событий BGP FlowSpec в ответ на обнаружение сетевых аномалий, что позволяет автоматически блокировать порты, используемые в атаках.

Функционал Router ID
Добавлена поддержка Router ID для группировки источников NetFlow с общими или индивидуальными счетчиками, что предотвращает ошибки, вызванные обработкой дублирующихся потоков.

Хранение SNMP в ClickHouse
Теперь данные SNMP сохраняются в ClickHouse рядом с NetFlow и используются в Data Explorer для классификации интерфейсов и анализа направлений сетевого трафика.

Поддержка IPv6 для ICMP Flood
Добавлена полная поддержка IPv6 для детекции атаки ICMP-flood.

Интеграция с NATS
Реализована поддержка NATS для передачи информации и событий о сетевых аномалиях, что повысило производительность системы.

Новые векторы детекции аномалий

- Детекция атаки HTTPS-flood
- Детекция атаки GRE-flood

Улучшения производительности

Запись метрик напрямую в ClickHouse
Метрики теперь записываются непосредственно в ClickHouse, что значительно повысило производительность параллельной записи метрик и статистики.

Оптимизация механизма детекции аномалий
Улучшена общая производительность системы, протестирована обработка атак на 2 миллиона IP-адресов одновременно.

Оптимизация обработки NetFlow

- Все входящие потоки NetFlow теперь сохраняются в ClickHouse для анализа в Data Explorer.
- Реализовано сжатие NetFlow-данных старше 24 часов для экономии места.
- Добавлена возможность произвольного поиска данных в SP Spider.

Улучшение обработки зеркалированного трафика
Общие оптимизации производительности.

Автоматическое распределение нагрузки по CPU
Оптимизирована работа сервиса Analyzer, теперь ресурсы автоматически распределяются между всеми ядрами процессора.

Повышение эффективности записи отчетов
Увеличена скорость записи отчетов и точность фиксируемых данных.

Исправления ошибок и мелкие улучшения

- Исправлена ошибка зависания маршрутов GoBGP.
- Поддержка 32-битных значений в IPFIX для совместимости со старыми или менее продвинутыми маршрутизаторами.
- Добавлены новые системные метрики: В FlowCollector появились метрики для отладки, включая количество экспортируемых метрик и число защищаемых хостов под анализом.
- Мелкие исправления ошибок и улучшения удобства использования.

Версия 1.0

Ручное создание дампов трафика
Добавлена возможность создавать дампы трафика в режиме port mirroring через API и Telegram-бота.

Новый формат оповещений Telegram и SMTP
Оповещения содержат ссылку на отчёт по аномалии.

Оптимизация производительности
Внесены общие улучшения работы системы.

Исправления ошибок
Исправлены выявленные ошибки.

Версия 0.6.94

Отчёты по аномалиям
Добавлен дашборд reports с подробными отчётами по каждой аномалии.

Интеграция с GoBGP
Добавлена поддержка отправки анонсов через GoBGP.

Автоматическая запись дампов трафика
Реализована запись дампов при аномалии в режиме port mirroring.

Новый формат Telegram-оповещений
Поддержана отправка дампов сразу после записи.

Дашборд Top-12 IP addresses
Добавлен новый дашборд по ведущим IP-адресам.

Версия 0.6.9

Поддержка Mellanox
Добавлена совместимость с сетевыми картами Mellanox.

Увеличение производительности
Общая оптимизация работы системы.

Версия 0.6.8

Отчёты по аномалиям
Добавлена генерация отчётов по каждой аномалии с детальной информацией о трафике.

SMTP-оповещения о начале аномалии
Добавлена поддержка SMTP-уведомлений.

Установка FlowCollector на Ubuntu 22.04

В зависимости от выбранного варианта интеграции требования к аппаратной платформе, процессу установки и уровню производительности могут различаться. Рекомендуется следовать инструкциии на всех этапах установки. В случае возникновения вопросов обращайтесь к поставщику программного обеспечения.

Актуальная версия FlowCollector: `1.6`

1. Аппаратные и программные требования

- **Операционная система:** Ubuntu Server 22.04.2 LTS
  Использование других версий или дистрибутивов не гарантирует корректную работу ПО.
- **Процессор:** Intel Xeon, не менее 10 физических ядер.
  Допускается также использование процессоров AMD EPYC с сопоставимой производительностью. Использование других процессоров сторонних производителей не рекомендуется и не поддерживается.
- **Оперативная память:** не менее 8 ГБ
- **Дисковое пространство:** не менее 100 ГБ
- **Сетевые интерфейсы:**
  - Для режима зеркалирования (DPDK): минимум 2 физических интерфейса:
    - Управляющий интерфейс
    - Интерфейс для обработки трафика (должен поддерживаться DPDK).
      Интерфейс резервируется полностью для FlowCollector, его использование для других целей невозможно.

    - Для режима NetFlow: минимум 1 сетевой интерфейс.
- **Рекомендуемые сетевые адаптеры:** Mellanox (mlx5 или mlx6), Intel X520-DA2
- **Права пользователя:** Требуется пользователь с правами *sudo* для запуска установочных скриптов.

1.1. Выбор и подготовка режима интеграции

Режим зеркалирования (Port Mirror, DPDK)

- Рекомендуется для производительных конфигураций и скоростей обработки выше 30 Gbps.
- Требует выделения отдельного физического интерфейса для обработки трафика (режим DPDK).
- На сетевом оборудовании необходимо настроить зеркалирование трафика (port-mirror) с использованием GRE-туннеля и необходимого коэффициента (например, 1:1000).
- После настройки зеркалирования интерфейс будет полностью использоваться FlowCollector.

Пример настройки зеркалирования трафика

```
# Port mirroring
port-mirroring
mirror-once;
input {
    rate 1000;
    run-length 0;
}

instance {
    flowcollector {
        input {
            rate 250;
            run-length 0;
        }
        family inet {
            output {
                interface gr-0/0/0.15 {
                    next-hop 172.20.5.2;
            }
```

```
        }
    }

gr-0/0/0.15
description --Tunnel-to-flowcollector;
tunnel {
    source IP;
    destination IP;
}
family inet {
    address 172.20.5.1/30;
}

firewall filter flowcollector-input

term default {
    then {
        port-mirror-instance flowcollector;
        accept;
    }
}

ae0.2020
description UPSTREAM;
vlan-id 2020;
family inet {
    filter {
        input flowcollector-input;
    }
    sampling {
        input;
    }
    address 10.12.0.2/30;
}
```

Режим потоковых данных

Используется для семплирования трафика. Поддерживает несколько потоковых протоколов:

- **NetFlow v10 (IPFIX)**
- **NetFlow v9**
- **sFlow**
- **IMON**

Для работы потока настройте семплирование пакетов и направьте их на выделенный сетевой интерфейс FlowCollector.

1.2. Подготовка сетевого оборудования

Для режима DPDK:

- Настройте зеркалирование пакетов на уровне сетевого оборудования в сторону выделенного интерфейса.
- Проверьте, что драйвер выбранного сетевого адаптера поддерживается DPDK.

Для режима NetFlow v10 (IPFIX):

- Настройте отправку семплированных пакетов на соответствующий сетевой интерфейс.

1.3. Подготовка аппаратной платформы

- Установите рекомендованную для запуска ОС: Ubuntu Server 22.04 LTS
  **В случае использования других ОС - успешный запуск ПО не гарантирован*
- Используйте рекомендованные CPU: только Intel Xeon
  **В случае использования CPU сторонних производителей (например, AMD) - успешный запуск ПО не гарантирован*
- Используйте рекомендованные сетевые карты: Mellanox (mlx5 или mlx6) или Intel (X520-DA2)

2. Подготовка операционной системы

2.1 Установка обновлений ОС

Для обновления ОС Ubuntu необходимо выполнить следующие команды:

```
sudo apt update
```

```
sudo apt upgrade
```

2.2 Подключение репозитория Servicepipe

Подключить репозиторий Servicepipe возможно двумя способами: через скрипт или вручную. Для подключения к репозиторию потребуются логин и пароль. Эти учетные данные предоставляются индивидуально для каждого заказчика. Получить их возможно запросив у вендора (Servicepipe или партнёра).

2.3 Настройка системного логирования

Открыть файл **/etc/systemd/journald.conf**:

```
sudo nano /etc/systemd/journald.conf
```

Раскомментировать и задать параметры:

```
SystemMaxUse=500M
RuntimeMaxUse=200M
MaxRetentionSec=1day
```

Перезапустить службу:

```
sudo systemctl restart systemd-journald
```

Открыть файл **/etc/logrotate.d/rsyslog**:

```
sudo nano /etc/logrotate.d/rsyslog
```

Рекомендуемая конфигурация:

```
/var/log/syslog
/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
/var/log/auth.log
/var/log/user.log
/var/log/lpr.log
```

```
/var/log/cron.log
/var/log/debug
/var/log/messages
{
        rotate 2
        size 500M
        missingok
        notifempty
        compress
        delaycompress
        sharedscripts
        postrotate
                /usr/lib/rsyslog/rsyslog-rotate
        endscript
}
```

2.1.1 Подключение с помощью скрипта

Выполнить скрипт для автоматической настройки репозитория:

```
curl -o "./setup-repo.sh" "https://public-
repo.svcp.io/setup_script/setup-repo.sh" && \
    sudo chmod +x "./setup-repo.sh" && \
    sudo ./setup-repo.sh
```

При запуске скрипта потребуется ввести логин и пароль. После ввода учетных данных скрипт выполнит все необходимые действия автоматически. В случае некорректной работы скрипта рекомендуется использовать метод ручной настройки репозитория.

2.1.2 Подключение вручную

Добавить ключ:

```
sudo wget --http-user=[ваш логин] --http-password=[ваш пароль] -O
- https://public-repo.svcp.io/keyFile | \
            sudo gpg --dearmor -o
/etc/apt/keyrings/servicepipe.gpg
```

Добавить репозиторий:

```
echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/servicepipe.gpg]
https://public-repo.svcp.io/ubuntu/ xenial contrib" >
/etc/apt/sources.list.d/servicepipe.list
```

Добавить авторизационные данные:

```
echo 'machine public-repo.svcp.io login [ЛОГИН] password [ПАРОЛЬ]'
> /etc/apt/auth.conf
```

Проверить доступность репозитория:

```
sudo apt update
```

3 Установка компонентов FlowCollector

3.1 Состав метапакетов

Установка FlowCollector включает четыре метапакета:

Метапакет	Компоненты
flowcollector Основные компоненты	• analyzer • sp-events • spider-only (включает свои зависимости, см. ниже) • dpdk
spider-only (в составе flowcollector)	• nodejs • libpq-dev • postgresql • rabbitmq-server • sp-spider-broker • sp-spider
flowcollector-additional Сторонние зависимости	• curl • postgresql • mongodb-org • clickhouse-server • clickhouse-client • nats-server • natscli • nginx • libpq-dev • libatomic1

Метапакет	Компоненты
	• zlib1g-dev • libpcap-dev • libnuma-dev • libssl-dev • libbpf-dev • libfdt-dev • libisal-dev • libibverbs-dev • ibverbs-providers • libprotobuf-dev • libgrpc++-dev • protobuf-compiler • protobuf-compiler-grpc • libsnmp-dev
flowcollector-monitoring Компоненты мониторинга	• carbon-clickhouse • graphite-clickhouse • carbonapi

Примечание
Пакет **flowcollector-monitoring** можно не устанавливать, если система мониторинга развернута на отдельном сервере.

3.2 Установка компонентов

Выполнить следующую команду:

```
sudo NEEDRESTART_MODE=a apt-get install -y \
    flowcollector-additional \
    flowcollector \
    flowcollector-monitoring
```

3.3 Расчёт и настройка Hugepages

Hugepages - это крупные страницы памяти. Используются для повышения производительности при обработке большого объёма сетевых данных и оптимизации работы с памятью.

Количество hugepages вычисляются по формуле:

> H = (R x 1024) / (P x S)

- **H** - количество hugepages

- **R** - объем RAM, выделяемый для FlowCollector (в GB)
- **P** - размер страницы (фиксированное значение 2048 kB)
- **S** - количество NUMA-узлов (определяется командой): `ls -d /sys/devices/system/node/node* | wc -l`
- Коэффициент 1024 используется для перевода GB в MB.

Пример для двух нод NUMA:

- RAM: 160 GB (из них выделяемые для FlowCollector - 80 GB)
- Sockets: 2

H = 80 GB x 1024 / (2 MB x 2 sockets) = 20480

Открыть текстовый редактор для создания скрипта:

```
sudo nano /opt/spfc/bin/create_hugepages.sh
```

Вставить следующий код в открытый файл:

```
#!/bin/bash

mkdir -p /dev/hugepages
mountpoint -q /dev/hugepages || mount -t hugetlbfs nodev
/dev/hugepages

#node 0 (CPU 0)
echo 20480 | sudo tee
/sys/devices/system/node/node0/hugepages/hugepages-
2048kB/nr_hugepages

#node 1 (CPU 1)
echo 20480 | sudo tee
/sys/devices/system/node/node1/hugepages/hugepages-
2048kB/nr_hugepages
```

Пример для одной ноды NUMA:

- RAM: 16 GB (из них выделяемые для FlowCollector - 12GB)
- Sockets: 1

H = 12 GB x 1024 / (2 MB x 1 sockets) = 6144

Открыть текстовый редактор для создания скрипта:

```
sudo nano /opt/spfc/bin/create_hugepages.sh
```

Скопировать и вставить следующий код в открытый файл:

```
#!/bin/bash

mkdir -p /dev/hugepages
mountpoint -q /dev/hugepages || mount -t hugetlbfs nodev
/dev/hugepages

#node 0 (CPU 0)
echo 6144 | sudo tee
/sys/devices/system/node/node0/hugepages/hugepages-
2048kB/nr_hugepages
```

Внимание!
В настоящий момент проводится тестирование различных конфигураций, и формула расчёта количества hugepages может быть скорректирована. Для получения актуальных рекомендаций по настройке hugepages рекомендуется обращаться к специалистам команды Servicepipe.

3.4 Настройка Clickhouse

Clickhouse — это высокопроизводительная аналитическая колоночная СУБД, используемая для хранения, обработки и анализа больших объёмов данных в реальном времени. В рамках работы FlowCollector, *clickhouse* предназначен для хранения метрик, flow-данных и справочной информации, а также для обеспечения быстрого доступа к аналитическим данным.

1. Запустить службу *clickhouse-server* и проверить её состояние на наличие ошибок:

```
sudo systemctl start clickhouse-server && systemctl status
clickhouse-server
```

2. Для корректной работы FlowCollector требуется создать следующие таблицы в Clickhouse:

- *graphite* — метрики;
- *graphite_index* — индексы метрик;
- *graphite_tagged* — теги graphite;
- *flows* — таблица для хранения flow-данных;
- *asn_dict* — справочник ASN (для SP-Spider Explorer);
- *cidr_location_dict* — справочник CIDR-локаций (для SP-Spider Explorer);
- *geo_name_dict* — справочник географических названий (для SP-Spider Explorer);
- *flows_fast_dataset* — основная таблица для обработки flows;
- *flows_fast_dataset_mv* — материализованное представление fast_dataset;
- *flows_full_dataset* — полная таблица для хранения flows;
- *flows_full_dataset_mv* — материализованное представление full_dataset.

Примечание
Если при установке *clickhouse* был установлен пароль для пользователя, использовать соответствующую команду с параметром `--password`. Если пароль не задавался, выполнять команду без этого параметра.

Clickhouse без пароля:

```
clickhouse-client --multiline --multiquery <
/usr/share/doc/clickhouse-server/graphite/fc-init.sql
```

Clickhouse с паролем:

```
clickhouse-client --multiline --multiquery --password=[пароль
clickhouse] < /usr/share/doc/clickhouse-server/graphite/fc-
init.sql
```

3. Проверить, что все необходимые таблицы созданы (ожидается 11 таблиц):

Clickhouse без пароля:

```
clickhouse-client --query="SHOW TABLES" | wc -l
```

Clickhouse с паролем:

```
clickhouse-client --query="SHOW TABLES" --password=[пароль
clickhouse] | wc -l
```

4. Открыть файл конфигурации для настройки уровня логирования:

```
sudo nano /etc/clickhouse-server/config.xml
```

5. Установить уровень логирования *information*:

```
<level>information</level>
```

6. Перезапустить службу:

```
sudo systemctl restart clickhouse-server
```

3.5 Настройка Clickhouse-server (только если для Clickhouse задан пароль)

1. Если при установке *clickhouse* был установлен пароль для пользователя, необходимо отредактировать конфигурационный файл:

```
sudo nano /etc/carbon-clickhouse/carbon-clickhouse.conf
```

В секциях `[upload.graphite]` и `[upload.graphite_index]` указать параметры подключения в формате:

```
default:[пароль clickhouse]@localhost:8123
```

вместо стандартного `localhost:8123`.

2. Включить автозапуск сервиса и проверить его состояние:

```
sudo systemctl enable --now carbon-clickhouse && systemctl status carbon-clickhouse
```

3.6 Настройка Graphite-clickhouse (только если для Clickhouse задан пароль)

1. Если при установке *clickhouse* был установлен пароль для пользователя, необходимо отредактировать конфигурационный файл:

```
sudo nano /etc/graphite-clickhouse/graphite-clickhouse.conf
```

В секции `[clickhouse]` указать параметры подключения в формате:

```
default:[пароль clickhouse]@localhost:8123
```

вместо стандартного `localhost:8123`.

2. Включить автозапуск сервиса и проверить его состояние:

```
sudo systemctl enable --now graphite-clickhouse && systemctl status graphite-clickhouse
```

3.7 Настройка Carbonapi

Carbonapi — это сервис для обработки и агрегации запросов к временным рядам метрик, получаемых из хранилища Clickhouse и других совместимых back-end систем. Carbonapi реализует совместимый с Graphite API, обеспечивая быстрый доступ к данным метрик и поддержку различных функций агрегации.

Включить автозапуск службы *carbonapi* и проверить её состояние:

```
sudo systemctl enable --now carbonapi && systemctl status carbonapi
```

3.8 Настройка MongoDB

MongoDB — это документо-ориентированная база данных, используемая для хранения событий и другой структурированной информации, необходимой для работы компонентов FlowCollector.

1. Запустить службу MongoDB и включить автозапуск, затем проверить текущее состояние:

```
sudo systemctl enable --now mongod && systemctl status mongod
```

2. Создать коллекцию *reports* в базе данных *test*:

```
mongosh test --eval 'db.createCollection("reports")'
```

3. Создать индексы для базы данных:

```
mongosh test --eval 'db.reports.createIndex({ mo: 1,
unixStartTime: 1, unixLastTime: 1 })'
```

4. Создать пользователя для работы приложений. Указать имя пользователя и пароль в соответствии с политикой безопасности вашей организации.

Выполнить в командной строке:

```
mongosh
```

```
db.createUser(
  {
    user: "events",
    pwd: "events",
    roles: [ { role: "readWrite", db: "test" } ]
  }
)
exit
```

5. Отредактировать файл конфигурации MongoDB:

```
sudo nano /etc/mongod.conf
```

6. Добавить или раскомментировать секцию *security*, включив авторизацию:

```
security:
  authorization: enabled
```

Для автоматизации изменения можно использовать команду:

```
sudo sed -i 's/^#\?security:/security:\n  authorization: enabled/'
/etc/mongod.conf
```

7. Для снижения объёма логов MongoDB установите минимальный уровень логирования:

```
systemLog:
  verbosity: 0
```

8. Перезапустить службу MongoDB для применения изменений:

```
sudo systemctl restart mongod
```

Примечание
После включения авторизации для всех подключений к MongoDB потребуется указание имени пользователя и пароля.

3.9Настройка PostgreSQL

PostgreSQL — это объектно-реляционная система управления базами данных, используемая для хранения служебной информации и сессий различных компонентов FlowCollector.

1. Подключиться к PostgreSQL под пользователем *postgres*:

```
sudo -u postgres psql
```

2. Создать базы данных и пользователей, назначить права доступа, выполнив соответствующие команды в интерактивной консоли (пароли задать согласно требованиям безопасности вашей организации).:

```
CREATE DATABASE spider;
CREATE DATABASE grafana;

CREATE USER spider WITH ENCRYPTED PASSWORD 'spider';
GRANT ALL PRIVILEGES ON DATABASE spider TO spider;

CREATE USER grafana WITH ENCRYPTED PASSWORD 'grafana';
GRANT ALL PRIVILEGES ON DATABASE grafana TO grafana;
```

3. Переключиться в базу данных *grafana* и создать таблицу сессий:

```
\c grafana
CREATE TABLE session (
  key CHAR(16) NOT NULL,
  data bytea,
  expiry INT NOT NULL,
  PRIMARY KEY (key)
);
\q
```

3.10 Настройка NATS

NATS — это высокопроизводительная система обмена сообщениями (message broker), применяемая для интеграции сервисов FlowCollector и доставки событий между компонентами.

1. Включить автозапуск и запустить сервис NATS-server:

```
sudo systemctl enable --now nats-server
```

2. Создать поток с использованием конфигурационного файла:

```
nats stream add --config /opt/nats/stream.conf
```

3. Создать обработчик сообщений (используются настройки по умолчанию):

```
nats consumer add \
    --pull \
    --deliver all \
    --ack explicit \
    --wait 30s \
```

```
    --replay instant \
    --max-pending 1000 \
    --max-waiting 512 \
    --inactive-threshold 0 \
    analyzer first
```

3.11 Настройка RabbitMQ

RabbitMQ — это брокер сообщений, обеспечивающий обмен данными между различными сервисами и модулями в инфраструктуре FlowCollector.

1. Создать пользователя с именем и паролем, соответствующими требованиям безопасности вашей организации:

```
sudo rabbitmqctl add_user "spider" "spider"
```

2. Выдать пользователю *spider* разрешения на операции `configure`, `write` и `read` для всех объектов системы:

```
sudo rabbitmqctl set_permissions -p "/" "spider" ".*" ".*" ".*"
```

3.12 Настройка конфигурации FlowCollector

Конфигурация разделена на логические блоки для удобства восприятия.

