Межсетевой экран «SGroups»

Руководство по установке, настройке и администрированию

Содержание

Информация

Введение

Что такое SGroups

SGroups — это Host Based NGFW (Межсетевой экран нового поления) с использованием технологии nftables. Данный продукт распространяется по лицензии MIT.

Он был создан для:

1. упрощения процесса настройки/поддержки правил сетевого трафика;
2. увеличения надежности передаваемых данных в условиях нулевого доверия.

Это делает его идеальным решением для использования в крупных корпоративных сетях, а также в малых и средних предприятиях.

Какие проблемы решает

Проект разрабатывается с учетом потребностей крупных финансовых компаний, которые столкнулись с рядом проблем в области сетевой изоляции. Среди этих проблем можно выделить:

1) неоднородность конфигурации межсетевого оборудования от разных производителей;
2) сложность и высокая стоимость точечной изоляции (ip to ip, team to team);
3) отсутствие единого декларативного подхода к конфигурации сетевых правил;
4) высокая стоимость оборудования.

Пользователи

Приложение не имеет конечного пользователя, так как его работа связана с защитой сетевой инфраструктуры без непосредственного взаимодействия с пользователями. Основная роль в приложении отводится администратору, который производит установку, настройку и мониторинг работы приложения, а также при необходимости вносит коррективы в параметры его работы. Данное руководству поможет в полной мере освоить все необходимые приёмы работы с приложением SGroups.

Преимущества

Использование SGroups, работающего на уровне операционной системы Linux, обладает несколькими преимуществами по сравнению с железными брандмауэра от популярных производителей сетевого оборудования:

1. Удобство управления.
   Управление и настройка SGroups осуществляется с помощью инструментов управления операционной системы, что обеспечивает простоту и удобство в работе, а также сокращает затраты на обслуживание. Это значительно упрощает задачу администрирования и управления системой.
2. Низкая стоимость.
   SGroups предоставляет возможность использовать уже имеющееся оборудование, что позволяет снизить затраты на приобретение дополнительного оборудования. Это особенно полезно для небольших организаций или отделов, у которых ограниченный бюджет на IT-инфраструктуру.
3. Гибкость.
   SGroups обеспечивает возможность настройки политик безопасности на уровне отдельных приложений и сервисов, что позволяет более гибко контролировать защиту.
4. Масштабируемость.

SGroups обладает возможностью легкого масштабирования на большом количестве серверов, без необходимости приобретения дополнительного оборудования и проведения сложной интеграции.

5. Улучшенная защита.
   SGroups обладает высоким уровнем безопасности, так как он работает на уровне операционной системы каждого устройства в сети, что позволяет обеспечить защиту на более глубоком уровне.

Терминология

Security group*(SG)* — это логическая группа для виртуального брандмауэра, которая включает набор инстансов или подсетей для фильтрации ingress/Egress правилами для сетевого трафика. Security group работает на уровне инстансов, контролируя трафик на основе правил, определенных в нем. Каждое правило содержит исходные и целевые IP-адреса/диапазоны, протокол и порт. Если трафик соответствует любому из определенных правил, он будет разрешен или запрещен в зависимости от настроек Security group.
FQDN*(Fully Qualified Domain Name)* — имя домена, не имеющее неоднозначностей в определении. Включается в себя имена всех родительских доменов иерархии DNS.

ICMP*(Internet Control Message Protocol)* — сетевой протокол, входящий в стек протоколов TCP/IP. В основном используется для передачи данных сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашивая услуга недоступна или хост, или маршрутизатор не отвечают. Так же на ICMP возлагают некоторые сервисные функции.

Ingress — трафик исходящий из инстанса в HBF правил описывающее разрешающее правило для входящего трафика инстанса.

Egress — трафик исходящий из инстанса в HBF правил описывающее разрешающее правило для исходящего трафика инстанса.

CIDR*(Classless Inter-Domain Routing)* — это подсеть которая включает в себя диапазон IP адресов.

Namespace*(пространство имен)* — пространство имен.

Netspace*(сеть)* — сетевое пространство имен.

Subnet*(подсеть)* — часть сети с присвоенным адресом CIDR

BM*(деплоймент/compute instance)* — виртуальная машина.

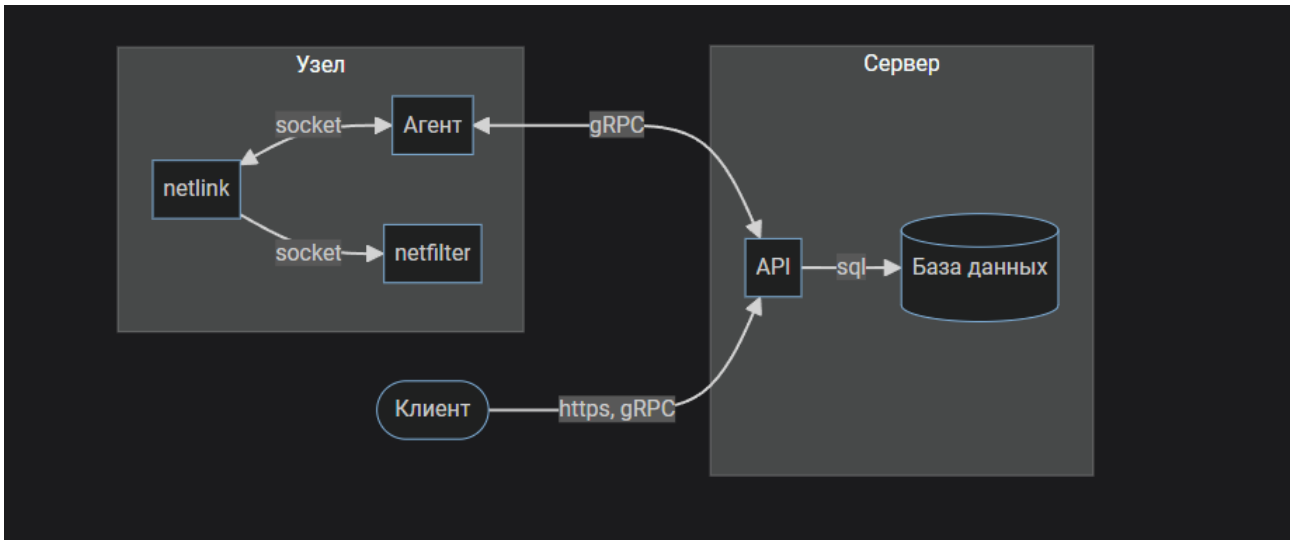
Array*(массив)* — упорядоченный набор элементов, каждый из которых хранит одно значение.

