ОБНОВЛЕНО 16 МАР

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 • 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

# 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
  # 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'а 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'а
  # 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
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'а может
использоваться как сокращение в опции action.
    - name: transit                   # Имя next-hop'а
      ip: 192.168.10.1                # IP-адрес next-hop'а
    - name: blackhole
      ip: 192.168.20.1                # IP-адрес next-hop'а
      # 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
```