С неразделённым вариантом конфигурации можно ознакомиться здесь

```
# Параметры логирования
log:
  # Уровень логирования
  # trace, debug, info, warn/warning, error, fatal
  # ПРИМЕЧАНИЕ: уровни trace и debug могут быть отключены в
определенных типах сборки (Release)
  level: debug

# Параметры привязки логических ядер
# Опционально. Если не указать, распределение произойдёт
автоматически
# lcore-mapping:
  # Режим привязки: auto, manual
  # Рекомендуется устанавливать количество dpdk-rx кратным
```

```
степени 2
  # Опционально. По умолчанию - auto
  # mode: auto
  # schema определяет количество логических ядер, назначенных
для каждой активности
  # schema:
  #   auto:
  #     detection: 4
  #     event-processing: 2
  #     metrics: 2
  #     dpdk-rx: 1
  #     dpdk-worker: 4
  #     ipfix-rx: 1
  #     ipfix-worker: 2
  #     reports: 2
  #     free: 2
  #   manual:
  #     detection: 4
  #     event-processing: 2
  #     metrics: 2
  #     dpdk-rx: 1
  #     dpdk-worker: 4
  #     ipfix-rx: 1
  #     ipfix-worker: 2
  #     reports: 2
  #     free: 2

# Параметры обнаружения
detection:
  # true - включено, false - выключено
  # По умолчанию - true
  enable: true

# Параметры метрик
# В Clickhouse метрики записываются в формате [hostname].
[plugin].*
metrics:
  enable: true
  carbon:
    # Отправка метрик в carbon-clickhouse
    endpoint: 127.0.0.1:2003
  # Hostname - ключ для метрик FlowCollector, с которым они
будут записываться в ClickHouse
  hostname: flowcollector
  # Plugin - второй ключ для метрик FlowCollector, с которым они
будут записываться в ClickHouse
  plugin: analyzer
  # Частота сбора метрик в секундах
  interval: 5
  # Частота отправки метрик в секундах
  export-timeout: 5
```

```yaml
# Параметры нативного BGP для анонсов
bgp:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  asn: 1
  id: 192.168.1.2
  host: 127.0.0.1
  port: 1179

# Параметры GoBGP для анонсов
gobgp:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  # Хост API GoBGP
  # По умолчанию - localhost
  host: localhost
  # Порт API GoBGP
  # По умолчанию - 50051
  port: 50051
  # Канал API GoBGP выполняет запросы в синхронном режиме,
  # поэтому работает в отдельных потоках
  # Обычно достаточно 1
  # По умолчанию - 1
  # thread-number: 1
  # true - включено, false - выключено
  # По умолчанию - false
  enable-subnet-splitting: false
  # Установка длины маски для разделения CIDR на подсети
  # Обратите внимание, если min-subnet-length <= длины маски
CIDR,
  # то разделение на подсети выполняться не будет
  # По умолчанию - 16
  min-subnet-length: 16
  # Включение/выключение FlowSpec
  # true - включено, false - выключено
  enable-flow-spec: false
  # Ограничение максимального количества правил, генерируемых
анализатором
  # Опционально. По умолчанию - 100
  max-rules-number: 100
  # Правила FlowSpec. Содержит набор правил, состоящих из
числовых полей или полей битовой маски
  # Числовые поля могут комбинироваться со следующими
операторами:
  #   [&] [== | > | >= | < | <= | !=] (см. пример ниже)
  # Поля битовой маски могут комбинироваться со следующими
операторами:
  #     [&] [= | ! | !=] (см. пример ниже)
```

```
  # Обратите внимание, что оператор '&' ставится только перед
операторами сравнения
  # Между оператором и его аргументом нет пробела
  # Опционально. Может использоваться как правила по умолчанию
для управляемых объектов
  flow-spec-rules:
  -
    # Имя правила FlowSpec. Должно быть уникальным
    # Обязательный параметр
    name: "flow-spec"
    # CIDR назначения. Это поле извлекается из атаки
    # Если указано, будет использовано это значение
    # Опционально. По умолчанию - пусто
    dst-cidr: 10.0.0.1/24
    # CIDR источника
    # Опционально. По умолчанию - пусто
    src-cidr: 10.0.0.1/24
    # Имя протокола, десятичное число, true или false. Числовое
поле
    # Доступные опции: egp, gre, icmp, igmp, igp, ipip, ospf,
pim, rsvp, sctp, tcp, udp
    # Опционально. По умолчанию - пусто
    ip-protocols: "==tcp &==udp icmp >igmp >=egp <igp <=rsvp
!=gre &!=ospf true"
    # Тип фрагмента или их комбинация, соединенная знаком +.
Поле битовой маски
    # Доступные опции: dont-fragment, is-fragment, first-
fragment, last-fragment, not-a-fragment
    # Опционально. По умолчанию - пусто
    fragments: "dont-fragment is-fragment+first-fragment"
    # Флаг TCP или их комбинация. Поле битовой маски
    # Доступные опции: F, S, R, P, A, U, E, C
    # Опционально. По умолчанию - пусто
    tcp-flags: "=S &=SA A !F !=U !=C"
    # Порт источника ИЛИ назначения. Десятичное число, true или
false. Числовое поле
    # Опционально. По умолчанию - пусто
    # Простой пример: ports: 80
    ports: "==80 &==90 8080 >9090 >=10080 <10090 <=18080 !=19090
&!=443 true"
    # Порт назначения TCP или UDP пакета. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
    destination-ports: "==80 >=8080&<=8888"
    # Порт источника TCP или UDP пакета. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
    source-ports: "443"
    # Поле типа ICMP пакета. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
```

```
    icmp-types: "0"
    # Поле кода ICMP пакета. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
    icmp-codes: "==0 >1&<3 true"
    # Общая длина IP пакета. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
    packet-lengths: "64"
    # Поле DSCP. Числовое поле
    # Десятичное число, true или false
    # Опционально. По умолчанию - пусто
    dscp: ">=0&<=32"
    # Действие для фильтрации трафика
    # Доступные опции:
    #   - accept (принять трафик)
    #   - discard (отбросить трафик, используя traffic-rate 0)
    #   - rate-limit <RATE> [as <AS>] (указать скорость трафика
в виде числа с плавающей точкой)
    #   - redirect <CIDR> (перенаправить в VRF, который имеет
данный RT в своей политике импорта)
    #   - mark <DEC_NUM> (изменяет DSCP в IPv4 или Traffic Class
in IPv6 на указанное значение)
    #   - action sample (включает сэмплирование и логирование
трафика)
    #   - action terminal (указывает завершение фильтра трафика)
    #   - action sample-terminal (указывает одновременно sample
и terminal)
    # Опционально. По умолчанию - accept
    actions:
    - "rate-limit 100.0 as 65000"
    - "action sample"
  # Удалить все существующие маршруты и правила FlowSpec в GoBGP
при перезапуске
  # true - включено, false - выключено
  # По умолчанию - true
  drop-on-restart: true

# Аргументы DPDK EAL
# По умолчанию - пусто
# Пример использования сетевого интерфейса в режиме PCAP
# --vdev=net_pcap0,rx_iface_in=eth0
# Пример использования сетевого интерфейса с dpdk-testpmd
# --vdev=net_af_packet0,iface=tap0,framecnt=512,qpairs=1 --in-
memory --no-pci
# Пример использования сетевого интерфейса
# -a <port>
# dpdk-args:

# Параметры HTTP API FlowCollector
http-api:
```

```
  # true - включено, false - выключено
  # По умолчанию - false
  enable: true
  # Локальные точки подключения, где FC слушает соединения
  endpoint: inet://127.0.0.1:10505
    # - unix://path/to/socket
    # - inet://127.0.0.1:8082
    # - 127.0.0.1

# Параметры DPDK NIC
dpdk-nic:
  # true - включено, false - выключено
  # По умолчанию - true
  enable: false
  # Маска для включенных ethernet портов
  # Опционально. По умолчанию - max(uint64)
  # enable-eth-ports: 0xff
  # Опционально. Количество очередей Rx на сетевой интерфейс
  # По умолчанию - 1
  # port-rx-queue-number: 1
  # Коэффициент регулирования (1:throttling-rate)
  # Опционально. По умолчанию - 1
  # throttling-rate: 1
  # Параметры пула пакетов DPDK
  # packet-pool:
    # Размер пула. По умолчанию - (64*1024)-1
    # size: 8191
    # Размер кэша. По умолчанию - 128
    # cache-size: 128
  # Параметры ARP
  # arp:
    # true - включено, false - выключено
    # По умолчанию - false
    # enable: false
    # Список устройств для обработки. По умолчанию отсутствует
    # Если указаны и devices, и default-ip, будет использовано
значение default-ip
    # Если arp включен, но ни одна из следующих опций не
указана, возникнет исключение
    # devices:
      # -
        # MAC-адрес устройства
        # mac: 00:00:00:00:00:00
        # Список IP-адресов устройства
        # ip:
          # - 127.0.0.1
          # - 192.168.0.0
    # IP-адрес по умолчанию машины, на которой будет работать
анализатор
    # По умолчанию - 127.0.0.1
    # default-ip: 127.0.0.1
```

```yaml
  # Список источников трафика DPDK для сопоставления с группами
  sources:
  -
    # Имя группы. Опционально. Если не указано, используется
автогенерируемый id группы как имя
    # group: all
    group: all
    # Если не установлены ни local, ни remote, это сырое
зеркалирование
    # Локальная точка. Опционально. Маска. Пустой 'local'
(local: {}) означает любой
    # local:
      # IP-адрес. Опционально. Отсутствие host означает любой
      # host: 192.168.0.1
    local: {}
    # Удаленная точка. Опционально. Маска. Пустой 'remote'
(remote: {}) означает любой
    # remote:
      # IP-адрес. Опционально. Отсутствие host означает любой
      # host: 192.168.0.2
    remote: {}

# Параметры IPFIX
ipfix:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  # Локальные точки подключения для приема IPFIX сообщений
  endpoint:
    - 127.0.0.1:4739
    # - 192.168.0.1:4739
  # Включение режима эксклюзивных сокетов
  # true - включено, false - выключено
  # По умолчанию: false
  # Доступны два режима для сокетов:
  # - кооперативный: каждое ядро ipfix rx обрабатывает все
сокеты. Это режим по умолчанию
  # - эксклюзивный: сокеты равномерно распределяются между
ядрами, каждое ядро обрабатывает свой набор сокетов
  # exclusive-sockets-mode: false
  # Пул памяти для IPFIX сообщений
  # memory-pool:
    # Размер пула. По умолчанию - 1024*64-1
    # number: 16383
    # Размер кэша. По умолчанию - 128
    # cache-size: 64
  # Список источников трафика IPFIX для сопоставления с группами
  sources:
  -
    # Имя группы. Опционально. Если не указано, используется
автогенерируемый id группы как имя
```

```
    group: all
    # Локальная точка. Опционально. Маска. Пустой 'local'
(local: {}) означает любой
    # local:
      # IP-адрес. Опционально. Отсутствие host означает любой
      # host: 192.168.0.1
      # Порт. Опционально. Не установлен означает любой
      # port: 20000
    local: {}
    # Удаленная точка. Опционально. Маска. Пустой 'remote'
(remote: {}) означает любой
    # remote:
      # IP-адрес. Опционально. Отсутствие host означает любой
      # host: 192.168.0.2
      # Порт. Опционально. Не установлен означает любой
      # port: 30000
    remote: {}

# Параметры счетчиков пакетов
packet-counters:
  # Коэффициент. Означает "каждый RATE пакет отправляется в flow
collector",
  # другими словами FlowCollector вычисляет реальное количество
пакетов как
  # полученные * rate
  # По умолчанию - 1
  rate: 1
  # Временной период, используемый для расчета текущей скорости.
Миллисекунды
  # По умолчанию - 5000
  average-period: 5000
  # Временной период для расчета текущих скоростей
  # Текущие скорости рассчитываются каждые <interval>
миллисекунд
  interval: 5000

# Параметры для отправки BGP анонсов
announce:
  # Включение/выключение отправки анонсов
  # true - включено, false - выключено
  enable: false

  # BGP next-hop'ы
  # Имя next-hop'а может использоваться как сокращение в опции
action
  # Опционально
  nexthops:
  -
    # Имя next-hop'а
    name: transit
    # IP-адрес next-hop'а
```

```
    ip: 192.168.10.1

  -
    name: blackhole
    ip: 192.168.20.1
    # Список сообществ или одно сообщество для BGP Update
сообщения
    # Доступные форматы:
    # - <целое_число_основание_10>:<целое_число_основание_10>
    # - <целое_число_основание_10>
    # - 0x<целое_число_основание_16>
    # Опционально
    # communities: 666
    # communities
    # - 666:666
    # - 777:777

  # Временной интервал отправки анонсов. Секунды
  # Только для нативного BGP
  timeout: 20
# Конец announce

# Глобальное действие для обнаруженной аномалии
# Выполняется, если любое из правил совпадает
# и если нет специфического действия, назначенного правилу
# На данный момент доступен только BGP анонс как действие
action:
  # IP-адрес next-hop'a или имя из announce.nexthops
  # Обязательно
  nexthop: transit
  # Список сообществ или одно сообщество для BGP Update
сообщения
  # Доступные форматы:
  # - <целое_число_основание_10>:<целое_число_основание_10>
  # - <целое_число_основание_10>
  # - 0x<целое_число_основание_16>
  # Опционально
  # Если присутствует, перезаписывает
announce.nexthops.communities для данного next-hop'a
  # communities: 666:666
  # communities:
  # - 666:666
  # - 777:777
# Конец action

# Глобальный период подтверждения аномалии. Секунды
# Если FlowCollector обнаруживает аномалию, он ждет confirm-
period секунд
# и подтверждает аномалию, если она все еще присутствует
# FlowCollector выполняет действия для подтвержденных аномалий
# По умолчанию - 0
```

```yaml
confirm-period: 0

# Глобальный таймаут истечения действия. Секунды
# Если FlowCollector выполняет действия для аномалии
# и перестает ее обнаруживать, он продолжает выполнять действие
# action-expiry-timeout секунд, если аномалия не обнаружена
снова
# По умолчанию - 60
action-expiry-timeout: 60

action-events:
  enable: true
  # Если graphite: false, оповещения отправляются в компонент
sp-events
  graphite: false
  # Хост, на котором располагается компонент sp-events
  host: 127.0.0.1
  # Порт, на котором располагается компонент sp-events. По-
умолчанию 8081
  port: 8081
  # Путь к компоненту
  path: /

# Отправка событий
notification-service:
  enable: true
  nats:
    enable: true
    host: localhost
    port: 4222
    path: /
    subject: analyzer.notification
    persistance: true

# Параметры IpLookupTable
ip-lookup-table:
  # Указать режим отслеживания CIDR
  # Доступные значения:
  #  - overlap: разрешены перекрывающиеся подсети в разных
управляемых объектах
  #  - nooverlap: запрещены перекрывающиеся подсети в разных
управляемых объектах
  # По умолчанию: nooverlap
  mode: overlap

# Параметры отчетов
reports:
  enable: true
  host: localhost
  port: 9080
  path: /report
```

```
  export-timeout: 10
  max-src-addrs: 10000
  max-dst-addrs: 10000
  max-src-ports: 1000
  max-dst-ports: 1000

# Параметры PcapngDumper
pcapng-dumper:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  # Указать путь для записи дампов
  # По умолчанию - /var/dump/
  # dumps-path: /var/dump/
  # Указать максимальное количество одновременных сессий дампа
  # По умолчанию - 8
  max-sessions-number: 8
  # Указать максимальное количество пакетов в одном дампе
  # По умолчанию - 10'000
  max-packets-number: 10000
  # Указать максимальный таймаут в секундах без пакетов для
ручных дампов
  # По умолчанию - 10
  keep-alive: 10

# Параметры FlowWriter
save-flow:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  # Указать, нужно ли сохранять поток для всех перечисленных МО
  # Если выключено, то опция save-flow будет работать только для
МО, у которых эта опция включена
  # true - включить сохранение IPFIX потока для всех МО, false -
выключить
  # По умолчанию - true
  enable-all-mo: true
  # Указать, нужно ли сохранять поток для всего полученного
трафика
  # true - включить сохранение IPFIX потока для всего трафика,
false - выключить
  # По умолчанию - false
  enable-save-all: false
  # Указать имя таблицы для записи записей потока
  # По умолчанию - default.flows
  table-name: "default.flows"
  # Указать адрес хоста базы данных
  # По умолчанию - localhost
  db-host: "localhost"
  # Указать пользователя базы данных
  # По умолчанию - default
```

```
  db-username: "default"
  # Указать пароль к базе данных
  db-password: "[пароль clickhouse]"
  # Указать порт базы данных
  # По умолчанию - 9000
  dp-port: 9000
  # Указать имя секции в конфигурации clickhouse, где находятся
правила свертки
  # По умолчанию - flows_rollup
  rollup-section: "flows_rollup"
  # Указать количество ipfix записей для записи в один блок
  # По умолчанию - 10'000
  save-records-number: 10000
  # Указать таймаут в секундах для сброса ipfix записей
  # По умолчанию - 60
  flush-records-timeout: 60

# Параметры SNMP
# Примечание: Flowcollector отслеживает только интерфейсы типов
# - ethernet-csmacd(6)
# - prop-virtual(53)
# - tunnel(131)
snmp:
  # true - включено, false - выключено
  # По умолчанию - false
  enable: false
  # Список агентов для запроса
  agents:
    # Хост пира и опционально порт - host[:port]
    # Порт по умолчанию - 161
  - peer: example.org:161
    # Версия SNMP. Поддерживаемые версии - 1, 2c, 3
    # Опционально. По умолчанию - 1
    version: 1
    # Частота запросов
    # Опционально. По умолчанию - 60s
    request-timeout: 60s
    # Таймаут ожидания ответа
    # Опционально. По умолчанию - 1s
    response-timeout: 1s
    # Локальный адрес для привязки. Опционально
    # local-address: 192.168.0.1
    # Локальный порт для привязки. Опционально
    # local-port: 11111
    # Поле community SNMP для авторизации на SNMP агентах версии
1 и 2c. Опционально
    community: public
    # Поле security name SNMP для авторизации на SNMP агентах
версии 3. Опционально
    # security-name: secret
    # Поле security auth key SNMP для авторизации на SNMP
```

```
агентах версии 3. Опционально
    # security-auth-key: secret-key
    # Псевдоним для метрик
    # Опционально. По умолчанию - хост пира
    # alias: example
```

Внимание!

- Перед началом работы рекомендуется изучить комментарии к каждому параметру.
- Для режима DPDK необходимо раскомментировать секцию dpdk-nic.
- Для режима IPFIX необходимо раскомментировать секцию ipfix.

Открыть файл конфигурации для редактирования:

```
sudo nano /opt/spfc/etc/analyzer.yaml
```

3.12.1 Параметры логирования

```
log:
  level: debug                    # Уровень логирования: trace,
debug, info, warn/warning, error, fatal
```

Примечание
Уровни логирования *trace* и *debug* могут быть отключены в определённых типах сборки.

3.12.2 Основные параметры

Привязка логических ядер

Если секция закомментирована — распределение потоков по ядрам происходит автоматически. Для оптимизации нагрузки можно раскомментировать и отредактировать значения.

```
# lcore-mapping:
#   mode: auto                        # auto | manual
#   schema:                           # определяет количество
логических ядер, назначенных для каждой активности
#     auto:
#       detection: 4
#       event-processing: 2
#       metrics: 2
#       dpdk-rx: 1
#       dpdk-worker: 4
#       ipfix-rx: 1
#       ipfix-worker: 2
#       reports: 2
#       free: 2
#     manual:
#       detection: 4
#       event-processing: 2
#       metrics: 2
#       dpdk-rx: 1
#       dpdk-worker: 4
#       ipfix-rx: 1
#       ipfix-worker: 2
#       reports: 2
#       free: 2
```

Параметры обнаружения

```
detection:
  enable: true                        # true — включено, false —
выключено (по умолчанию — true)
```

Параметры метрик

В Clickhouse метрики записываются в формате `[hostname].[plugin].*`

```
metrics:
  enable: true
  carbon:
    endpoint: 127.0.0.1:2003          # Отправка метрик в carbon-
clickhouse
  hostname: flowcollector             # Hostname - ключ для метрик
FlowCollector, с которым они будут записываться в ClickHouse
  plugin: analyzer                    # Plugin - второй ключ для
метрик FlowCollector, с которым они будут записываться в
ClickHouse
  interval: 5                         # Частота сбора метрик в
```

```
секундах
  export-timeout: 5                  # Частота отправки метрик в
секундах
```

3.12.3 Настройки BGP и GoBGP

Нативный BGP

```
bgp:
  enable: false    # true — включено, false — выключено (по
умолчанию — false)
  asn: 1
  id: 192.168.1.2
  host: 127.0.0.1
  port: 1179
```

GoBGP и FlowSpec

BGP и GoBGP используются для анонсирования маршрутов и применения фильтрации через FlowSpec. Рекомендуется активировать только нужные секции и тщательно настраивать правила.

```
gobgp:
  enable: false                      # true — включено, false —
выключено (по умолчанию — false)
  host: localhost                    # Хост API GoBGP (по
умолчанию — localhost)
  port: 50051                        # Порт API GoBGP, (по
умолчанию — 50051)
  # thread-number: 1                 # Количество потоков API
GoBGP (по умолчанию — 1)
  enable-subnet-splitting: false     # Разделение подсетей true —
включено, false — выключено
  min-subnet-length: 16              # Длина маски для разбиения
(по умолчанию — 16). Если min-subnet-length <= длины маски CIDR,
то разделение на подсети выполняться не будет
  enable-flow-spec: false            # FlowSpec true — включено,
false — выключено
  max-rules-number: 100              # Ограничение максимального
количества правил (по умолчанию — 100)
```

```
  # Примеры правил FlowSpec
  flow-spec-rules:
```

```
    - name: "flow-spec"
      dst-cidr: 10.0.0.1/24
      src-cidr: 10.0.0.1/24
      ip-protocols: "==tcp &==udp icmp >igmp >=egp <igp <=rsvp
!=gre &!=ospf true"
      fragments: "dont-fragment is-fragment+first-fragment"
      tcp-flags: "=S &=SA A !F !=U !=C"
      ports: "==80 &==90 8080 >9090 >=10080 <10090 <=18080 !=19090
&!=443 true"
      destination-ports: "==80 >=8080&<=8888"
      source-ports: "443"
      icmp-types: "0"
      icmp-codes: "==0 >1&<3 true"
      packet-lengths: "64"
      dscp: ">=0&<=32"
      actions:
        - "rate-limit 100.0 as 65000"
        - "action sample"
  drop-on-restart: true              # Удалить все существующие
маршруты и правила FlowSpec при перезапуске
```

3.12.4 Аргументы DPDK EAL

```
# По умолчанию - пусто
# --vdev=net_pcap0,rx_iface_in=eth0  # Пример использования
сетевого интерфейса в режиме PCAP
# --vdev=net_af_packet0,iface=tap0,framecnt=512,qpairs=1 --in-
memory --no-pci    # Пример использования сетевого интерфейса с
dpdk-testpmd
# -a <port>                          # Пример использования
сетевого интерфейса
# dpdk-args:
```

3.12.5 Параметры HTTP API FlowCollector

```
http-api:
  enable: true                       # true - включено, false -
выключено (по умолчанию — false)
  endpoint: inet://127.0.0.1:10505  # Локальные точки подключения,
где FC слушает соединения
    # - unix://path/to/socket
    # - inet://127.0.0.1:8082
    # - 127.0.0.1
```

3.12.6 Сетевые интерфейсы (DPDK, IPFIX)

DPDK и IPFIX настраиваются только для соответствующих режимов работы FlowCollector. Оставьте ненужные секции закомментированными.

DPDK NIC (режим DPDK)

```
dpdk-nic:
  enable: false                        # true — включено, false — выключено
  # enable-eth-ports: 0xff             # Маска включённых ethernet портов  (по умолчанию - max(uint64))
  # port-rx-queue-number: 1            # Количество очередей Rx на сетевой интерфейс (по умолчанию - 1)
  # throttling-rate: 1                 # Коэффициент регулирования (по умолчанию - 1)
  # packet-pool:                       # Параметры пула пакетов DPDK
  #   size: 8191                       # Размер пула (по умолчанию - (64*1024)-1)
  #   cache-size: 128                  # Размер кэша (по умолчанию - 128)
  # arp:                               # Параметры ARP
  #   enable: false                    # true - включено, false - выключено (по умолчанию — false)
  #   devices:                         # Список устройств для обработки. Если указаны и devices, и default-ip, будет использовано значение default-ip. Если arp включен, но ни одна из следующих опций не указана, возникнет исключение
  #     -
  #       mac: 00:00:00:00:00:00       # MAC-адрес устройства
  #       ip:                          # Список IP-адресов устройства
  #         - 127.0.0.1
  #         - 192.168.0.0
  #   default-ip: 127.0.0.1            # IP-адрес по умолчанию машины, на которой будет работать анализатор
  sources:                             # Список источников трафика DPDK для сопоставления с группами
    -
      group: all                       # Имя группы. Если не указано, используется автогенерируемый id группы как имя group: all
  # Если не установлены ни local, ни remote, это сырое зеркалирование
  # local:                             # Локальная точка. Маска. Пустой 'local' (local: {}) означает любой
    # host: 192.168.0.1                # IP-адрес. Отсутствие host означает любой
```

```
  local: {}                          # Локальная точка. Маска.
Пустой 'local' (local: {}) означает любой
  # remote:                          # Удаленная точка. Маска.
Пустой 'remote' (remote: {}) означает любой
    # host: 192.168.0.2              # IP-адрес. Отсутствие host
означает любой
  remote: {}                         # Удаленная точка. Маска.
Пустой 'remote' (remote: {}) означает любой
```

IPFIX (режим IPFIX)

```
ipfix:
  enable: false                        # true - включено, false -
выключено. По умолчанию - false
  endpoint:
    - 127.0.0.1:4739                   # Локальные точки подключения
для приема IPFIX сообщений
   # - 192.168.0.1:4739
  # exclusive-sockets-mode: false     # Включение режима
эксклюзивных сокетов. true - включено, false - выключено. По
умолчанию: false. Доступны два режима: кооперативный (по
умолчанию), эксклюзивный
  # memory-pool:                       # Пул памяти для IPFIX
сообщений
    # number: 16383                    # Размер пула. По умолчанию -
1024*64-1
    # cache-size: 64                   # Размер кэша. По умолчанию -
128
  sources:                             # Список источников трафика
IPFIX для сопоставления с группами
   -
     group: all                        # Имя группы. Если не
указано, используется автогенерируемый id группы как имя
     # local:
      # host: 192.168.0.1              # IP-адрес. Отсутствие host
означает любой
      # port: 20000                    # Порт.  Не установлен
означает любой
     local: {}                         # Локальная точка. Маска.
Пустой 'local' (local: {}) означает любой
     # remote:
      # host: 192.168.0.2              # IP-адрес. Отсутствие host
означает любой
      # port: 30000                    # Порт. Не установлен
означает любой
     remote: {}                        # Удаленная точка. Маска.
Пустой 'remote' (remote: {}) означает любой
```

3.12.7. Обработка потоков и политики

Параметры счетчиков пакетов

```
packet-counters:
  rate: 1                              # Коэффициент. Означает
"каждый RATE пакет отправляется в flow collector", другими словами
FlowCollector вычисляет реальное количество пакетов как полученные
* rate (по умолчанию - 1)
```

```
  average-period: 5000                # Временной период,
используемый для расчета текущей скорости, в миллисекундах (по
умолчанию - 5000)
  interval: 5000                      # Временной период для расчета
текущих скоростей. Текущие скорости рассчитываются каждые
<interval> миллисекунд
```

Анонсы и действия (announce, action)

```
# Параметры для отправки BGP анонсов
announce:
  enable: false                       # Включение/выключение
отправки анонсов. true - включено, false - выключено
  nexthops:                           # BGP next-hop'ы. Имя next-
hop'a может использоваться как сокращение в опции action.
    - name: transit                   # Имя next-hop'a
      ip: 192.168.10.1                # IP-адрес next-hop'a
    - name: blackhole
      ip: 192.168.20.1                # IP-адрес next-hop'a
      # communities: 666              # Список сообществ или одно
сообщество для BGP Update сообщения
      # communities:                  # Доступные форматы:
      #   - 666:666                   # -
<целое_число_основание_10>:<целое_число_основание_10>
      #   - 777:777                   # - <целое_число_основание_10>
                                      # -
0x<целое_число_основание_16>
  timeout: 20                         # Временной интервал отправки
анонсов (секунды). Только для нативного BGP
# Конец announce



# Глобальное действие для обнаруженной аномалии. Выполняется, если
любое из правил совпадает
# и если нет специфического действия, назначенного правилу. На
данный момент доступен только BGP анонс как действие
action:
  nexthop: transit                    # IP-адрес next-hop'a или имя
из announce.nexthops (обязательно)
  # communities: 666:666              # Список сообществ или одно
сообщество для BGP Update сообщения
  # communities:                      # Доступные форматы:
  #   - 666:666                       # -
<целое_число_основание_10>:<целое_число_основание_10>
  #   - 777:777                       # - <целое_число_основание_10>
                                      # -
0x<целое_число_основание_16>
                                      # Если присутствует,
```

```
перезаписывает announce.nexthops.communities для данного next-hop'a
# Конец action
```

Глобальные параметры обнаружения

```
confirm-period: 0                    # Глобальный период подтверждения аномалии (секунды). Если FlowCollector обнаруживает аномалию, он ждет confirm-period секунд и подтверждает аномалию, если она все еще присутствует. FlowCollector выполняет действия для подтвержденных аномалий. По умолчанию - 0

action-expiry-timeout: 60            # Глобальный таймаут истечения действия (секунды). Если FlowCollector выполняет действия для аномалии и перестает ее обнаруживать, он продолжает выполнять действие action-expiry-timeout секунд, если аномалия не обнаружена снова. По умолчанию - 60
```

Оповещения и интеграция с внешними сервисами

```
action-events:
  enable: true                       # Включение/выключение action-events
  graphite: false                    # Если graphite: false, оповещения отправляются в компонент sp-events
  host: 127.0.0.1                    # Хост, на котором располагается компонент sp-events
  port: 8081                         # Порт, на котором располагается компонент sp-events (по умолчанию 8081)
  path: /                            # Путь к компоненту

# Отправка событий
notification-service:
  enable: true                       # Включение/выключение notification-service
  nats:
    enable: true                     # Включение/выключение NATS
    host: localhost                  # Хост NATS
    port: 4222                       # Порт NATS
    path: /                          # Путь
    subject: analyzer.notification   # Subject для NATS
    persistance: true                # Включение/выключение постоянства сообщений
```

Параметры IpLookupTable

```
ip-lookup-table:
  mode: overlap                       # Режим отслеживания CIDR.
Доступные значения: overlap — разрешены перекрывающиеся подсети в
разных управляемых объектах, nooverlap — запрещены перекрывающиеся
подсети в разных управляемых объектах. По умолчанию: nooverlap
```

3.12.8 Параметры отчетов

```
# Параметры отчетов
reports:
  enable: true                        # Включение/выключение отчетов
  host: localhost                     # Хост для экспорта отчетов
  port: 9080                          # Порт для экспорта отчетов
  path: /report                       # Путь для экспорта отчетов
  export-timeout: 10                  # Таймаут экспорта отчетов
(секунды)
  max-src-addrs: 10000                # Максимальное количество
исходных адресов в отчете
  max-dst-addrs: 10000                # Максимальное количество
целевых адресов в отчете
  max-src-ports: 1000                 # Максимальное количество
исходных портов в отчете
  max-dst-ports: 1000                 # Максимальное количество
целевых портов в отчете
```

3.12.9 Хранение данных

PCAP-дампы

```
# Параметры PcapngDumper
pcapng-dumper:
  enable: false                       # Включение/выключение дампа
(true - включено, false - выключено, по умолчанию - false)
  # dumps-path: /var/dump/            # Путь для записи дампов (по
умолчанию - /var/dump/)
  max-sessions-number: 8              # Максимальное количество
одновременных сессий дампа (по умолчанию - 8)
  max-packets-number: 10000           # Максимальное количество
пакетов в одном дампе (по умолчанию - 10'000)
  keep-alive: 10                      # Максимальный таймаут в
секундах без пакетов для ручных дампов (по умолчанию - 10)
```

Параметры FlowWriter

```
save-flow:
  enable: false                        # Включение/выключение
сохранения потока (true - включено, false - выключено, по
умолчанию - false)
  enable-all-mo: true                  # Сохранять поток для всех
перечисленных MO (true - для всех MO, false - только для MO с
включенной опцией, по умолчанию - true)
  enable-save-all: false               # Сохранять поток для всего
полученного трафика (true - весь трафик, false - выключить, по
умолчанию - false)
  table-name: "default.flows"          # Имя таблицы для записи
потока (по умолчанию - default.flows)
  db-host: "localhost"                 # Адрес хоста базы данных (по
умолчанию - localhost)
  db-username: "default"               # Пользователь базы данных
(по умолчанию - default)
  db-password: "[пароль clickhouse]" # Пароль к базе данных
  dp-port: 9000                        # Порт базы данных (по
умолчанию - 9000)
  rollup-section: "flows_rollup"       # Имя секции в конфиге
ClickHouse с правилами свертки (по умолчанию - flows_rollup)
  save-records-number: 10000           # Количество ipfix записей
для одного блока (по умолчанию - 10'000)
  flush-records-timeout: 60            # Таймаут (секунды) для
сброса ipfix записей (по умолчанию - 60)
```

3.12.10 Параметры SNMP

```
# Примечание: Flowcollector отслеживает только интерфейсы типов
ethernet-csmacd(6), prop-virtual(53), tunnel(131)
snmp:
  enable: false                        # Включение/выключение SNMP
(true - включено, false - выключено, по умолчанию - false)
  agents:                              # Список агентов для запроса
    - peer: example.org:161            # Хост пира и опционально
порт - host[:port] (порт по умолчанию - 161)
      version: 1                       # Версия SNMP (1, 2c, 3).
Опционально, по умолчанию - 1
      request-timeout: 60s             # Частота запросов (по
умолчанию - 60s)
      response-timeout: 1s             # Таймаут ожидания ответа
(опционально, по умолчанию - 1s)
      # local-address: 192.168.0.1     # Локальный адрес для
привязки
      # local-port: 11111              # Локальный порт для привязки
      community: public                # SNMP community для
```

```
авторизации на агентах версии 1 и 2c
      # security-name: secret        # SNMP security name для
авторизации на агентах версии 3
      # security-auth-key: secret-key # SNMP security auth key для
авторизации на агентах версии 3
      # alias: example               # Псевдоним для метрик (по
умолчанию — хост пира)
```

3.13 Настройка тестового объекта

1. Для создания директории выполнить следующие команды:

```
sudo mkdir /opt/spfc/etc/mo/test_logic_folder
sudo mkdir /opt/spfc/etc/mo.enabled/test_logic_folder
```

2. Открыть и отредактировать файл конфигурации:

```
sudo nano /opt/spfc/etc/mo/test_logic_folder/test_object.yaml
```

3. Добавить в конфигурационный файл следующие тестовые настройки:

```
name: test_object           # Имя объекта, case-sensitive

cidrs:                      # Анализируемые сети
- 0.0.0.0/1
- 128.0.0.0/1

rules:                      # Блок правил
-
  type:
  - local                   # Счетчик на /32
  vectors:
  - total-traffic
  limit-threshold: 250Mb/s  # 250 мбит/с
-
  type:
  - global                  # Счетчик на весь объект
  vectors:
  - total-traffic
  limit-threshold: 2500Mb/s # 2.5 гбит/с
```

4. Создать символическую ссылку в директории активированных объектов:

```
sudo ln -s /opt/spfc/etc/mo/test_logic_folder/test_object.yaml
/opt/spfc/etc/mo.enabled/test_logic_folder/test_object.yaml
```

3.14 Настройка bind_driver (актуально только для режима DPDK)

> **Примечание**
> При использовании сетевых адаптеров Mellanox передача порта под управление DPDK не требуется.

1. Определить NIC порта, задействованного для получения зеркального трафика:

```
sudo /usr/local/bin/dpdk-devbind.py -s
```

Из полученного вывода требуется значение с последними пятью символами, например:

```
0000:13:00.0 'VMXNET3 Ethernet Controller 07b0' drv=vfio-pci
unused=vmxnet3</u>
```

В данном примере идентификатор порта — 13:00.0.

2. Открыть конфигурационный файл:

```
sudo nano /opt/spfc/etc/analyzer.yaml
```

3. Внести порт как аргумент DPDK в файл конфигурации:

```
dpdk-args: -a 13:00.0
```

4. Указать параметры сетевого интерфейса, работающего под управлением драйвера DPDK:

```
dpdk-nic:
  enable: true
  arp:
```

```
    enable: true
    #devices:
    #  - mac: "00:50:56:a8:51:79"
    #    ip:
    #    - 10.0.101.10
 # Если задействована одна сетевая карта, то достаточно будет
указать её IP (без маски)
    default-ip: 10.0.101.2
```

5. Задать значение порта, работающего под управлением драйвера DPDK:

```
echo "13:00.0" | sudo tee /opt/spfc/etc/nic_port
```

3.15 Запуск сервисов

1. Назначить права на выполнение скриптов и активировать системные сервисы:

```
sudo chmod +x /opt/spfc/bin/bind_driver.sh
/opt/spfc/bin/create_hugepages.sh
sudo systemctl enable --now \
    /opt/spfc/lib/systemd/system/bind_driver.service \
    /opt/spfc/lib/systemd/system/create_hugepages.service \
    /opt/spfc/lib/systemd/system/analyzer.service
```

2. Проверить состояние сервиса *bind_driver*:

```
sudo systemctl status bind_driver
```

Примечание
При работе в режиме IPFIX возможно сообщение:
`Warning: Configuration file /opt/spfc/etc/nic_port not found. Interface binding was skipped`
Это ожидаемое поведение.

3. Проверить статус сервиса *create_hugepages*:

```
sudo systemctl status create_hugepages
```

При корректной работе должно отображаться: `Finished Create hugepages`

4. Проверить статус сервиса *analyzer*:

```
sudo systemctl status analyzer
```

5. Убедиться, что после запуска трафика происходит запись метрик:

Без пароля для ClickHouse:

```
clickhouse-client --query "SELECT * FROM graphite WHERE Path LIKE '%analyzer%' LIMIT 10"
```

С паролем:

```
clickhouse-client --password=[пароль clickhouse] --query "SELECT * FROM graphite WHERE Path LIKE '%analyzer%' LIMIT 10"
```

В случае возникновения ошибок — просмотреть журнал сервиса:

```
sudo journalctl -fu analyzer
```

3.16 Создание SSH-пользователя

Для синхронизации и выполнения проверок веб-интерфейс устанавливает SSH-соединение с каждой системой Flowcollector.

Убедитесь что на каждой системе Flowcollector есть настроенный SSH-пользователь с доступом к `sudo`.

1. Создать нового пользователя:

```
sudo adduser fc-web
```

2. Добавить пользователя в группу sudo:

```
sudo usermod -aG sudo fc-web
```

3. Убедиться, что авторизация по SSH через пароль разрешена для этого пользователя.

3.17 Настройка NGINX

1. Удалить стандартную конфигурацию NGINX:

```
sudo rm /etc/nginx/sites-available/default /etc/nginx/sites-enabled/default
```

2. Создать файл конфигурации для FlowCollector:

```
sudo nano /etc/nginx/sites-available/flowcollector.conf
```

3. Вставить следующую конфигурацию:

```
server {
  listen 80 default_server;
  listen [::]:80 default_server;

  server_name REPLACE_ON_DOMAIN_OR_IP;

  location /broker {
    rewrite ^/broker(.*)$ $1 break;
    proxy_pass http://localhost:3335;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_cache_bypass $http_upgrade;
  }
}
```

Заменить `server_name REPLACE_ON_DOMAIN_OR_IP` на домен или IP-адрес.

4. Создать ссылку:

```
sudo ln -s /etc/nginx/sites-available/flowcollector.conf
/etc/nginx/sites-enabled
```

5. Перезапустить NGINX:

```
sudo systemctl restart nginx
```

3.18 Настройка веб-интерфейса

В зависимости от условий установки требуется сменить авторизационные данные и порты для базы данных и др. информацию в .env-файле. Сначала настраивается веб-интерфейс, после него - брокер.

1. Открыть для редактирования файл окружения веб-интерфейса:

```
sudo nano /opt/sp-spider/.env
```

2. Внести изменения в файл в соответствии с вашей конфигурацией:

```
VITE_APP_PORT=3333
NODE_ENV=production
HTTP_TIMEOUT=10000

# Если это основной интерфейс (даже при отсутствии резервирования)
IS_PRIMARY=true

# Секретный ключ. Нежелательно менять после первого запуска
APP_SECRET="salt_salt_salt"

# Данные от пользователя и БД postgresql
DB_HOST="localhost"
DB_PORT="5432"
DB_USER="spider"
DB_DATABASE="spider"
DB_PASSWORD="spider"

# Активация rabbitmq для синхронизации и брокера
RMQ_ENABLE="true"
RMQ_URL="amqp://spider:spider@localhost:5672"
RMQ_RECONNECT_INTERVAL="5000"
```

Использование AMQPs
В случае, если требуется поддержка TLS в рамках протокола AMQP, замените
`RMQ_URL="amqp://USER:PASSWORD@localhost:5672"`
на
`RMQ_URL="amqps://USER:PASSWORD@localhost:5672"`

3. Открыть для редактирования файл окружения брокера:

```
sudo nano /opt/sp-spider-broker/.env
```

4. Внести изменения в файл в соответствии с вашей конфигурацией:

```
# Порт, на котором запустится сервис
APP_PORT=3335

# Ключ из .env интерфейса, аналогичен /opt/sp-spider/.env
APP_SECRET="salt_salt_salt"

# Данные от базы данных из .env интерфейса
DB_HOST="localhost"
DB_PORT="5432"
DB_USER="spider"
DB_DATABASE="spider"
DB_PASSWORD="spider"

# Данные RabbitMQ из .env интерфейса
RMQ_URL="amqp://spider:spider@localhost:5672"
RMQ_RECONNECT_INTERVAL="5000"

# Путь к папке с политиками DosGate UH. Обязательно в конце
ставить "/"
POLICY_PATH="/var/lib/dosgate-uh/profiles/"

# Путь к конфигурации обработчика оффендеров DosGate UH
OFFENDERS_CONF_PATH="/opt/sp-spider-
broker/offenders/offenders.conf"

# Путь к объектам защиты Flowcollector. Обязательно в конце
ставить "/"
FC_MO_PATH="/opt/spfc/etc/mo/"

# Путь к симлинкам на объекты защиты Flowcollector. Обязательно в
конце ставить "/"
FC_MO_SYMLINK_PATH="/opt/spfc/etc/mo.enabled/"

# Путь к объектам обучения Treshold Learner. Обязательно в конце
ставить "/"
```

```
FC_LEARNER_PATH="/opt/spfc/etc/learner/"

# Путь к симлинкам на объекты обучения Treshold Learner.
Обязательно в конце ставить "/"
FC_LEARNER_SYMLINK_PATH="/opt/spfc/etc/learner.enabled/"

# Путь к конфигу dosgate-uh
DGUH_CONF="/etc/dosgate-uh.conf"

# Путь к снэпшотам дампов dosgate-uh
DGUH_SNAPSHOTS="/var/cache/dosgate-uh-snapshots"

#Путь к основному конфигурационному файлу анализатора
FC_ANALYZER_CONF_PATH="/opt/spfc/etc/analyzer.yaml"

# Путь к конфигу dosgate-uh
DGUH_CONF="/etc/dosgate-uh.conf"

# Путь к снэпшотам дампов dosgate-uh
DGUH_SNAPSHOTS="/var/cache/dosgate-uh-snapshots/"

#Путь к mmdb файлу
MMDB_PATH="/etc/dosgate/GeoLite2-Country.mmdb"

#Путь к дефолтному mmdb файлу
MMDB_DEFAULT_PATH="/usr/share/dosgate/GeoLite2-Country.mmdb"

# Путь до правил обработки syslog сообщений сервиса Rlog
RLOG_RULES_PATH="/var/lib/rlog/rules/"
```

3.19 Установка Grafana (пакет из репозитория Servicepipe)

1. Установить пакет Grafana:

```
sudo NEEDRESTART_MODE=a apt-get install grafana-enterprise -y
```

2. Открыть для редактирования файл конфигурации:

```
sudo nano /etc/grafana/grafana.ini
```

3. В секции `[database]` указать параметры подключения к PostgreSQL (см. шаг 3.9 Настройка PostgreSQL):

```
[database]
type = postgres
host = 127.0.0.1:5432
name = grafana
user = grafana
password = grafana
```

4. Запустить сервис Grafana и проверить его состояние:

```
sudo systemctl enable --now grafana-server && sudo systemctl
status grafana-server
```

5. Установить необходимые плагины:

```
sudo grafana-cli plugins install williamvenner-timepickerbuttons-
panel && \
sudo grafana-cli plugins install marcusolsson-json-datasource
```

3.20 Отключение механизма provisioning

Отключить механизм provisioning для сохранения изменений в дашбордах при перезапуске сервиса:

```
sudo sed -i 's|^\(provisioning = /etc/grafana/provisioning\)|#\1|'
/etc/grafana/grafana.ini && \
sudo systemctl restart grafana-server
```

3.21 Настройка Grafana (при использовании пакета из репозитория Servicepipe)

Открыть веб-интерфейс Grafana в браузере. Интерфейс доступен по порту 3000. По умолчанию используются учетные данные: *admin / admin*. При первом входе задать новый пароль, соответствующий требованиям информационной безопасности.

3.21.1 Конфигурация datasource

1. Настроить источник данных PostgreSQL `events`:

Открыть: **Connections → Data Sources → events → Save & Test**

Указать актуальный пароль:

```
Password: [значение из шага 3.9 Настройка PostgreSQL]
```

При изменении параметров подключения (например, если PostgreSQL размещён на другом хосте), обновить адрес подключения:

```
Connection: [IP-адрес сервера FC]:5432
(при локальной установке — `localhost:5432`)
```

Ожидаемый результат: статус подключения — **"Database Connection OK"**

2. Настроить источник данных `graphite`, если база Graphite размещена на отдельном хосте:

Открыть: **Connections → Data Sources → graphite → Save & Test**

Обновить URL подключения:

```
URL: http://[IP-адрес сервера FC]:8088
(при локальной установке — `http://localhost:8088`)
```

Ожидаемый результат: статус подключения — **"Data source is working"**

3. При размещении Grafana и API-сервисов `analyzer` и `reports` на разных хостах, обновить параметры подключения для `analyzer-api` и `reports-api`.

3.21.2 Конфигурация дашбордов

1. Перейти на вкладку **Dashboards**

2. Последовательно отредактировать все 5 дашбордов, пройдя по пути:

Dashboards → Analyzer → Edit → Settings → Variables → Hostname

В разделе переменных указать значение `hostname`, соответствующее параметру из секции `metrics` файла **/opt/spfc/etc/analyzer.yaml**

3. Установить флажок **Update default variable values**, затем нажать **Save dashboard** для сохранения изменений.

3.22 Настройка Grafana (в случае установки вне пакета Servicepipe)

Документация по настройке Grafana при установке вне дистрибутива Servicepipe размещены по ссылке.

3.23 Настройка компонента SP-events

1. Открыть для редактирования файл **/opt/sp-events/.env**:

> **Примечание**
> В качестве шаблона можно использовать **/opt/sp-events/.env.example**

```
# sp-events
EH_SERVER_PORT = 8081                                    # Порт
работы сервера sp-events
DASHBOARD_URL = "http://127.0.0.1:3000/d/fc-reports/reports" # URL
для просмотра отчета

TEMPLATE_FOLDER = "/opt/sp-events/template"              # Полный
путь к шаблонам сообщений

WATCHER_FOLDER_AUTO_DUMP = "/var/dump"                   # Полный
путь к папке для автоматических дампов
WATCHER_FOLDER_COMMAND_DUMP = "/opt/sp-events/dumps"     # Полный
путь к папке для ручных дампов через Telegram

SMTP_USED = false                                        #
Включение/выключение использования SMTP
SMTP_SENDER_MAIL = "test@mail.ru"                        # Почта
SMTP
SMTP_PASSWORD = "YOUR_MAIL_PASSWORD"                     # Пароль
SMTP
SMTP_SERVER = "smtp.mail.ru"                             # Сервер
```

```
SMTP
SMTP_PORT = 465                                          # Порт SMTP
MAIL_RECEIVERS = test@servicepipe.ru, test@yandex.ru     # Адреса
почт для уведомлений

SQLITE_PATH = "/opt/sp-events/sqlite.db"                 # Полный
путь к файлу db

TG_USED = false                                          #
Включение/выключение использования Telegram
CHATS_IDS_DUMP = "345389346,2239564"                     # ID чатов
для отправки автоматических дампов
CHATS_IDS_EVENT = "-1002416780921"                       # ID чатов
для отправки сообщений о начале/окончании атак
TG_PORT = 3009                                           # Порт для
работы Telegram сервера для отправки сообщений о событиях
TG_TOKEN = "7656101759:test"                             # Токен
Telegram-бота
TG_KEEP_ALIVE = 10                                       # Keep-
alive параметр для отправки при создании дампа

DUMP_API_URL = "http://localhost:5001/sp-events"         # URL для
создания дампа

# reports
REPORTS_SERVER_PORT = 9080                               # Порт
работы сервера Reports
RATE = 1                                                 # Sample
rate из /opt/spfc/etc/analyzer.yaml
MONGO_HOST = 127.0.0.1
MONGO_PORT = 27017
MONGO_DATABASE = test
MONGO_USERNAME = events
MONGO_PASSWORD = events

# NATS В случае, если используется NATS в
/opt/spfc/etc/analyzer.yaml
NATS_URL = nats://localhost:4222                  # URL Nats
NATS_EVENT_SUBJ = analyzer.notification          # Nats event
subject

# Postgres
POSTGRES_USED = true
POSTGRES_HOST = localhost
POSTGRES_PORT = 5432
POSTGRES_USER = spider
POSTGRES_DATABASE = spider
POSTGRES_PASSWORD = spider
```

2. Настроить шаблоны сообщений:

Отредактировать шаблоны Telegram:

```
sudo nano /opt/sp-events/template/tg/
```

Отредактировать шаблоны e-mail:

```
sudo nano /opt/sp-events/template/mail/
```

Указать URL дашборда Reports в Grafana для корректного отображения ссылок в сообщениях.

3. Запустить сервис *sp-events* и проверить его состояние:

```
sudo systemctl enable --now sp-events && systemctl status sp-
events
```

3.24 Финальный этап: запуск и проверка состояния сервисов

1. Перезапустить брокер сообщений *sp-spider-broker*:

```
sudo systemctl restart sp-spider-broker
```

2. Запустить и активировать при загрузке основные сервисы:

```
sudo systemctl enable --now sp-spider sp-spider-broker
```

3. Проверка состояния сервиса RabbitMQ:

```
sudo systemctl status rabbitmq-server
```

4. Проверка состояния PostgreSQL:

```
sudo systemctl status postgresql
```

5. Проверка состояния NGINX:

```
sudo systemctl status nginx
```

6. Проверка состояния основного сервиса SP-Spider:

```
sudo systemctl status sp-spider
```

7. Проверка состояния сервиса SP-Spider Broker:

```
sudo systemctl status sp-spider-broker
```

Инструкция по обновлению

1. Остановка основного компонента FlowCollector *analyzer*

Перед обновлением рекомендуется остановить основной сервис во избежание конфликтов и для корректного применения изменений:

```
sudo systemctl stop analyzer
```

2. Обновление основного компонента *analyzer*

Выполнить следующую команду:

```
sudo apt-get install analyzer
```

Если в рамках релиза были обновлены вспомогательные компоненты, например *sp-events*, выполнить их обновление:

```
sudo apt-get install sp-events
```

Актуальный список компонентов, входящих в состав системы, приведён по следующей ссылке. Он может быть использован для проверки состава установленных модулей и планирования обновлений.

3. Обновление файлов конфигурации

При наличии новых параметров в релизе, необходимо обновить файл конфигурации вручную, сохраняя существующие настройки:

```
sudo nano /opt/spfc/etc/analyzer.yaml
```

Если изменения затрагивают конфигурацию сопутствующих компонентов (например, *sp-events*), необходимо также внести соответствующие правки в их конфигурационные файлы:

```
sudo nano /opt/sp-events/.env   # конфигурационный файл модуля sp-
```

```
events
```

4. Запуск сервиса *analyzer*

После внесения изменений в конфигурацию необходимо снова запустить основной сервис:

```
sudo systemctl start analyzer
```

5. Проверка состояния сервисов

Убедиться в стабильной работе компонентов после обновления. Проверить статус каждого сервиса, который был установлен или обновлён:

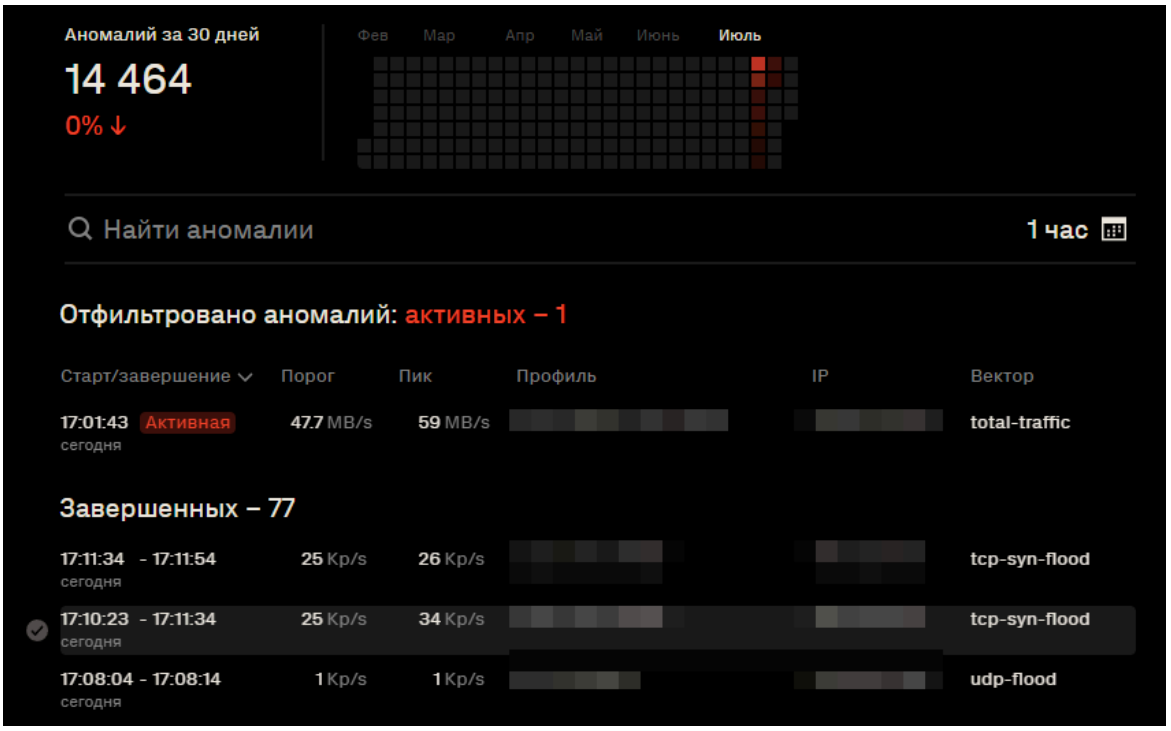
```
sudo systemctl status analyzer
```

```
sudo systemctl status sp-events    # пример дополнительного
компонента
```

Дашборд

Дашборд предназначен для оперативного мониторинга сетевой активности, анализа аномалий и визуализации трафика защищаемых объектов в режиме реального времени.

Рабочая область Дашборд:



Общая статистика по аномалиям

В верхней части интерфейса отображается количество сетевых аномалий, зафиксированных за последние 30 дней, а также процентное изменение по сравнению с предыдущим периодом. Дополнительно отображается тепловая карта активности по дням за последние полгода.

Поиск аномалий

Для фильтрации аномалий доступен текстовый поиск по профилю, вектору атаки и IP-адресу. Справа доступен фильтр временного диапазона с предустановленными интервалами и возможностью ручного выбора начала и конца периода.

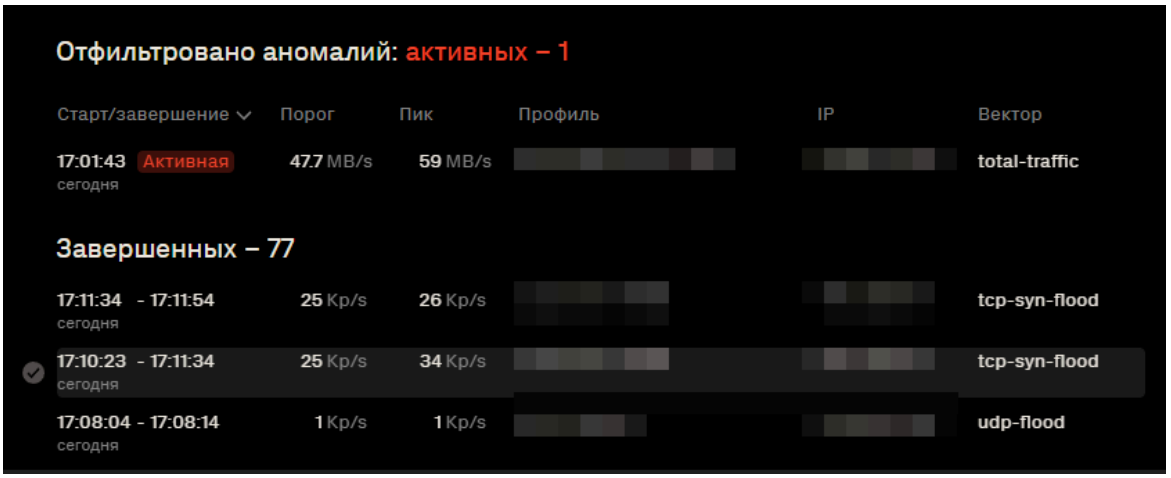
Список аномалий

Блок содержит таблицу с перечнем зафиксированных аномалий. Аномалии разделяются на два типа:

- **Активные аномалии** — события, в данный момент превышающие заданные пороговые значения.
- **Завершённые аномалии** — ранее зафиксированные события, уже завершённые по текущему профилю.

Для каждой аномалии отображаются следующие параметры:

- **Старт/завершение** — временные метки начала и окончания события (если завершено).
- **Порог** — заданное пороговое значение пропускной способности.
- **Пик** — максимальное зафиксированное значение, приведшее к срабатыванию аномалии.
- **Профиль** — название объекта.
- **IP** — IP-адрес назначения, по которому зафиксирована аномалия.
- **Вектор атаки** — классификация аномалии (например, *udp-flood*, *icmp-flood*, *total-traffic*).



Визуализация трафика

В интерфейсе представлены четыре графика, отображающих статистику по трафику защищаемых объектов:

- **Total Bits**
  Отображает общий объем входящего трафика по всем объектам в битах за выбранный временной период.
- **Total Packets**
  Показывает общее количество сетевых пакетов в секунду.
- **Top 12 Managed Objects by Bits**
  Отображает топ-12 объектов с наибольшим объемом входящего трафика в битах. Каждый объект представлен отдельной линией на графике. Цветовая легенда расположена справа.
- **Top 12 Managed Objects by Packets**
  Отображает топ-12 объектов по количеству сетевых пакетов в секунду.

FlowSpec-правила

FlowSpec-правила — это механизм BGP Flow Specification для распространения фильтрационных политик на маршрутизаторы.

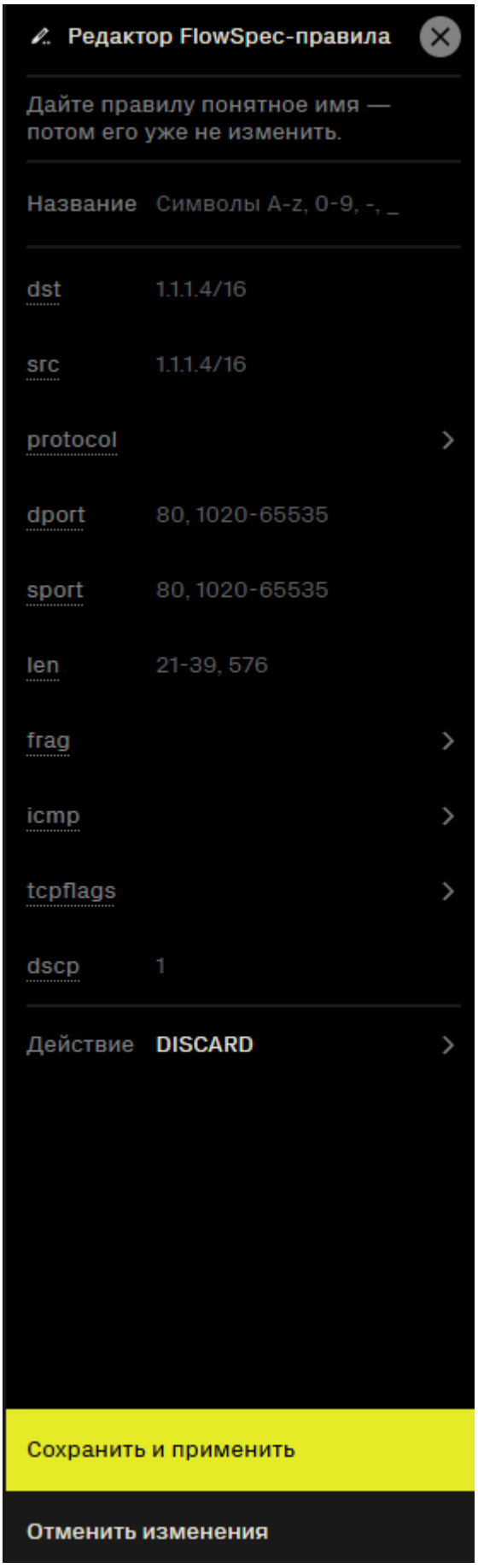
Область применения

FlowSpec-правила применяются для динамической фильтрации на уровне маршрутизаторов в ситуациях, когда требуется быстро ограничить, перенаправить или маркировать трафик без изменений в нижестоящих сервисах. Исполнение происходит на границе сети, до системы очистки; при DDoS-атаках правила позволяют отсечь вредоносный поток на входе и не допустить перегрузки очистителя.

В инфраструктурах без выделенного очистителя FlowSpec-правила выступают основным механизмом быстрой фильтрации. Также используются как дополнительный защитный слой: первичная отсечка нежелательного трафика выносится на вышестоящие BGP-маршрутизаторы, что разгружает внутренние узлы и каналы.

Добавление FlowSpec-правила

Для добавления нового правила в правой части интерфейса **нажать** серую кнопку **«Добавить правило»**. Откроется редактор **FlowSpec-правил**, содержащий три блока: **Название**, **Совпадение** и **Действие**. В блоке **Совпадение** указываются условия отбора трафика, в блоке **Действие** — способ обработки трафика маршрутизатором. Для применения изменений нажать жёлтую кнопку **Сохранить и применить**.


Редактор FlowSpec-правила
Дайте правилу понятное имя — потом его уже не изменить.
Название
Символы A-z, 0-9, -, _
dst
1.1.1.4/16
src
1.1.1.4/16
protocol
dport
80, 1020-65535
sport
80, 1020-65535
len
21-39, 576
frag
icmp
tcpflags
dscp
1
Действие
DISCARD
Сохранить и применить
Отменить изменения

Название

- Допустимые символы: A–Z, 0–9, дефис -, подчёркивание _.
- Рекомендуем давать правилам краткие и однозначные имена, отражающие их назначение. Примеры: *udp-amp-auto*, *ipfrag-drop*, *invalid-ports*

Совпадение

Совпадения — это техническое условие, по которому маршрутизатор выбирает трафик для применения правила.

Для большинства совпадений доступен флаг **NOT**. Он выполняет логическое отрицание: при включении условие работает наоборот, и правило применяется ко всем значениям *кроме указанного*.

> **Примечание**
> Для полей **dst** и **src** флаг **NOT** отсутствует. Отрицание не поддерживается

dst — IP назначения

Параметр	Описание
Адрес	Адрес назначения (IPv4/IPv6 в формате CIDR)

Если поле **оставить пустым — правило используется как шаблон**. Во время аномалии адрес назначения автоматически подставляется из события/счётчика.

src — IP источника

Параметр	Описание
Адрес	Адрес источника (IPv4/IPv6 в формате CIDR)

protocol — Протокол

Протокол	Описание
TCP	Протокол управления передачей (Transmission Control Protocol)
UDP	Протокол пользовательских дейтаграмм (User Datagram Protocol)
ICMP	Протокол управления интернет-сообщениями (Internet Control Message Protocol)
IGMP	Управление членством в мультикаст-группах для IPv4
IP-in-IP	Инкапсуляция IP-пакетов в IP (туннелирование без шифрования)
RSVP	Резервирование сетевых ресурсов и сигнализация QoS
GRE	Протокол инкапсуляции (Generic Routing Encapsulation)
OSPF	Внутренний протокол маршрутизации (IGP) на основе состояния каналов
PIM	Протокол независимой мультикаст-маршрутизации
SCTP	Протокол управления потоками сообщений (Stream Control Transmission Protocol)

dport — Порт назначения

Параметр	Описание
dport	Порты назначения для сопоставления трафика. Формат: значения и/или диапазоны 0–65535, через запятую. Примеры: 80; 53, 123; 1020–65535

sport — Порт источника

Параметр	Описание
sport	Порты источника для сопоставления трафика. Формат: значения и/или диапазоны 0–65535, через запятую. Примеры: 0, 17, 19, 69, 123; 49152–65535

len — Длина пакета

Параметр	Описание
len	Длина IP-пакета в байтах. Формат: значения и/или диапазоны, через запятую. Примеры: 1280; 60–80; 576

frag — Фрагментация

Значение	Описание
any	Любой фрагмент (первый, внутренний, последний)
df	Установлен флаг Don’t Fragment
first	Первый фрагмент
internal	Внутренний фрагмент
last	Последний фрагмент
unfrag	Не фрагментированный пакет

icmp — Тип/код

Проверяются сочетания *тип–код*: правило срабатывает только на пакеты, где **тип** и **код** встречаются одновременно; недопустимые комбинации игнорируются.

Параметр	Описание
icmp-тип	Значения или диапазоны типа ICMP. Примеры: 8 (Echo Request), 11 (Time Exceeded)
icmp-код	Значения или диапазоны кода ICMP. Примеры: 0; 1

tcpflags — Флаги TCP

Флаг	Описание
syn	Инициализация соединения
ack	Подтверждение соединения
psh	Передача без задержки
rst	Принудительный сброс
fin	Завершение соединения
ece	Индикация перегрузки (ECN Echo)
cwr	Снижение окна при перегрузке (Congestion Window Reduced)
urg	Срочные данные (Urgent)

Опция **Учитывать совместно (+)** задаёт логику проверки TCP-флагов:

- Если опция включена, правило срабатывает только когда все выбранные флаги выставлены одновременно.
- Если опция выключена, достаточно любого одного из выбранных флагов.

dscp — Класс обслуживания

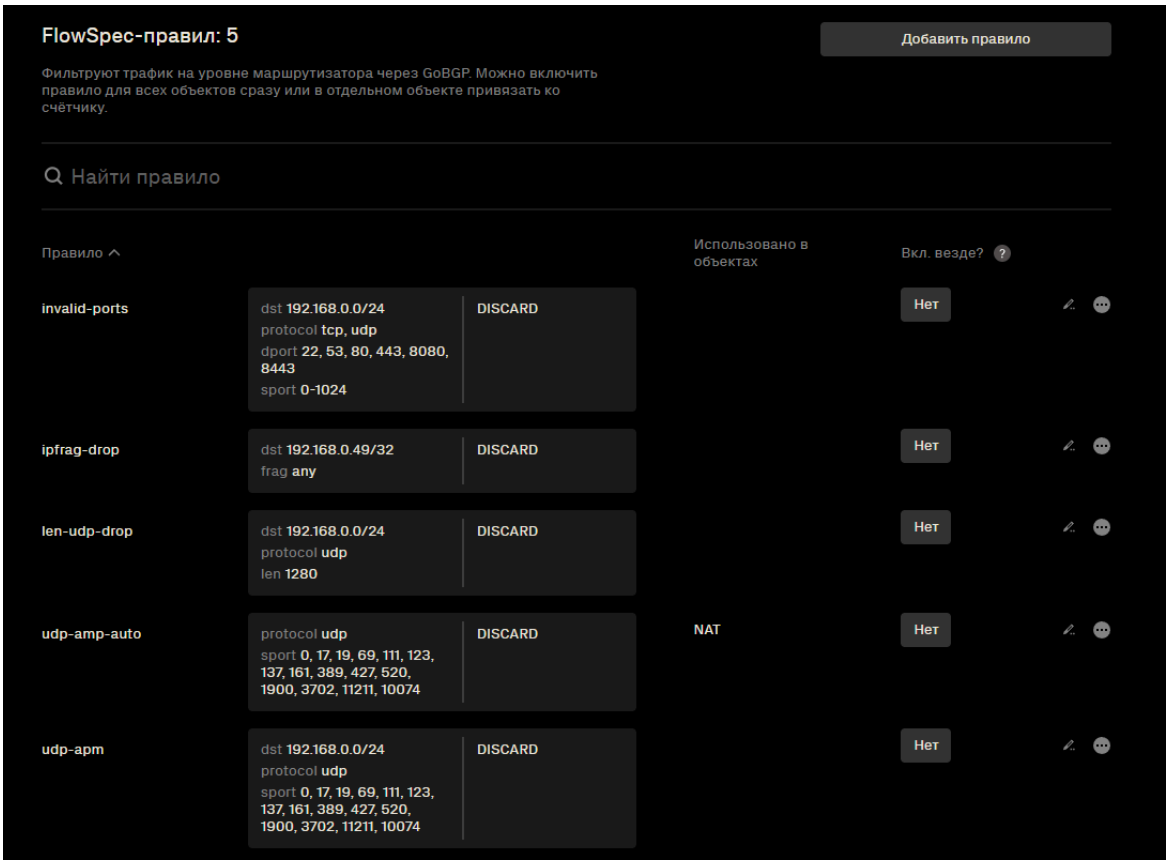
Параметр	Описание
dscp	Класс обслуживания DSCP, диапазон 0–63. Пример: 46

Действие

Определяет реакцию маршрутизатора на трафик, попавший под условия.

Параметр	Аргумент	Описание
ACCEPT	—	Разрешить трафик
DISCARD	—	Отбросить трафик
rate-limit	—	Ограничение скорости; трафик выше порога отбрасывается
	rate-limit	Порог скорости; единицы — по конфигурации системы
	AS	Номер автономной системы для сообщества traffic-rate
redirect	—	Перенаправление в целевой VRF
	route-target (rt)	Формат: **as:vrf** или **cidr:vrf**, например 65000:123
mark	—	Маркировка DSCP для последующей приоритезации
	dscp	Значение 0–63
action sample	—	Семплирование и логирование трафика
action terminal	—	Прекращение дальнейшей фильтрации для совпавшего трафика
action sample-terminal	—	Семплировать и завершить дальнейшую фильтрацию (комбинированное действие)

Примеры FlowSpec-правил



Ниже представлены примеры FlowSpec-правил. Для каждого правила указаны условия совпадения и действие маршрутизатора. Примеры позволяют понять, какой трафик фильтруется и каким образом правило применяется в сети.

- **invalid-ports — DISCARD**
  Отбросить TCP/UDP-трафик к подсети 192.168.0.0/24 на неразрешённые по условию порты (22, 53, 80, 443, 8080, 8443; возможны диапазоны). Назначение: блокировка несанкционированных подключений и сканирований.
- **ipfrag-drop — DISCARD**
  Отбросить фрагментированные IP-пакеты к адресу 192.168.0.49/32. Назначение: защита от атак с использованием фрагментации.
- **len-udp-drop — DISCARD**
  Отбросить UDP-пакеты фиксированной длины 1280 байт к сети 192.168.0.0/24. Назначение: подавление сигнатурных шумов в рамках DDoS.
- **udp-amp-auto — DISCARD**
  Отбросить UDP-трафик с характерных исходных портов усилителей (0, 17, 19, 69, 111, 123, 137, 161, 389, 427, 520, 1900, 3702, 11211, 10074).

При пустом поле **dst** правило используется как шаблон: при срабатывании счётчика система подставляет адрес назначения из события и формирует анонс FlowSpec для этого адреса. Действие применяется точечно к подставленному **dst** и не действует глобально само по себе.

- **udp-apm — DISCARD**
  Отбросить UDP-амплификации в подсети 192.168.0.0/24 по тем же исходным портам. Назначение: локальная фильтрация в заданной сети.

В столбце **Использовано в объектах** указывается, к каким объектам системы привязано правило.

Столбец **Вкл. везде?** — ручная публикация анонса FlowSpec. При включении система немедленно формирует и отправляет BGP FlowSpec-анонс правила на маршрутизаторы. Пороги и счётчики объектов при этом не учитываются.

Data Explorer

Data Explorer — модуль системы FlowCollector для анализа данных сетевых потоков. Он сохраняет весь собранный трафик в базе данных и обеспечивает возможности для его анализа и визуализации.

С помощью модуля можно задать фильтры по параметрам трафика (протоколы, порты, IP-адреса, длина пакета и т.д) и получить статистику и графические отчеты. Это позволяет выявлять аномальные или интересующие паттерны трафика, а так же проводить ретроспективный анализ трафика.

Для отображения раздела Data Explorer в веб-интерфейсе требуется включённый параметр save-flow в конфигурации анализатора FlowCollector.

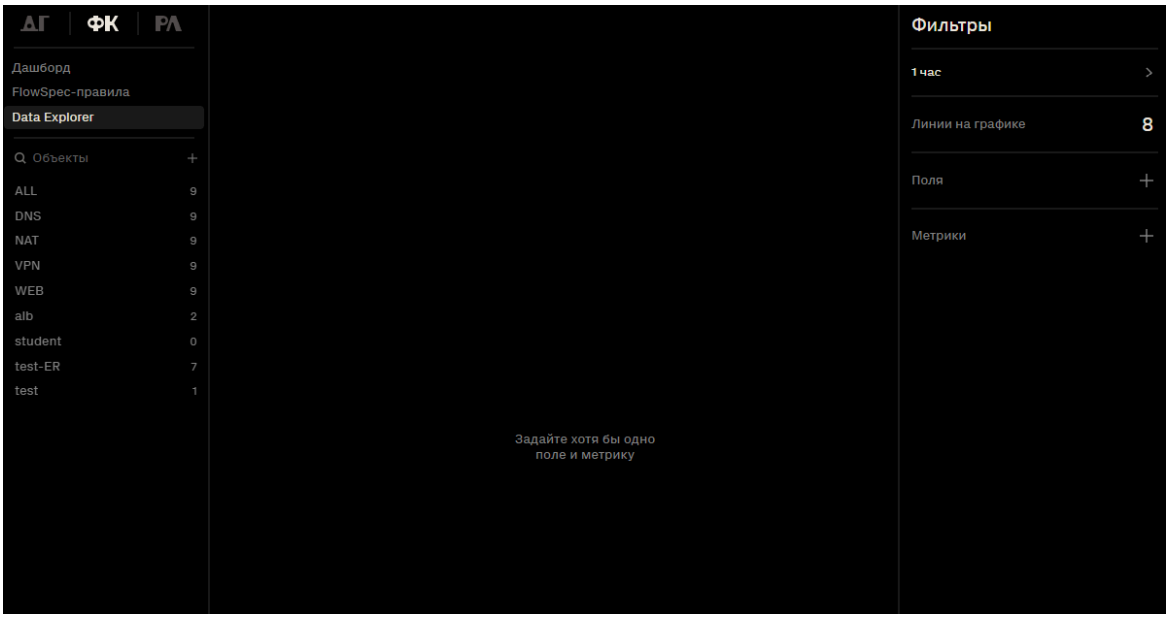
Открыть конфигурационный файл анализатора:

```
sudo nano /opt/spfc/etc/analyzer.yaml
```

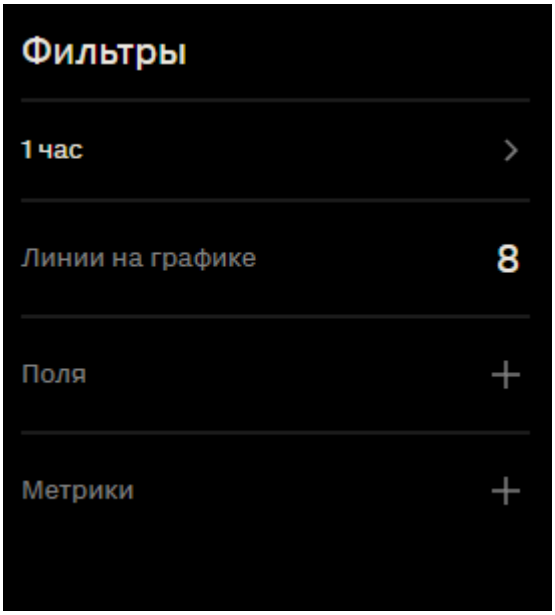
Включить параметр save-flow:

```
save-flow: true
```

Для доступа к модулю выберите **Data Explorer** в боковом меню интерфейса. Затем откроется рабочая область модуля.

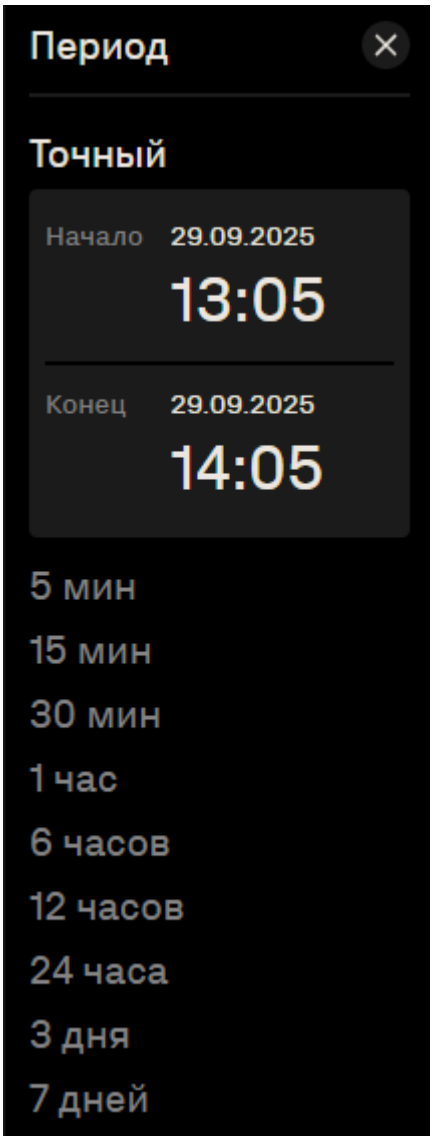


Для построения графиков в **Data Explorer** необходимо предварительно задать параметры фильтра.



Период

Период - это временной интервал, за который выводятся данные. Его можно задать вручную, указав начальное и конечное время, или выбрать из готовых вариантов, например, последние 24 часа или 7 дней.



Линии на графике

Линии на графике — параметр, определяющий число отображаемых линий графика. На график выводятся верхние *N* строк таблицы по текущей сортировке. Каждая строка отображается отдельной линией на графике.

Поля

Поля - это атрибуты данных, по которым производится агрегация и фильтрация. Они определяют, какие именно характеристики данных будут использоваться для анализа или отображения. В системе можно выбрать несколько атрибутов для более детального анализа.

Поле	Описание
Das	Номер автономной системы получателя
Dasname	Имя автономной системы получателя
Dport	Порт получателя
Dst	IP-адрес получателя
Geoipdst	Страна получателя
Geoipsrc	Страна отправителя
Len	Длина пакета
Protocol	Протокол
Sas	Номер автономной системы отправителя
Sasname	Имя автономной системы отправителя
Sport	Порт отправителя
Src	IP-адрес отправителя
Tcpflags	TCP флаги
Tos	Тип обслуживания

Метрики

Метрики – это показатели, которые вычисляются на основе значений выбранных полей и отражают количественные характеристики данных в сети. Метрики позволяют анализировать и оценивать различные параметры сетевого трафика такие как количество переданных данных, количество пакетов, географическое распределение трафика и другие показатели.

Метрика	Описание
Bits	Биты
Das	Номер автономной системы получателя
Dport	Порт получателя
Dst	IP-адрес получателя
Geoipdst	Страна получателя
Geoipsrc	Страна отправителя
Packets	Пакеты
Sas	Номер автономной системы отправителя
Sport	Порт отправителя

Метрика	Описание
Src	IP-адрес отправителя

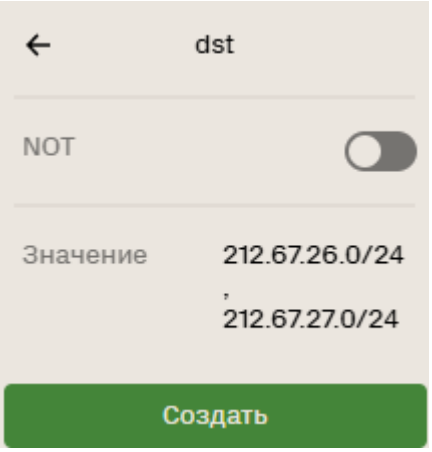
При выборе метрик необходимо указать дополнительный параметр:

Параметр	Описание
Avg	Среднее значение
Max	Максимальное значение
Percentile95	95-й перцентиль
Percentile99	99-й перцентиль
Total	Общее значение

Допустимо выбрать несколько параметров. Параметр **Total** установлен по умолчанию и не может быть отключён.

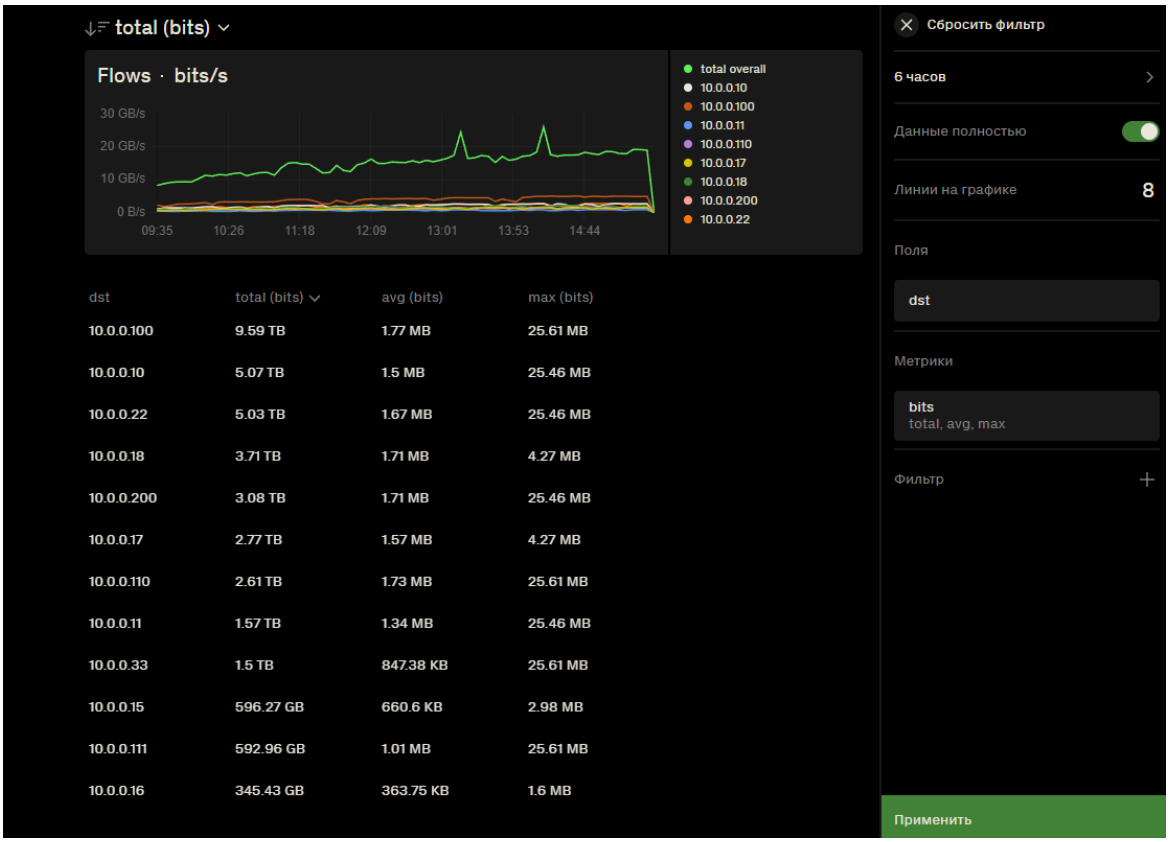
Фильтр

Результаты можно отфильтровать по выбранным атрибутам **Поля**.
При выборе параметра откроется меню, в котором нужно указать значения для фильтрации. Несколько значений можно указать через запятую без пробелов. Переключатель «NOT» - является логическим операндом «НЕ», при его активации будут отображены все значения, кроме указанного.



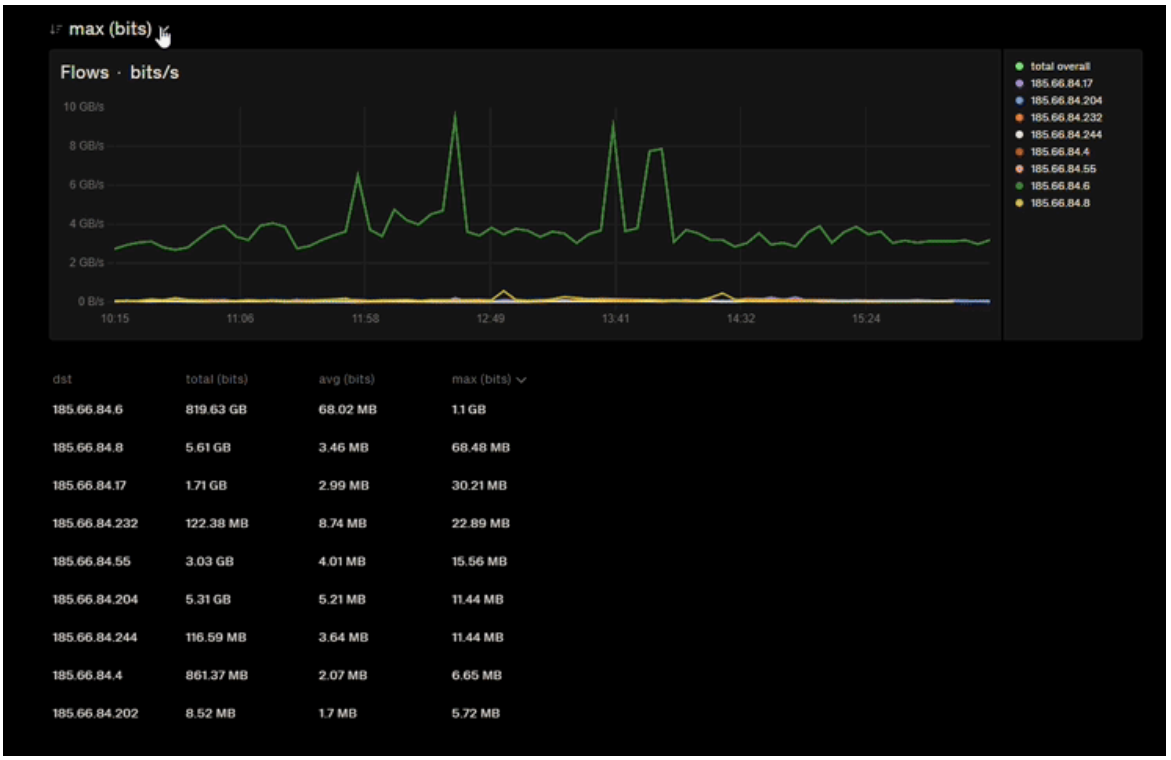
После того как все необходимые параметры будут указаны, необходимо нажать зелёную кнопку **Применить**.

В результате на рабочей области появится график и таблица с данными:



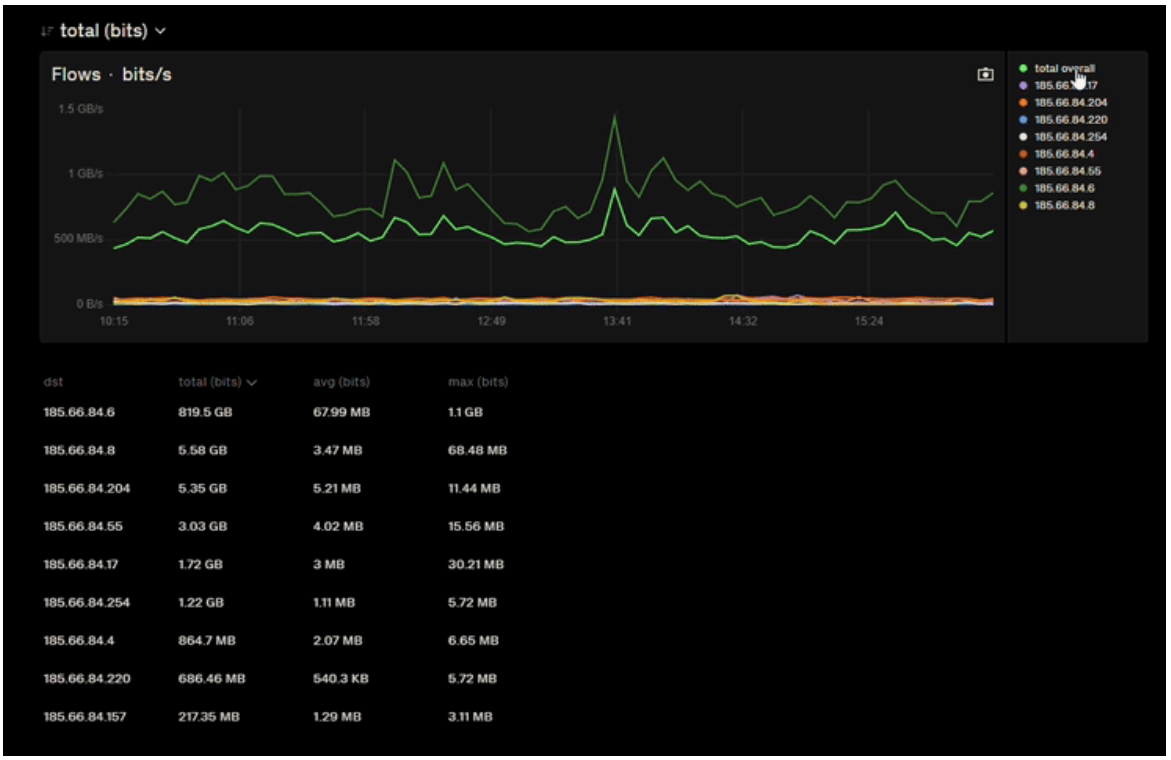
На графике представлено количество переданных бит данных в течение шести часов. Под графиком расположена таблица, в которой указаны IP-адрес назначения, общее количество переданных бит, среднее и максимально значение переданных данных.

Если в метрике было выбрано несколько параметров, то возможно переключаться между графиками выбирая нужный параметр.



При сортировке параметров в таблице под графиком также отображается график этого параметра.

По умолчанию на графике отображаются все значения атрибутов. Для отображения конкретного атрибута необходимо выбрать его в правой части интерфейса графика.



dst	total (bits)	avg (bits)	max (bits)
185.66.84.6	819.5 GB	67.99 MB	1.1 GB
185.66.84.8	5.58 GB	3.47 MB	68.48 MB
185.66.84.204	5.35 GB	5.21 MB	11.44 MB
185.66.84.55	3.03 GB	4.02 MB	15.56 MB
185.66.84.17	1.72 GB	3 MB	30.21 MB
185.66.84.254	1.22 GB	1.11 MB	5.72 MB
185.66.84.4	864.7 MB	2.07 MB	6.65 MB
185.66.84.220	686.46 MB	540.3 KB	5.72 MB
185.66.84.157	217.35 MB	1.29 MB	3.11 MB

Управление объектами FlowCollector

Объект — это набор IP-масок получателя, используемых для определения соответствия трафика заданным порогам.

Создание и редактирование объектов через веб-интерфейс

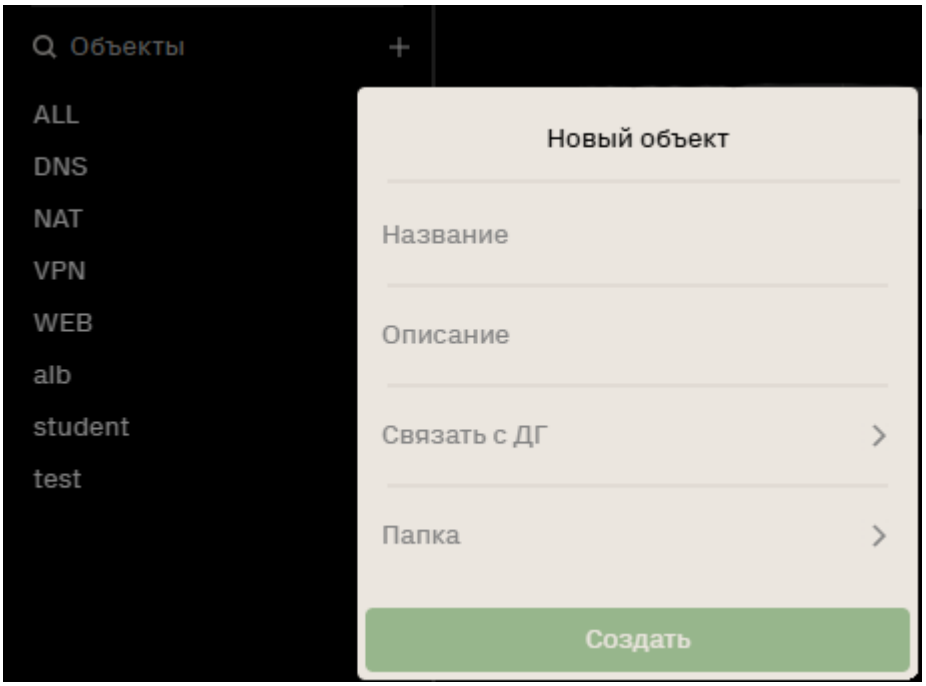
Создание объекта

Для создания нового объекта в системе выполнить следующие действия:

1. На главной странице в разделе **Объекты** нажать кнопку "+".

2. Заполнить следующие поля для создания объекта:

- **Название** - уникальное имя для объекта. Рекомендуется использовать комбинацию из обозначения сегмента инфраструктуры и названия сервиса, например, "zapadniy-filial-web" или "dmz-dns".
- **Описание** - краткое текстовое пояснение, которое поможет понять назначение объекта.
- **Связать с ДГ** - при наличии интеграции между FlowCollector и DosGate объект может быть привязан к профилю DosGate для корреляции событий и мониторинга трафика при обнаружении аномалий.
- **Папка** - для удобства объекты можно сгруппировать в папку, выбрав существующую или создав новую через опцию "Создать папку".



Изменение объекта

Для изменения объекта необходимо нажать правой кнопкой мыши на его названии в общем списке объектов и выбрать пункт **Изменить**. Откроется окно редактирования счётчиков.

Удаление объекта

В объекте предусмотрена возможность его удаления. Для удаления объекта необходимо нажать правой кнопкой мыши на его названии в общем списке объектов и выбрать пункт **Удалить**. После этого объект будет безвозвратно удален из системы.

Создание объекта через конфигурационный файл

Конфигурационный файл — это ещё один способ задания и управления параметрами объектов системы. Каждый объект системы настраивается индивидуально с помощью отдельного YAML-файла, который определяет

ключевые параметры и пороговые значения для мониторинга и защиты сети.

Конфигурационные файлы объектов размещаются по пути:

```
/opt/spfc/etc/mo/
```

Допускается создание вложенных директорий для организации конфигураций. Например, конфигурации веб-сервисов можно структурировать следующим образом:

```
/opt/spfc/etc/mo/web/
/opt/spfc/etc/mo/filial1/service1.yaml
/opt/spfc/etc/mo/filial2/service2.yaml
```

Создание конфигурационного файла объекта

Для создания нового объекта выполнить следующую команду:

```
sudo nano /opt/sfpc/etc/mo/new-object.yaml
```

Структура конфигурационного файла объекта

```
name: new-object                                    # Имя объекта,
чувствительно к регистру

cidrs:                                              # IP-маски
получателей объекта. Маска 0.0.0.0/0 не поддерживается
- 1.1.1.0/24
- 1.1.2.0/24

rules:                                              # Блок правил
(порогов)
-                                                   # Разделитель для
правила
  type:                                             # Ключ подсчета
трафика, global или local. Local - счетчик для каждого /32 в
рамках IP-маски, то-есть, ключем является /32. Используется для
детекции точных атак. Может быть оба сразу
```

```
  - local
  units:                                    # Единицы измерения.
bytes или packets. Может быть оба сразу
  - bytes
  vectors:                                  # Векторы. Может быть
несколько векторов сразу
 - total-traffic                            # Любой сетевой пакет
(IPv4/IPv6)
  # - dns-flood                             # UDP или TCP, порт
получателя 53
  # - ip-fragment-flood                     # Пакет с
установленным битом фрагментации
  # - icmp-flood                            # Протокол ICMP
(только IPv4)
  # - tcp-flood                             # Протокол TCP
  # - tcp-cwr-flood                         # TCP с флагом CWR
(Congestion Window Reduced, 0x80)
  # - tcp-ece-flood                         # TCP с флагом ECE
(ECN-Echo, 0x40)
  # - tcp-urg-flood                         # TCP с флагом URG
(Urgent Pointer, 0x20)
  # - tcp-ack-flood                         # TCP с флагом ACK
(Acknowledgment, 0x10)
  # - tcp-psh-flood                         # TCP с флагом PSH
(Push Function, 0x08)
  # - tcp-rst-flood                         # TCP с флагом RST
(Reset, 0x04)
  # - tcp-syn-flood                         # TCP с флагом SYN
(Synchronize, 0x02)
  # - tcp-fin-flood                         # TCP с флагом FIN
(No more data, 0x01)
  # - udp-flood                             # Протокол UDP
  # - total-traffic                         # Любой сетевой пакет
(IPv4/IPv6)
  # - invalid-protocol-flood                # Пакет с
идентификатором протокола 0
  # - chargen-amp                           # UDP, порт источника
19
  # - dns-amp                               # UDP, порт источника
53
  # - ntp-amp                               # UDP, порт источника
123
  # - snmp-amp                              # UDP, порт источника
161
  # - snmptrap-amp                          # UDP, порт источника
162
  # - ldap-amp                              # UDP, порт источника
389
  # - mssql-amp                             # UDP, порт источника
1434
  # - ibm-cics-amp                          # UDP, порт источника
```

```
1435
  # - ssdp-amp                                  # UDP, порт источника
1900
  # - apple-remote-desktop-amp                  # UDP, порт источника
3283
  # - ws-discovery-amp                          # UDP, порт источника
3702
  # - memcached-amp                             # UDP, порт источника
11211
  # - udp-zero-payload-flood                    # UDP-флуд с нулевой
полезной нагрузкой
  # - udp-big-packets-flood                     # UDP-флуд с крупными
по размеру пакетами
  limit-threshold: 250000000                  # Тип порога. Их
несколько:
                                              #   Limit-threshold -
статический порог, в выбранных единицах измерения (байтах или
пакетах)
                                              #   limit-diff -
разница между предыдущим значением (1000 мс.) и настоящим,
например, увеличение за секунду на 50000 условных единиц
                                              #   Limit-reldiff -
разница между предыдущим значением (1000 мс.) и настоящим,
например, увеличение за секунду в 5 раз (множитель)
                                              #   Можно использовать
сразу несколько, тогда любое из превышенных считается аномалией
- type:                                       # Ключ подсчета
трафика
  - global
  units:                                      # Единицы измерения
  - packets
  vectors:                                    # Векторы
  - total-traffic
  - tcp-syn-flood
  limit-diff: 50000
  limit-threshold: 200000
```

Пример конфигурации объекта

```
name: web-services                            # Имя объекта

cidrs:                                        # IP-маски получателей
объекта
- 1.1.1.0/24

rules:                                        # Блок правил
(порогов)
```

```
-                                          # Объем трафика по
любому из векторов в сторону объекта
  type:
  - global                                 # Тип подсчета —
глобальный
  units:
  - bytes                                  # Единицы измерения —
байты
  limit-threshold: 2000000000              # Статический порог по
суммарному трафику

-                                          # Объем трафика по
любому из векторов в сторону любого /32
  type:
  - local                                  # Тип подсчета —
локальный (по IP)
  units:
  - bytes                                  # Единицы измерения —
байты
  limit-threshold: 500000000               # Статический порог на
каждый IP

-                                          # DNS-flood
  type:
  - local                                  # Локальный и
глобальный подсчет
  - global
  units:
  - bytes                                  # Единицы измерения —
байты
  vectors:
  - dns-flood                              # Тип атаки — DNS-флуд
  limit-threshold: 100000000               # Статический порог по
DNS-флуду
```

Активация объекта

Для активации объекта необходимо создать символическую ссылку на конфигурационный файл в директории активированных объектов:

```
sudo ln -s /opt/spfc/etc/mo/<file> /opt/spfc/etc/mo.enabled/<file>
```

Перезапустить сервис *analyzer* для активации объекта:

```
sudo systemctl start analyzer
```

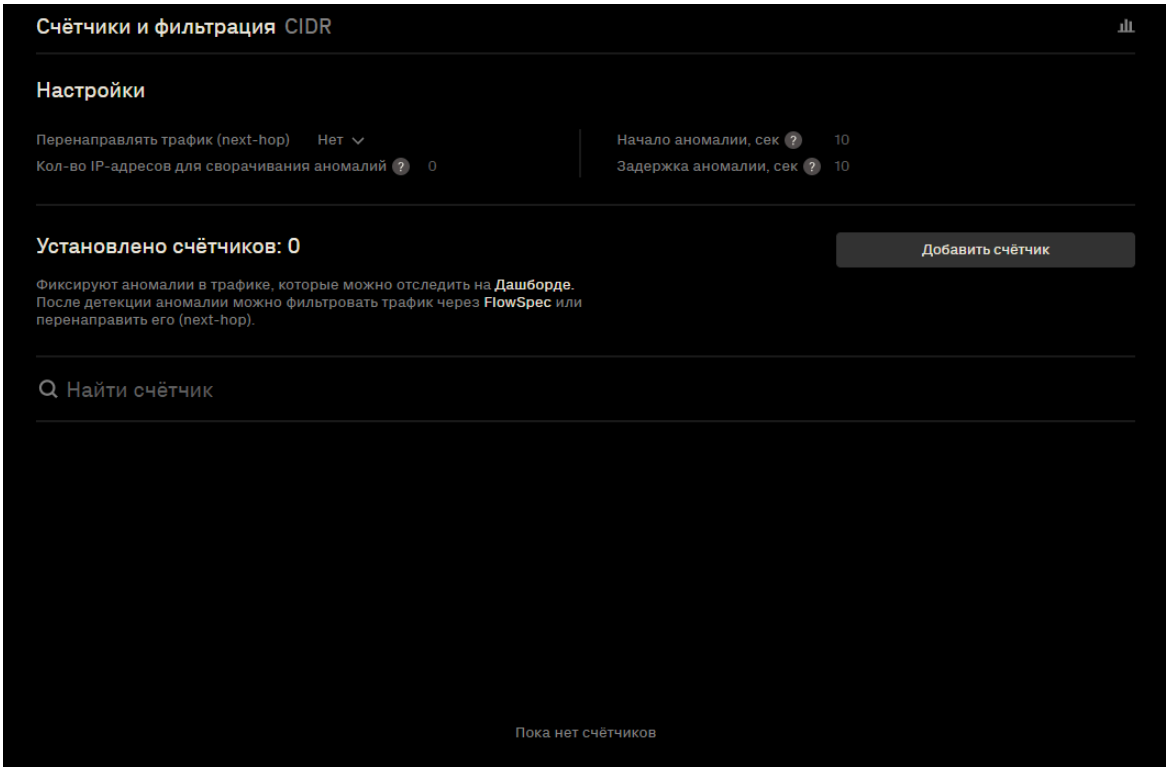
Проверить текущий статус службы:

```
sudo systemctl status analyzer
```

Активация объекта считается завершённой после успешного запуска службы и отсутствия ошибок в статусе.

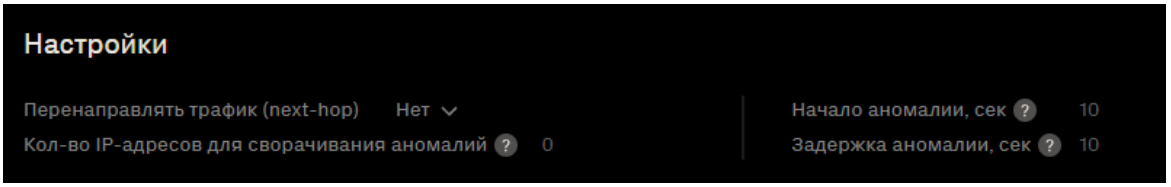
Счетчики и фильтрация

Раздел предназначен для управления счётчиками и фильтрацией зафиксированных аномалий. Данные о событиях доступны на **Дашборде** и могут обрабатываться через **FlowSpec-правила** или перенаправляться на **следующий хоп (next-hop)**.



Настройки

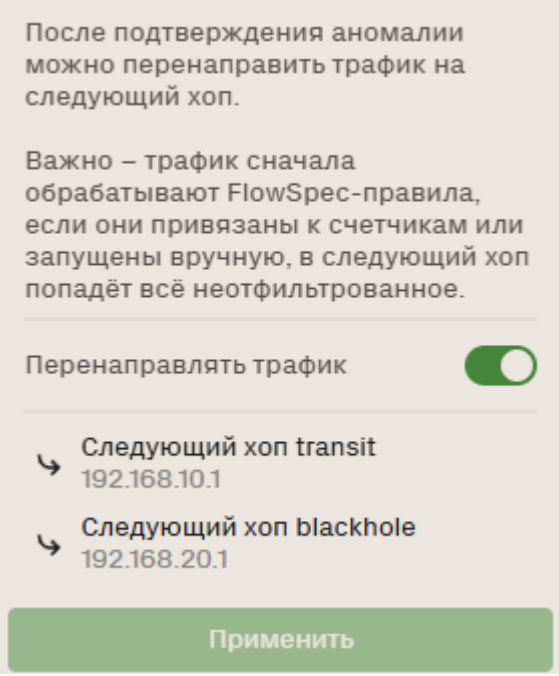
Раздел предназначен для управления параметрами обработки подтверждённых аномалий. Здесь можно задать условия фильтрации и выбрать действия при подтверждении аномалии.



Перенаправлять трафик (next-hop)

Опция задаёт, куда перенаправлять трафик при подтверждённой аномалии.
В качестве следующего хопа можно выбрать:

- систему фильтрации DosGate,
- blackhole для полной утилизации,
- любой next-hop-адрес.



Выбор действия осуществляется по имени, заданному в конфигурационном файле ***analyzer.yaml***.

Пример конфигурации:

```
nexthops:                       # BGP next-hop'ы.
  - name: transit               # Имя next-hop'a
    ip: 192.168.0.1             # IP-адрес next-hop'a

  - name:dosgate                # Имя next-hop'a
    ip: 192.168.10.1            # IP-адрес next-hop'a
```

Количество IP-адресов для сворачивания аномалий

Опция задает, при каком количестве IP-адресов локальные события объединяются в одну глобальную аномалию.

Система отслеживает уникальные IP-адреса, превысившие локальные пороговые значения. Когда их число достигает указанное значение, локальные аномалии сворачиваются и фиксируется одна глобальная.

> ***Важно:***
> Срабатывание возможно только если общий трафик превышает минимальный глобальный порог и в системе настроен хотя бы один глобальный счётчик.

Пример:
Если указано значение 20, то при превышении порога на 19 IP-адресах система зарегистрирует 19 отдельных локальных аномалий.
Когда количество IP достигнет 20 и суммарный трафик превысит минимальный глобальный порог, все локальные события сворачиваются в одну глобальную аномалию.

Начало аномалии

Опция задаёт минимальное время, в течение которого должно сохраняться превышение порога, прежде чем система зафиксирует начало аномалии и выполнит действие. Это позволяет избежать ложных срабатываний от кратковременных всплесков трафика.

Пример:
Если указано значение 10, то превышение порога должно сохраняться в течение 10 секунд, чтобы система система зарегистрировала аномалию и применила действие.

Задержка аномалии

Опция задаёт время, через которое аномалия считается завершённой, если новые совпадения не зафиксированы.

Задержка необходима для корректной обработки pulse-wave атак — периодических всплесков трафика с короткими интервалами. Если действие завершать немедленно при каждом снижении трафика, система может пропустить повторные атаки и вызвать нестабильность

маршрутизации. Установленная задержка позволяет объединять такие всплески в одну аномалию и удерживать трафик на очистке до истечения заданного времени.

Пример:
Если атака повторяется каждые 30 секунд, а задержка задана 60 секунд, система будет считать все всплески одной аномалией. Это исключает постоянное снятие и повторное включение маршрутизации между волнами.

Управление счётчиками

Счётчики — это набор правил для мониторинга трафика по заданным CIDR-адресам. Они позволяют задать пороговые значения в байтах, пакетах или битах.

Каждый счётчик работает как независимый модуль детекции, непрерывно анализирующий сетевой трафик по указанным параметрам. Основная задача — выявление аномалий путём динамического сравнения текущих характеристик трафика с предустановленным порогом. При превышении порога в течение заданного интервала времени счётчик фиксирует аномалию.

Добавить счётчик

> Нажать кнопку **Добавить счётчик**. Счётчик добавится в верхнюю часть списка в виде шаблона и станет доступен для редактирования параметров.

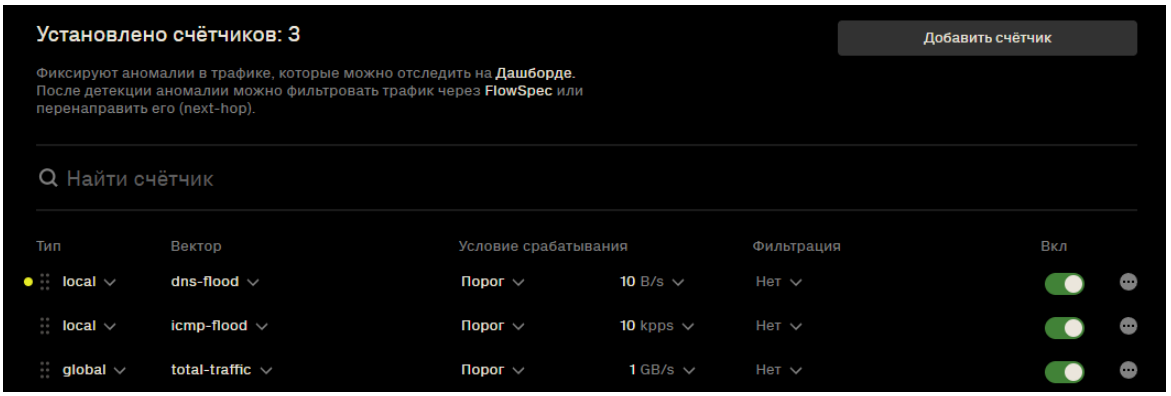
Удалить счётчик

> Нажать кнопку с тремя точками справа от строки счётчика, выбрать **Удалить**.

Дублировать счётчик

> Нажать кнопку с тремя точками справа от строки счётчика, выбрать **Дублировать**. Копия появится в списке и будет доступна для редактирования.

Изменения вступят в силу после нажатия жёлтой кнопки **Применить**.



Переключатель "Вкл"

Переключатель **"Вкл"** управляет активностью счётчика. При отключении счётчик автоматически переносится в конец списка и становится недоступным для редактирования. Фактическая деактивация выполняется **только после** нажатия кнопки **Применить**.

Приоритет счётчиков

Порядок расположения счётчиков определяет их приоритет. Счётчики, находящиеся выше в списке, обрабатываются первыми. При добавлении нового счётчика ему автоматически присваивается наивысший приоритет. При необходимости его можно переместить в списке, задав нужный порядок обработки.

Тип

Типы подсчета трафика. Доступны три режима: *global*, *local* и *subnet*. Тип подсчета определяет, каким образом система интерпретирует превышение лимита в рамках заданного диапазона IP-адресов (CIDR).

- **global**
  Подсчет трафика выполняется суммарно по всем IP-адресам, указанным во вкладке CIDR. Превышение порога регистрируется, если совокупный трафик на все адреса в диапазоне превысил заданное значение. Этот режим подходит для оценки общей нагрузки на подсеть или группу адресов.
- **local**
  Подсчет ведется по каждому отдельному IP-адресу внутри указанного диапазона. Например, для CIDR 192.168.0.0/24 порог будет контролироваться индивидуально для каждого из 256 адресов.

Аномалия фиксируется, если превышение лимита произошло на одном из адресов. Используется для точечной детекции атак на конкретные хосты внутри подсети.

- **subnet**
  Подсчёт ведётся по каждому отдельному префиксу, начиная с длины маски /31. Превышение фиксируется только в том случае, если трафик в пределах конкретной подсети превышает порог. Если нагрузка распределена между несколькими подсетями и в каждой из них порог не превышен, аномалия не регистрируется. Подходит для контроля нагрузки на изолированные адресные диапазоны.

Вектор

Вектор — это фильтр, определяющий, по каким признакам FlowCollector будет учитывать трафик при подсчёте и анализе. Указывается один или несколько векторов, каждый из которых задаёт условие по протоколу, порту, флагу TCP, размеру или структуре пакета. Трафик, не соответствующий заданным векторным условиям, исключается из подсчёта. Векторы необходимы для ограничения области анализа и настройки счётчиков под конкретные типы атак.

Категория	Вектор атаки	Описание
Общие	total-traffic	Любой сетевой пакет (IPv4/IPv6)
	ip-fragment-flood	Пакет с установленным битом фрагментации
	http-flood	TCP, порт получателя 80
	https-flood	TCP, порт получателя 443
	icmp-flood	Протокол ICMP
	dns-flood	UDP или TCP, порт получателя 53
	gre-flood	Протокол GRE
	ip-private-flood	IPv4 с адресом источника из диапазонов: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
	spoofing-flood	Пакеты с подменёнными IP-адресами источника
Flood – TCP/UDP	tcp-flood	Протокол TCP
	udp-flood	Протокол UDP
	tcp-syn-flood	TCP с флагом SYN (0x02)

Категория	Вектор атаки	Описание
	tcp-ack-flood	TCP с флагом ACK (0x10)
	tcp-cwr-flood	TCP с флагом CWR (0x80)
	tcp-ece-flood	TCP с флагом ECE (0x40)
	tcp-fin-flood	TCP с флагом FIN (0x01)
	tcp-null-flood	TCP без установленных флагов
	tcp-psh-flood	TCP с флагом PSH (0x08)
	tcp-rst-flood	TCP с флагом RST (0x04)
	tcp-urg-flood	TCP с флагом URG (0x20)
	udp-zero-payload-flood	UDP-пакет без данных (нулевой payload)
	udp-big-packets-flood	UDP-пакет ≥1498 байт
Amplification	apple-remote-desktop-amp	UDP, порт источника 3283
	chargen-amp	UDP, порт источника 19
	dns-amp	UDP, порт источника 53
	ibm-cics-amp	UDP, порт источника 1435
	ldap-amp	UDP, порт источника 389
	memcached-amp	UDP, порт источника 11211
	mssql-amp	UDP, порт источника 1434
	ntp-amp	UDP, порт источника 123
	snmp-amp	UDP, порт источника 161
	snmptrap-amp	UDP, порт источника 162
	ssdp-amp	UDP, порт источника 1900
	ws-discovery-amp	UDP, порт источника 3702

Условие срабатывания

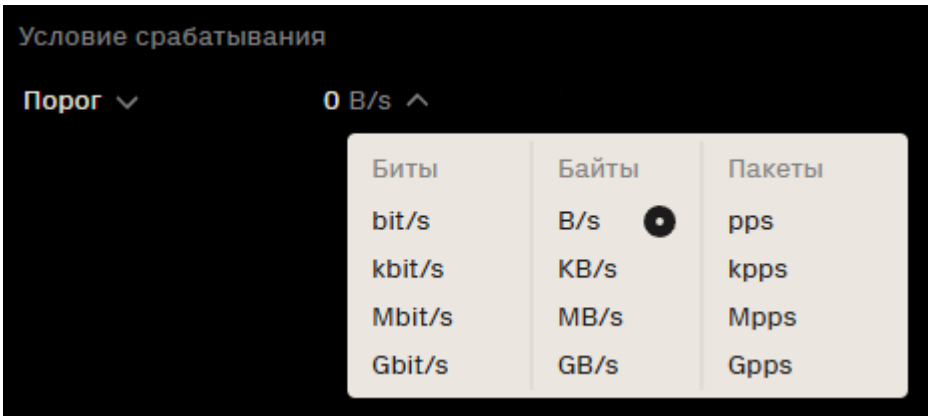
Условия срабатывания определяют критерий, по которому трафик признаётся аномальным.

Поддерживаются три типа:

- **Порог (Threshold)**
  Статический порог. Значение задаётся явно (например, 1 MB/s или 10 Kpps). Срабатывает при превышении заданного значения.

- **Прирост (Diff)**
  Динамический порог по абсолютной разнице. Сравниваются два соседних интервала измерения счётчика (по умолчанию шаг — 1000 мс). Если разница превышает заданный порог (например, рост на 5000 пакетов/с), фиксируется аномалия.
- **Прирост % (Reldiff)**
  Динамический порог по относительной разнице (множителю). Сравниваются два соседних интервала. Если трафик вырос больше, чем на заданный процент (например, на 300%), сработает счётчик.

Далее, указывается единица измерения трафика. Доступны варианты в битах или байтах в секунду, а также в пакетах.



Фильтрация

Для каждого счётчика можно задать способ обработки трафика при срабатывании условия. Ниже представлены три доступных варианта фильтрации.

Без фильтрации

> Аномалия фиксируется и отображается на Дашборде и в отчётах, но трафик остаётся без изменений. Такой режим используется для мониторинга, когда требуется анализировать подозрительную активность без вмешательства в сетевую маршрутизацию или фильтрацию.

Следующий хоп (next-hop)

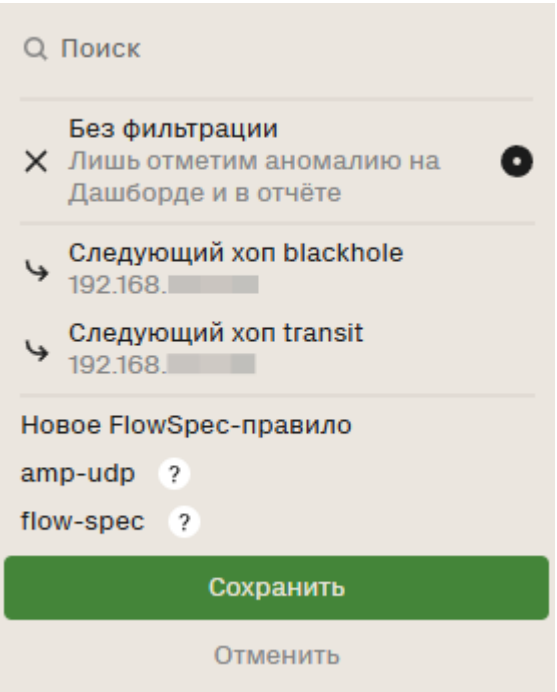
> В отличие от глобальной настройки next-hop, где определяется поведение всей системы при подтверждённой аномалии, здесь

> фильтрация применяется **локально — только к данному счётчику**. Это позволяет задать уникальный маршрут обработки для каждой конкретной аномалии.

FlowSpec-правила

> При срабатывании счётчика трафик обрабатывается по выбранному FlowSpec-правилу. Доступны три варианта:

- **Создать новое правило** — открыть редактор, задать параметры и сохранить. Правило добавится в список и станет доступно для привязки.
- **Выбрать из списка** — применить существующее правило в текущем виде.
- **Дублировать и редактировать** — выбрать правило в списке, нажать **Дублировать и редактировать**, внести правки в открывшемся редакторе и сохранить. Копия появляется в списке и доступна для привязки.

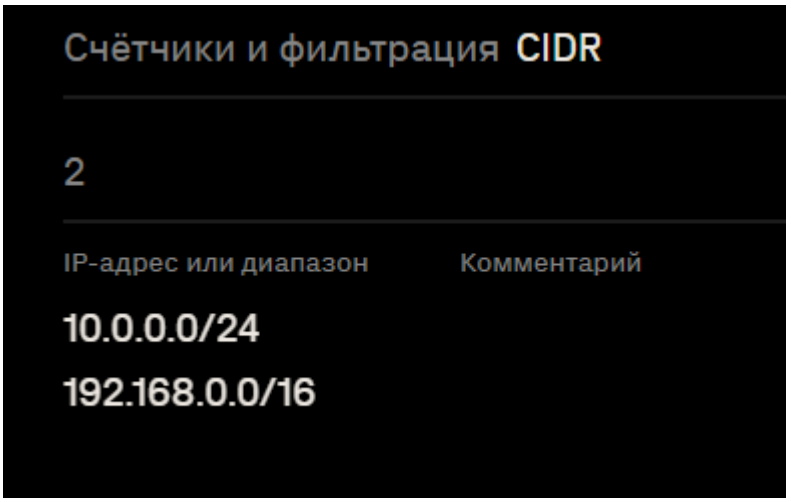


CIDR

Настройка CIDR-адресов объекта

CIDR — это IP-адрес или префикс, для которых FlowCollector выполняет анализ входящего трафика в соответствии с правилами в Объекте.

> **Внимание!**
> Добавление 0.0.0.0/0 не поддерживается.



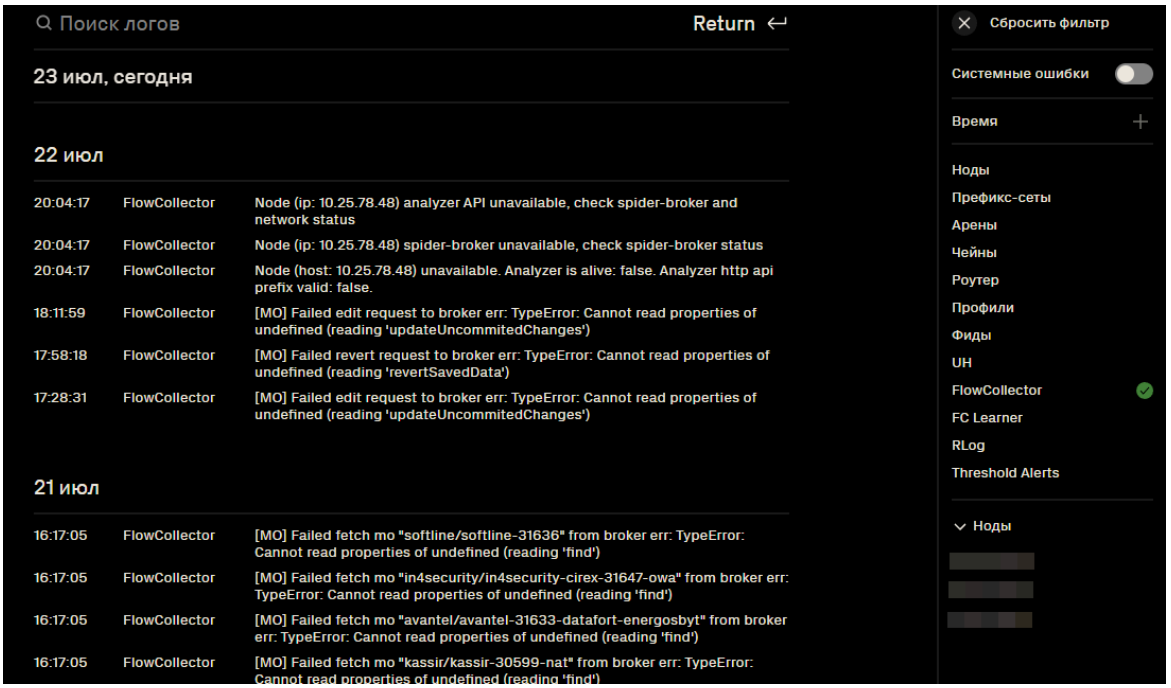
FlowCollector выполняет маршрутизацию пакетов внутри системы, сравнивая ip-адрес назначения в пакете с значением CIDR. На основе этого сравнения принимается решение о том, в какой объект следует направить трафик для дальнейшей обработки.

Панель управления

Логи и дампы

Логи

Логи предназначены для мониторинга событий, возникающих в системе, и позволяет пользователю анализировать ошибки, предупреждения и другие системные сообщения.



Поле ввода в верхней части экрана, позволяющее осуществлять поиск по записям журнала.

Список логов отображает хронологический перечень событий, включающий:

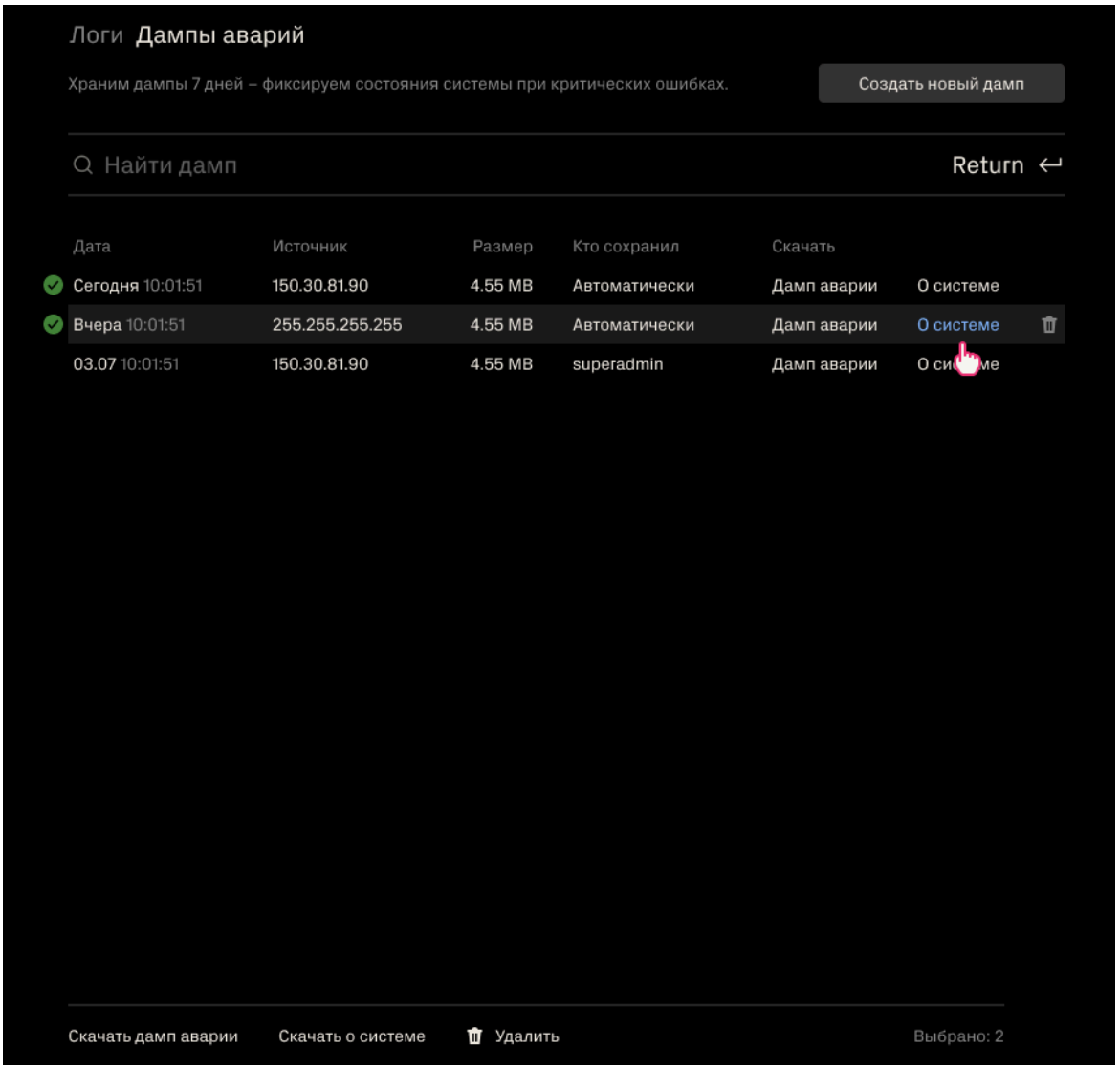
- Время события.
- Источник (Префикс-сеты, Профили, Арены и т.д).
- Описание события

Фильтр логов содержит следующие параметры:

- Системные ошибки – переключатель, позволяющий фильтровать критические ошибки.
- Время – настройка для ограничения выборки логов по дате и времени.
- Категории событий – возможность выбрать источники логов, такие как Ноды, Префикс-сеты, Роутеры, UH и другие. По умолчанию веб-интерфейс отображает логи всех категорий продуктов. Для просмотра сообщений, относящихся к FlowCollector, необходимо категорию **FlowCollector**.
- Ноды – меню для выбора ноды, логи которой отображаются.

Дампы аварий

Дампы аварий — это механизм быстрой диагностики. При сбое любого сервиса платформы система автоматически формирует архив с технической информацией за последние 5 минут.



Поиск дампов позволяет находить нужные записи журнала через поле ввода в верхней части экрана.

В верхней части интерфейса расположена кнопка **Создать новый дамп**. Она запускает ручное создание аварийного дампа. Система собирает данные и формирует архив.

Готовый дамп появляется в таблице. Его можно скачать или удалить.

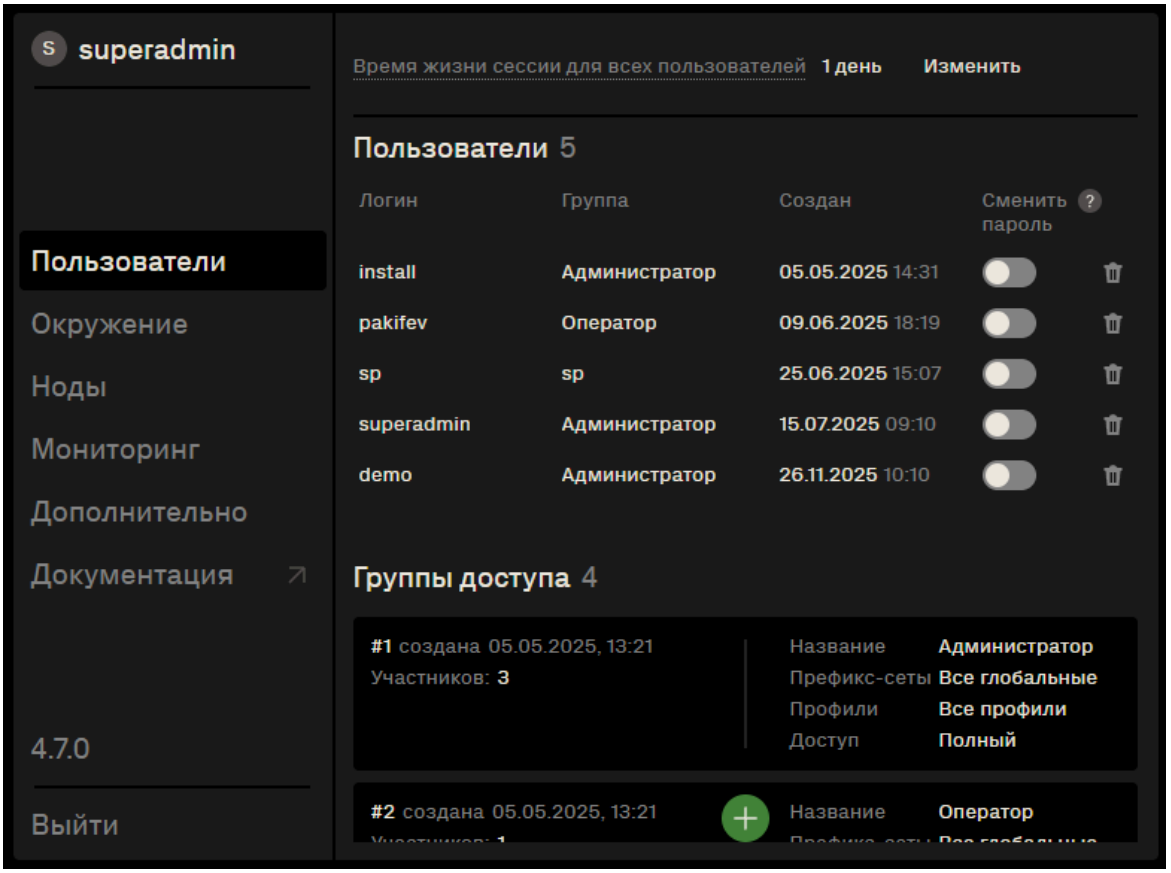
- **Дамп аварии** — полный архив с логами и служебными данными, собранными при сбое.
- **О системе** — краткая сводка о текущем состоянии сервера. Она включает информацию о оборудовании: сетевые интерфейсы, версию ядра, версию операционной системы и другие базовые параметры.

Дампы хранятся 7 дней и удаляются автоматически.

Настройки

Пользователи

Раздел **Пользователи** предназначен для управления пользователями системы и их принадлежностью к группам доступа.



Время жизни сессии для всех пользователей задаёт период, после которого пользователи должны заново войти в систему. При необходимости может быть изменено.

Пользователи - отображает учётные записи и их группы доступа. В списке можно удалить пользователя или включить переключатель **Сменить пароль** — в этом случае система завершит его сессию и попросит установить новый пароль при следующем входе.

Группы доступа - список доступных групп с детализацией параметров.

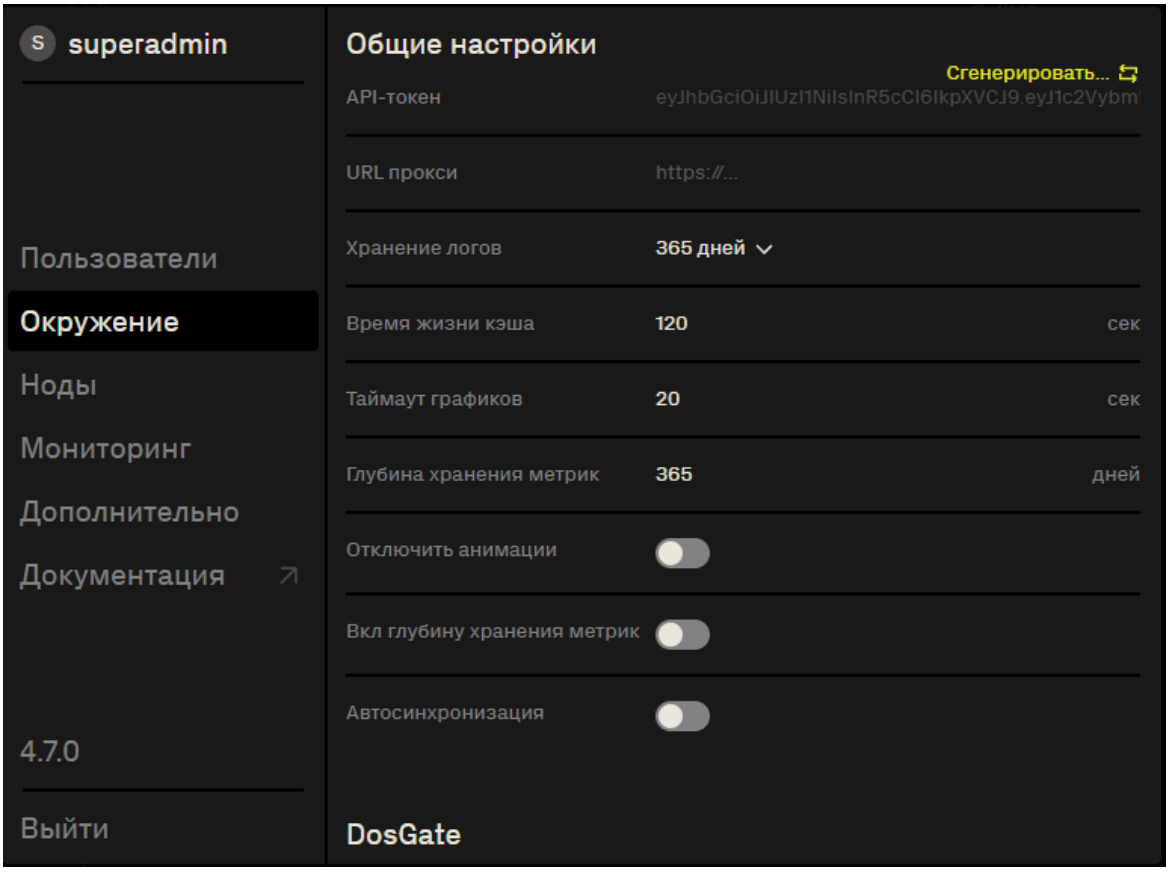
Группы доступа определяют права пользователей на доступ к профилям и префикс-сетам. По умолчанию в системе существуют три роли:

- Администратор
- Оператор
- Пользователь

Каждой группе можно назначить доступ ко всем или только определенным профилям и префикс-сетам. В системе есть возможность создавать собственные группы пользователей с индивидуальными правами доступа. Администратор может настраивать новые группы или изменять права существующих.
В нижней части интерфейса расположена кнопка "+", которая позволяет добавлять как новых пользователей, так и создавать новые группы доступа.

Окружение

Раздел **Окружение** содержит общие настройки системы, включая параметры API, прокси, кэширование, логи и интеграцию с внешними сервисами для работы с метриками и данными.



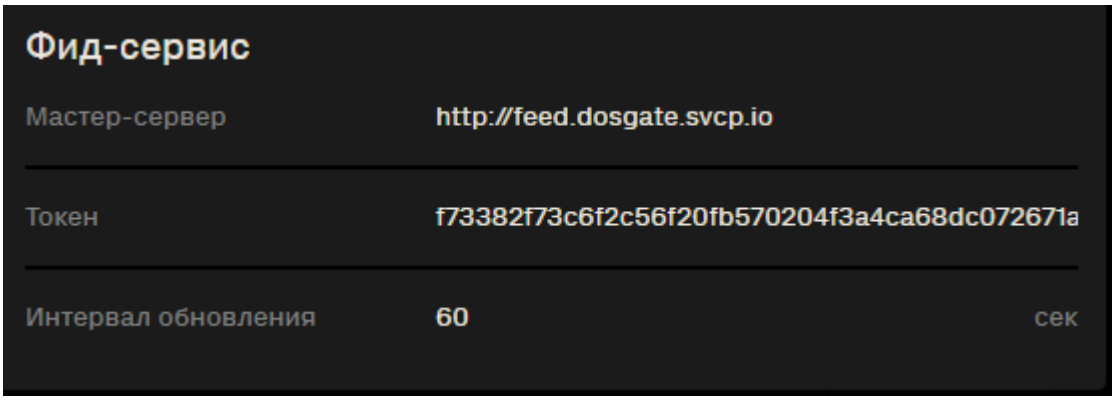
Общие настройки

- **API-токен** – позволяет сгенерировать уникальный ключ доступа к API.
- **URL прокси** – адрес прокси-сервера. Используется для доступа к фид-сервису Servicepipe через прокси-сервер
- **Хранение логов** – задает срок хранения логов в днях (по умолчанию 365 дней).
- **Время жизни кэша** – Хранит данные о профилях и их содержимом для ускорения работы системы. Чем больше значение кэша, тем реже он обновляется, что повышает производительность в крупных инсталляциях.
- **Отключить анимации** – переключатель, отключающий анимационные эффекты интерфейса. Рекомендуется отключать анимацию при подключении к веб-интерфейсу по RDP.
- **Автосинхронизация** – обеспечивает автоматическое приведение конфигурации узлов в соответствие с мастер-сервером при добавлении новых серверов или обнаружении расхождений в политиках.

FlowCollector

- **Graphite URL** – адрес сервера Graphite, с которого загружаются данные для построения графиков в интерфейсе FlowCollector.
- **Обновление графиков** – интервал обновления графиков (в секундах).
- **Обновление аномалий** – интервал опроса системы на наличие новых сетевых аномалий (в секундах).
- **Интервал захвата** – шаг между точками на графиках в отчётах (в секундах).

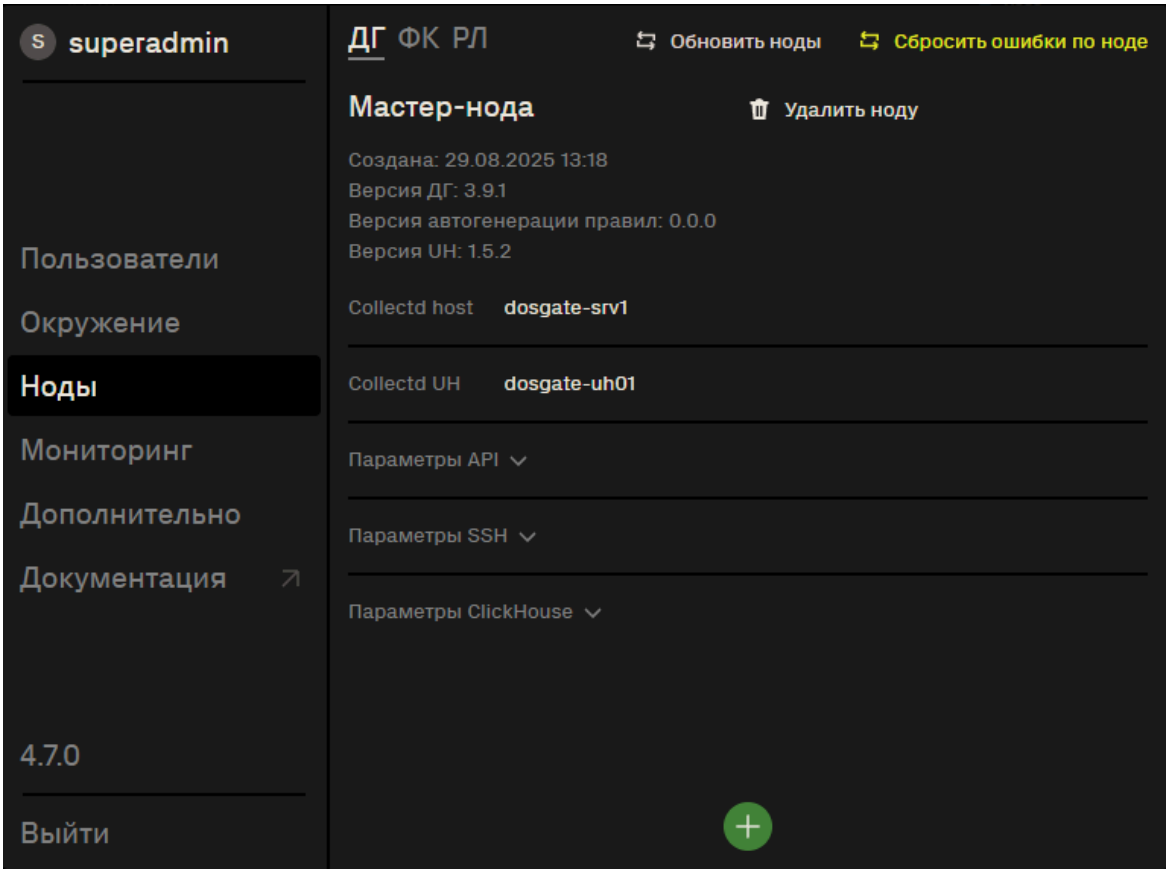
Фид-сервис



- **Мастер-сервер** – URL основного сервера фид-сервиса.
- **Токен** – ключ аутентификации, предоставленный вендором.
- **Интервал обновления**– частота обновления данных фид-сервиса в секундах.

Ноды

Раздел **Ноды** предназначен для управления узлами системы. Здесь отображается список доступных узлов, их статус и параметры.



Основные элементы:

- **Мастер-нода** – главный узел системы, управляющий синхронизацией и политиками других нод. Отображаются дата создания и установленная версия компонента.
- **Metrics host** – имя хоста, с которого производится сбор и обработка статистики.
- **Параметры API** – настройки доступа к API FlowCollector.
- **Параметры SSH** – параметры подключения к узлу по SSH.
- **Параметры MongoDB** – настройки подключения к базе MongoDB.
- **Параметры ClickHouse** – параметры подключения к хранилищу ClickHouse.

Доступные действия:

- **Обновить ноды** – инициирует обновление данных о текущих узлах.
- **Синхронизировать** – выполняет принудительную синхронизацию конфигурации узлов с мастер-нодой.
- **Удалить** – удаляет текущий узел из конфигурации.

Добавление новой ноды

Для добавления нового узла в систему выполните следующие шаги:

1. Открыть раздел **Ноды** в интерфейсе.
2. Нажать кнопку "+" для создания новой ноды.
3. Заполнить параметры узла:

- **Операционная система** – выбрать ОС из списка (Ubuntu 18, Альт 8 СП, Alma Linux).
- **Модуль** – выбрать FlowCollector.
- **Metrics host** – указать имя хоста.
- **MongoDB:**
  - **Хост** – IP-адрес сервера MongoDB.
  - **Порт** – порт подключения к MongoDB (по умолчанию: 27017).
  - **База данных** – имя базы данных.
  - **Пользователь** – имя пользователя MongoDB, под которым осуществляется подключение.
  - **Пароль** – пароль пользователя для аутентификации.
- **ClickHouse:**

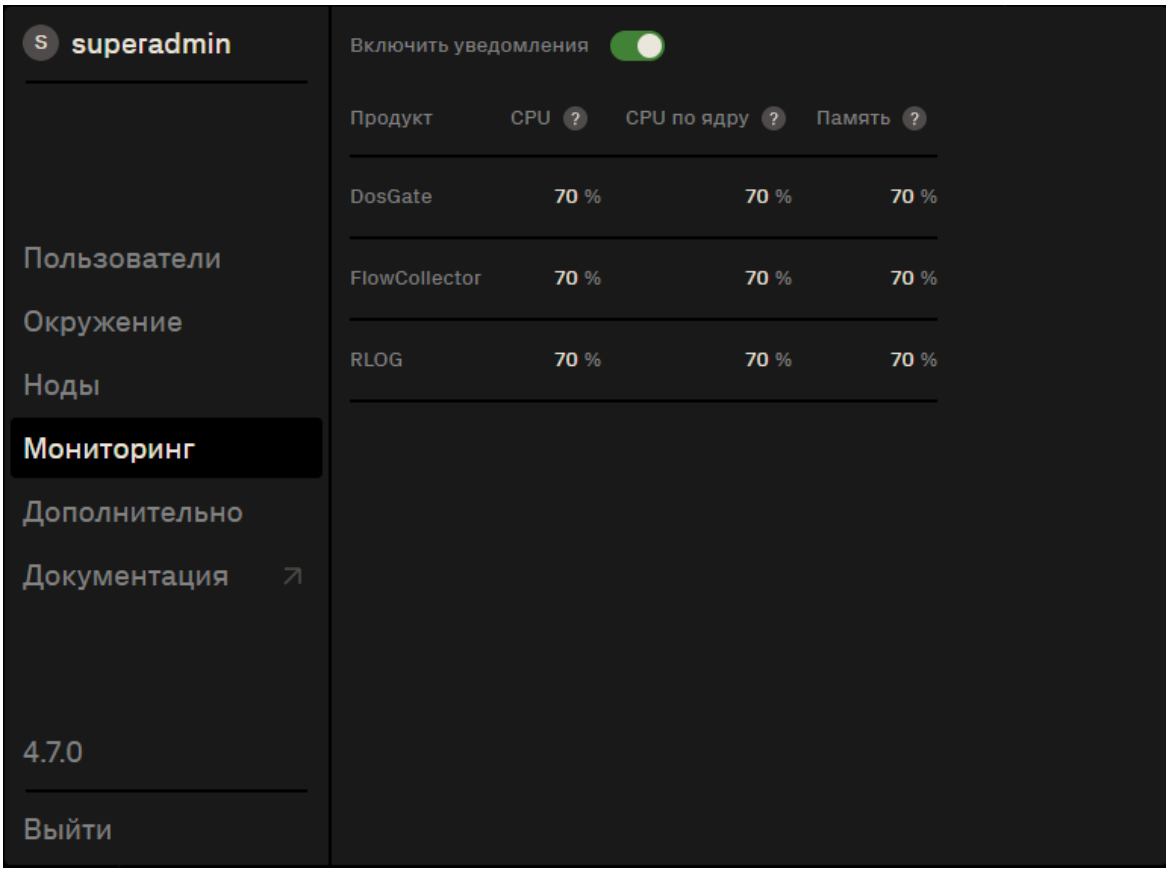
- **Хост** – IP-адрес сервера ClickHouse.
- **Порт** – порт подключения к ClickHouse (по умолчанию: 8123).
- **База данных** – имя базы данных.
- **Пользователь** – имя пользователя ClickHouse, под которым осуществляется подключение.
- **Пароль** – пароль пользователя для аутентификации.

1. Нажать **Подключение**.
   - Доступны два способа соединения – API и SSH, выберите любой из них и настройте соответствующие параметры.
2. Нажать кнопку **Применить**.
3. Завершить добавление узла, нажав **Добавить ноду**.

После этого узел будет добавлен и отображён в общем списке нод.

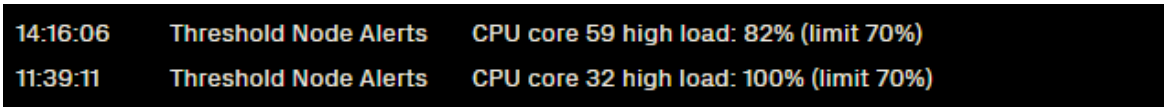
Мониторинг

Мониторинг отображает состояние ресурсов, используемых сервисами платформы. Интерфейс позволяет контролировать загрузку CPU, загрузку CPU по ядрам и объём используемой памяти для каждого продукта.

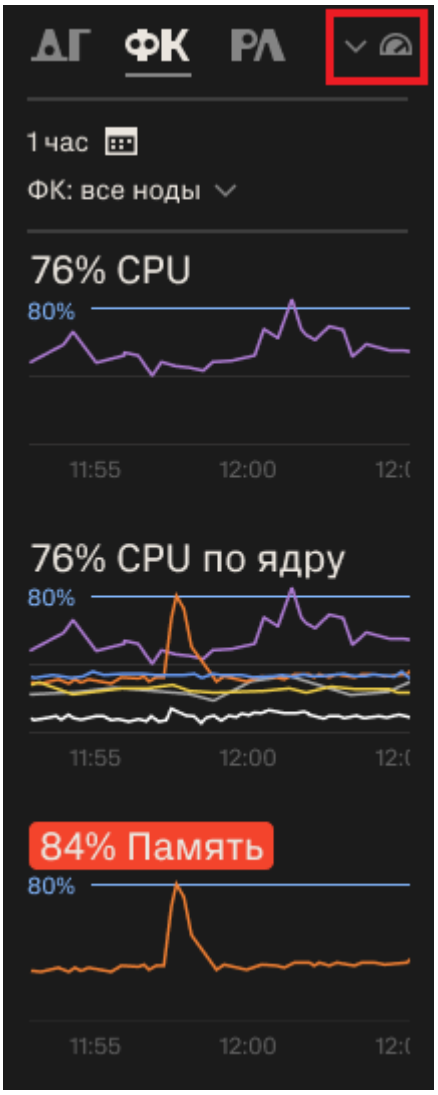


В верхней части расположен переключатель **Включить уведомления**, который активирует отправку предупреждений при превышении заданных порогов нагрузки.

Уведомления мониторинга отображаются в общем списке событий раздела **Логи**.



Для быстрого перехода к данным мониторинга в панели быстрого доступа доступна иконка мониторинга. Показатели, превышающие пороговые значения отображаются красным.

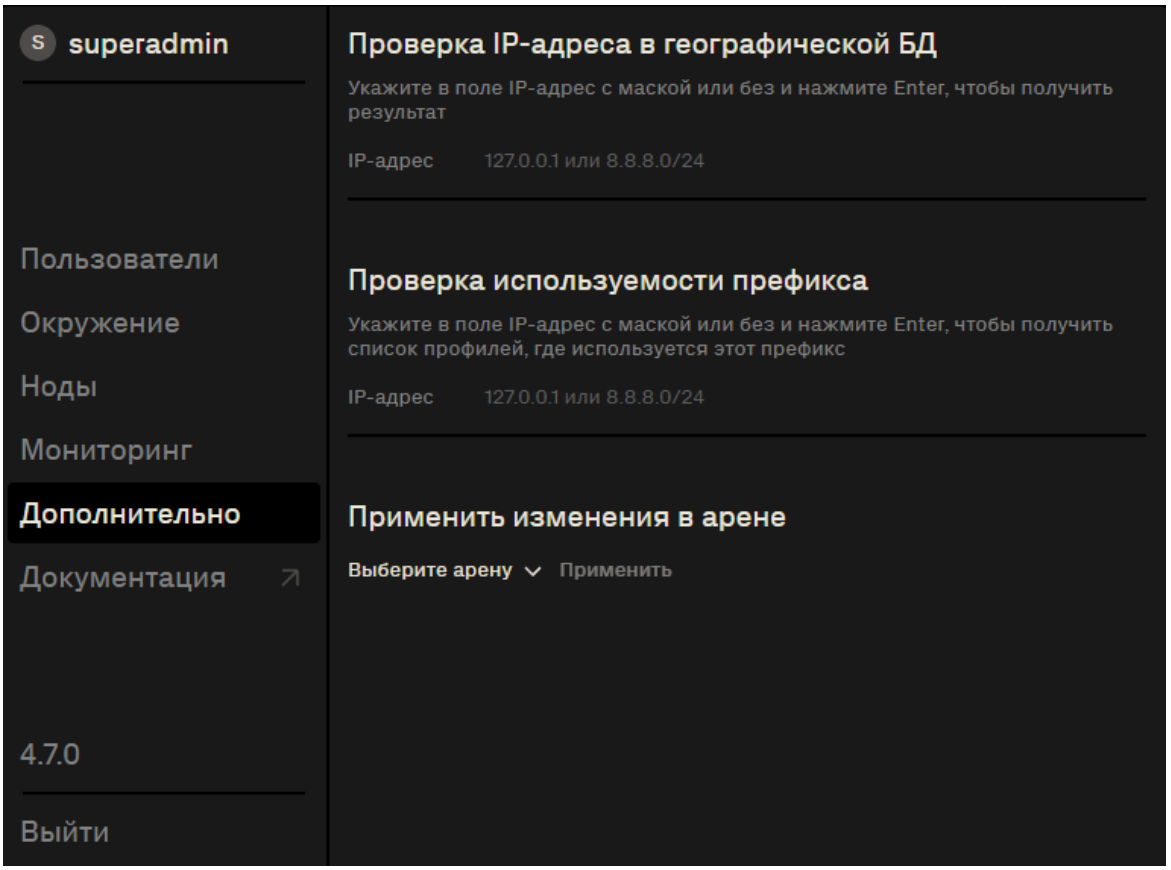


Дополнительно

Проверка IP-адреса в географической БД – позволяет определить географическую принадлежность IP-адреса.

Проверка используемости префикса – отображает список профилей, в которых используется заданный IP-адрес или префикс.

Применение изменений в арене – предоставляет возможность выбора арены из выпадающего списка и применения внесённых изменений с помощью кнопки **Применить**.



Документация

Раздел **Документация** в боковом меню является гиперссылкой, перенаправляющей пользователя к справочным материалам по системе.

Резервное копирование и восстановление

В разделе описаны способы создания и восстановления резервных копий системы.

Резервное копирование скриптом

Скрипт выполняет автоматическое формирование архивного файла, который включает резервные копии баз данных **PostgreSQL** и **MongoDB**, конфигурационные файлы сервисов, а также дополнительные данные: шаблоны и конфигурации **nginx**. Полученный архив может быть использован для восстановления системы при необходимости.

Для запуска необходимо выполнить команду:

```
curl -o "./backup.sh" "https://public-
repo.svcp.io/utility/backup.sh" && \
    sudo chmod +x "./backup.sh" && \
    ./backup.sh
```

Архив резервной копии сохраняется в директории: **/opt/backups/**

Ручное резервное копирование

Подготовка

Создать директорию для резервных копий:

```
sudo mkdir -p /opt/backups
```

Создать временную директорию:

```
sudo mkdir -p /opt/backups/node0
```

Резервное копирование базы данных PostgreSQL

Проверить наличие файла конфигурации:

```
sudo ls -la /opt/sp-spider-broker/.env
```

Открыть файл и посмотреть значения переменных подключения к базе (**DB_HOST**, **DB_PORT**, **DB_USER**, **DB_PASSWORD**, **DB_NAME**). Эти параметры понадобятся для бэкапа:

```
sudo grep -E '^DB_' /opt/sp-spider-broker/.env
```

Создать директорию для бэкапа PostgreSQL:

```
sudo mkdir -p /opt/backups/node0/postgres
```

Выполнить резервное копирование PostgreSQL с использованием считанных параметров:

```
sudo PGPASSWORD="DB_PASSWORD" pg_dump \
    -h "DB_HOST" \
    -p "DB_PORT" \
    -U "DB_USER" \
    "DB_DATABASE" \
    > /opt/backups/node0/postgres/DB_DATABASE.sql
```

Резервное копирование базы данных MongoDB

Проверить наличие утилиты **mongodump**:

```
sudo which mongodump
```

Если утилита не найдена — установить:

```
sudo apt-get update
sudo apt-get install -y mongodb-clients
```

Получить параметры подключения к MongoDB из PostgreSQL:

```
sudo PGPASSWORD="DB_PASSWORD" psql \
  -h "DB_HOST" \
  -p "DB_PORT" \
  -U "DB_USER" \
  -d "DB_DATABASE" \
  -t -A \
  -c 'SELECT "mongoConnection"::text FROM public.node LIMIT 1;'
```

Создать директорию для резервной копии MongoDB:

```
sudo mkdir -p /opt/backups/node0/mongo
```

Выполнить резервное копирование MongoDB (подставьте свои значения из mongoConnection):

```
sudo mongodump \
    --host "DB_HOST" \
    --port "DB_PORT" \
    --username "DB_USER" \
    --password "DB_PASSWORD" \
    --db "DB_DATABASE" \
    --out /opt/backups/node0/mongo
```

Резервное копирование конфигурационных файлов

Анализатор трафика (FlowCollector)

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/spfc
```

Скопировать конфигурационный файл:

```
sudo cp /opt/spfc/etc/analyzer.yaml /opt/backups/node0/spfc/
```

Скопировать каталог *etc*:

```
sudo cp -r /opt/spfc/etc /opt/backups/node0/spfc/
```

Веб-интерфейс (SP-Spider)

> **Внимание!**
> Компонент может быть установлен на отдельной ноде. Операции выполняются на той ноде, где установлен веб-интерфейс.

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/sp-spider
```

Скопировать конфигурационный файл:

```
sudo cp /opt/sp-spider/.env /opt/backups/node0/sp-spider/
```

Брокер сообщений (SP-Spider-Broker)

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/sp-spider-broker
```

Скопировать конфигурационный файл:

```
sudo cp /opt/sp-spider-broker/.env /opt/backups/node0/sp-spider-
broker/
```

Сервис событий (SP-Events)

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/sp-events
```

Скопировать конфигурационный файл:

```
sudo cp /opt/sp-events/.env /opt/backups/node0/sp-events/
```

Скопировать каталог *template*:

```
sudo cp -r /opt/sp-events/template /opt/backups/node0/sp-events/
```

Агент сбора метрик (Carbon-ClickHouse)

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/carbon-clickhouse
```

Скопировать конфигурационный файл:

```
sudo cp /etc/carbon-clickhouse/carbon-clickhouse.conf
/opt/backups/node0/carbon-clickhouse/
```

Хранилище метрик (Graphite-ClickHouse)

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/graphite-clickhouse
```

Скопировать конфигурационный файл:

```
sudo cp /etc/graphite-clickhouse/graphite-clickhouse.conf
/opt/backups/node0/graphite-clickhouse/
```

Nginx

Создать директорию для резервной копии:

```
sudo mkdir -p /opt/backups/node0/nginx
```

Скопировать конфигурационный файл:

```
sudo cp /etc/nginx/nginx.conf /opt/backups/node0/nginx/
```

Скопировать каталоги *sites-enabled* и *sites-available*:

```
sudo cp -r /etc/nginx/sites-* /opt/backups/node0/nginx/
```

Создание архива

Создать архив:

```
sudo tar -czf /opt/backups/node0.tar.gz -C /opt/backups node0
```

Удалить временную директорию:

```
sudo rm -rf /opt/backups/node0
```

Проверить результат:

```
ls -la /opt/backups/node0.tar.gz
```

Проверить содержимое архива (отобразить первые 20 файлов):

```
sudo tar -tzf /opt/backups/node0.tar.gz | head -20
```

В результате выполнения шагов сформирован архив **/opt/backups/node0.tar.gz**, содержащий резервные копии PostgreSQL и MongoDB, конфигурационные файлы сервисов и дополнительные данные (шаблоны, лицензии, конфигурации nginx)

Восстановление системы

Перейти в каталог **/opt/backups/**, найти архив с резервной копией и распаковать:

```
sudo tar -xzvf node0.tar.gz
```

Восстановление базы данных PostgreSQL

Перейти в каталог базы:

```
cd node0/postgres
```

Найти файл с расширением **.sql**.

При необходимости создать базу и пользователя:

```
sudo -u postgres psql
```

- В консоли PostgreSQL выполнить:

```
CREATE DATABASE DB_DATABASE;
CREATE USER DB_USER WITH ENCRYPTED PASSWORD 'DB_PASSWORD';
GRANT ALL PRIVILEGES ON DATABASE DB_DATABASE TO DB_USER;
\q
```

Выполнить восстановление базы:

```
sudo PGPASSWORD="DB_PASSWORD" psql \
  -h "DB_HOST" \
  -p "DB_PORT" \
  -U "DB_USER" \
  -d "DB_DATABASE" \
  -f DB_DATABASE.sql
```

При необходимости обновить параметры подключения в конфигурационных файлах (.env) сервисов *sp-spider*, *sp-spider-broker*, *sp-events*.

Перезапустить сервисы:

```
sudo systemctl restart sp-spider sp-spider-broker sp-events
```

Восстановление базы данных MongoDB

Перейти в каталог с резервной копией MongoDB:

```
cd node0/mongo
```

При необходимости создать пользователя MongoDB:

- Отключить авторизацию в конфигурации:

```
sudo nano /etc/mongod.conf
```

- Закомментировать строки:

```
security:
  authorization: enabled
```

- Перезапустить MongoDB:

```
sudo systemctl restart mongod
```

- Создать пользователя:

```
sudo mongosh

use DB_DATABASE;
db.createUser({
  user: "DB_USER",
  pwd: "DB_PASSWORD",
  roles: [{ role: "readWrite", db: "DB_DATABASE" }]
});
exit
```

- Включить авторизацию обратно и перезапустить сервис:

```
sudo nano /etc/mongod.conf
```

- Раскомментировать строки:

```
security:
  authorization: enabled
```

```
sudo systemctl restart mongod
```

Выполнить восстановление базы:

```
sudo mongorestore \
  --host "DB_HOST" \
  --port "DB_PORT" \
  --username "DB_USER" \
  --password "DB_PASSWORD" \
  --db "DB_DATABASE" \
  ./DB_DATABASE
```

При необходимости обновить параметры подключения в файле **.env** сервиса *sp-events* и в настройках ноды в веб-интерфейсе SP-Spider.

Перезапустить сервис:

```
sudo systemctl restart sp-events
```

Восстановление конфигурационных файлов

Анализатор трафика (FlowCollector)

Перейти в каталог с резервной копией:

```
cd node0/spfc/
```

Скопировать каталог *etc*:

```
sudo cp -r ./etc /opt/spfc/
```

Перезапустить сервис:

```
sudo systemctl restart analyzer
```

Веб-интерфейс (SP-Spider)

> **Внимание!**
> Компонент может быть установлен на отдельной ноде. Операции выполняются на той ноде, где установлен веб-интерфейс.

Перейти в каталог с резервной копией:

```
cd node0/sp-spider
```

Скопировать конфигурационный файл:

```
sudo cp ./.env /opt/sp-spider/.env
```

Перезапустить сервис:

```
sudo systemctl restart sp-spider
```

Брокер сообщений (SP-Spider-Broker)

Перейти в каталог с резервной копией:

```
cd node0/sp-spider-broker
```

Скопировать конфигурационный файл:

```
sudo cp ./.env /opt/sp-spider-broker/.env
```

Перезапустить сервис:

```
sudo systemctl restart sp-spider-broker
```

Сервис событий (SP-Events)

Перейти в каталог с резервной копией:

```
cd node0/sp-events
```

Скопировать конфигурационный файл:

```
sudo cp ./.env /opt/sp-events/.env
```

Скопировать каталог *template*:

```
sudo cp -r ./template /opt/sp-events/template
```

Перезапустить сервис:

```
sudo systemctl restart sp-events
```

Агент сбора метрик (Carbon-ClickHouse)

Перейти в каталог с резервной копией:

```
cd node0/carbon-clickhouse
```

Скопировать конфигурационный файл:

```
sudo cp ./carbon-clickhouse.conf /etc/carbon-clickhouse/carbon-
clickhouse.conf
```

Перезапустить сервис:

```
sudo systemctl restart carbon-clickhouse
```

Хранилище метрик (Graphite-ClickHouse)

Перейти в каталог с резервной копией:

```
cd node0/graphite-clickhouse
```

Скопировать конфигурационный файл:

```
sudo cp ./graphite-clickhouse.conf /etc/graphite-
clickhouse/graphite-clickhouse.conf
```

Перезапустить сервис:

```
sudo systemctl restart graphite-clickhouse
```

Nginx

Перейти в каталог с резервной копией:

```
cd node0/nginx
```

Скопировать конфигурационный файл:

```
sudo cp ./nginx.conf /etc/nginx/nginx.conf
```

Скопировать каталоги *sites-enabled* и *sites-available*:

```
sudo cp -r ./sites-* /etc/nginx/
```

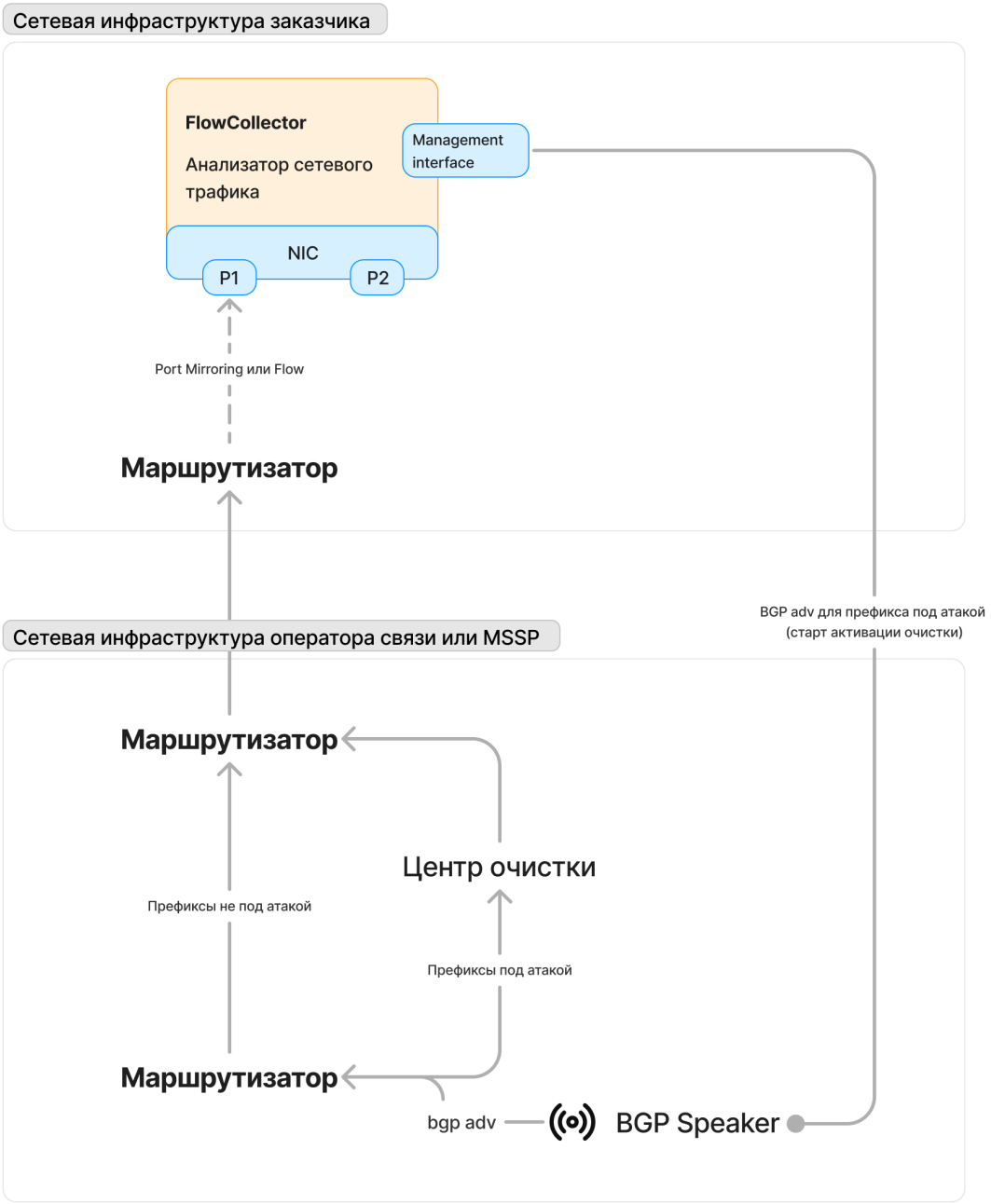
Перезапустить сервис:

```
sudo systemctl restart nginx
```

Облачная сигнализация (BGP)

Установленный на периметре сетевой инфраструктуры DosGate способен подавить вредоносный трафик до общей пропускной способности входящих каналов связи и производительности локальной инсталляции. В случае переполнения входящих каналов связи или локального центра очистки, FlowCollector может в автоматическом режиме подать сигнал о подключении фильтрации на вышестоящих операторов связи или MSSP (поставщиков услуг защиты).

FlowCollector реализует механизм облачной сигнализации за счет интеграции с GoBGP.



Настройка сигнализации по BGP

FlowCollector на основании настроенных объектов защиты автоматически определяет начало и конец сетевой аномалии, а также префиксы находящиеся под атакой, которые он направит на вышестоящих операторов связи или MSSP.

Установите GoBGP

```
sudo apt install gobgpd=3.19.0
```

Активируйте GoBGP на FlowCollector в файле analyzer.yaml

При детекции сетевой аномалии (DDoS-атаки) - будут активироваться политики указанные в GoBGP

```
gobgp:
  enable: true
  # Хост GoBGP API
  host: localhost
  # Порт GoBGP API
  port: 50051
```

Конфигурация политик маршрутизации

```
sudo nano /etc/gobgpd.conf
```

Замените в примере конфигурации следующие данные:

- `192.0.100.103` - IP-адрес FlowCollector
- `192.0.100.101` - IP первого BGP соседа
- `192.0.100.102` - IP второго BGP соседа
- `65003` - AS FlowCollector
- `65001` и `65002` - AS BGP соседей
- `192.0.1.101` - nexthop для первого BGP соседа
- `192.0.1.102` - nexthop для второго BGP соседа

```
[global.config]
  as = 65003
  router-id = "192.0.100.103"
  port = 179

[global.apply-policy.config]
export-policy-list = ["first-export-policy", "second-export-
policy"]

[[neighbors]]
```

```
  [neighbors.config]
    neighbor-address = "192.0.100.101"
    peer-as = 65001
    [neighbors.ebgp-multihop.config]
      enabled = true

    [[neighbors.afi-safis]]
    [neighbors.afi-safis.config]
      afi-safi-name = "ipv4-unicast"

    [neighbors.transport.config]
      local-address = "192.0.100.103"

    [neighbors.apply-policy.config]
      default-import-policy = "reject-route"
      default-export-policy = "reject-route"

[[neighbors]]
  [neighbors.config]
    neighbor-address = "192.0.100.102"
    peer-as = 65002
    [neighbors.ebgp-multihop.config]
      enabled = true

    [[neighbors.afi-safis]]
    [neighbors.afi-safis.config]
      afi-safi-name = "ipv4-unicast"

    [neighbors.transport.config]
      local-address = "192.0.100.103"

    [neighbors.apply-policy.config]
      default-import-policy = "reject-route"
      default-export-policy = "reject-route"

[[defined-sets.neighbor-sets]]
  neighbor-set-name = "first-neighbor"
  neighbor-info-list = ["192.0.100.101"]

[[defined-sets.neighbor-sets]]
  neighbor-set-name = "second-neighbor"
  neighbor-info-list = ["192.0.100.102"]

[[defined-sets.prefix-sets]]
  prefix-set-name = "allowed-prefixes"
  [[defined-sets.prefix-sets.prefix-list]]
    ip-prefix = "0.0.0.0/0"
    masklength-range = "25..32"

[[policy-definitions]]
  name = "first-export-policy"
```

```
  [[policy-definitions.statements]]
    name = "first-statement"
    [policy-definitions.statements.conditions.match-prefix-set]
      prefix-set = "allowed-prefixes"
    [policy-definitions.statements.conditions.match-neighbor-set]
      neighbor-set = "first-neighbor"
    [policy-definitions.statements.actions]
      route-disposition = "accept-route"
    [policy-definitions.statements.actions.bgp-actions]
      set-next-hop = "192.0.1.101"

[[policy-definitions]]
  name = "second-export-policy"
  [[policy-definitions.statements]]
    name = "second-statement"
    [policy-definitions.statements.conditions.match-prefix-set]
      prefix-set = "allowed-prefixes"
    [policy-definitions.statements.conditions.match-neighbor-set]
      neighbor-set = "second-neighbor"
    [policy-definitions.statements.actions]
      route-disposition = "accept-route"
    [policy-definitions.statements.actions.bgp-actions]
      set-next-hop = "192.0.1.102"
```

Перезапустите GoBGP

Перезапустите сервис, убедитесь что запустился демон

```
sudo systemctl enable --now gobgpd
sudo service gobgpd restart
sudo service gobgpd status
```

Перезапустите FlowCollector

Перезапуск сервиса нужен для применения изменений к analyzer.yaml

```
sudo service analyzer restart
sudo service analyzer status
```

Проверьте активные BGP-сессии

Команда покажет BGP-соседей и статус каждой сессии. Если всё настроено - все сессии должны быть успешно установлены ("Established")

```
sudo gobgp nei
```