Object*(объект)* — неупорядоченный набор пар ключ/значение. Объект начинается с открывающей фигурной скобки и заканчивается закрывающей фигурной скобкой. Каждое имя сопровождается двоеточием, пары ключ/значение разделяются запятой.

Hook — система отслеживания соединений, NAT-движок, инфраструктура логирования и пользовательские очереди остаются без изменений. Новым является только фреймворк классификации пакетов.

Техническая документация

Компоненты



Сервер

Сервером является приложение состоящие из API-сервиса и базы данных.

API: данный сервис разработан на языке программирования Go и предоставляет реализацию подходов GRPC и REST API с использованием protobuf схем. Он реализует интерфейс взаимодействия 'агентов' с данными, хранящимися в базе данных.

База данных: В качестве базы данных выступает PostgreSQL. Используется для хранения матрицы сетевого взаимодействия.

Узел

Агент: программное обеспечение осуществляющее конфигурацию netfilter'a в соответствии с правилами, указанные в API. Взаимодействие агента с API происходит через протокол gRPC, а получение конфигурации осуществляется посредством push/pull-запросов.

netlink: механизм, который позволяет различным компонентам операционной системы обмениваться информацией. Это как система передачи сообщений между разными частями операционной системы Linux, чтобы они могли работать вместе.

netfilter: часть ядра Linux, которая отвечает за фильтрацию и манипуляцию сетевым трафиком в операционной системе.

Клиент

Клиент — это инструмент, который использует для взаимодействия интерфейсы API.

Требования

"SGroups совместим с `x86_64`, `amd64`, `armhf`, `arm64`, `s390x` архитектурами."

Для установки потребуется:

linux kernel >= 3.10.0

nft --version >= v0.9.3 (Topsy)

go version >= 1.19

postgresql >= 14.8

HBF-агент

Установка

Вне зависимости от выбраного вида установки hbf-server, необходимо настроить общие переменные окружения: HBF_SERVER - IP-адрес hbf-server.

Setup HBF_SERVER

export HBF_SERVER=0.0.0.0

HBF_SERVER_PORT – порт, используемый hbf-server.

Setup HBF_SERVER_PORT

export HBF_SERVER_PORT=9650

DNS_SERVER – Список доверенных IP-адресов DNS серверов.

Setup DNS_SERVER

export DNS_SERVER=['8.8.8.8']

VERSION – версия агента, которую пользователь хочет установить.

Setup VERSION

export VERSION=1.9.1

docker

Перед развертыванием убедитесь, что у вас установлен **docker**:

Get docker version

docker -v

Клонируем репозиторий, переходим в директорию репозитория и переключаемся на тег необходимой версии командой:

Git setup

git clone https://github.com/H-BF/sgroups

cd sgroups

git fetch *--tags --all*

git checkout tags/*v${VERSION}*

Создайте конфигурационный файл для hbf-agent командой:

Configuration file setup

```
cat <<EOF > config-agent.yaml
---
graceful-shutdown: 10s

logger:
  level: DEBUG

fqdn-rules:
  strategy: dns

extapi:
  svc:
    def-daial-duration: 10s
    sgroups:
      dial-duration: 3s
      address: 'tcp://${HBF_SERVER}:${HBF_SERVER_PORT}'
      sync-status:
        interval: 3s #mandatory
        push: true

netlink:
  watcher: #netlink watcher
    linger: 1s

base-rules:
  networks: ['${HBF_SERVER}/32']

dns:
  nameservers: ${DNS_SERVER}
  proto: tcp
  port: 53
  dial-duration: 3s
  read-duration: 5s #default
  write-duration: 5s #default 5s
  retries: 5 #default 1
  retry-timeout: 3s #default 1s

telemetry:
  useragent: "string"
  endpoint: 0.0.0.0:9660
  metrics:
    enable: true
  healthcheck:
```

```
    enable: true
EOF
```

Собираем образ hbf-agent командой:

docker build *-f* docker/Dockerfile.agent *--tag* to-nft:v*${VERSION}* .

Запускаем hbf-agent командой:

```
docker run \
-d \
-v ./config-agent.yaml:/opt/swarm/etc/to-nft/config-agent.yaml \
--name hbf-agent \
--entrypoint "/app/bin/to-nft" \
--privileged \
--user=0 \
to-nft:v${VERSION} -config /opt/swarm/etc/to-nft/config-agent.yaml
```

deb

Настроим необходимые переменные окружения командой:

Environment setup

***export PACKAGE_TYPE*=deb**
***export URL*=https://github.com/H-BF/sgroups/releases/download**
export RELEASE*=v*$VERSION*/to-nft-*$VERSION*-any.*$PACKAGE_TYPE

Скачиваем и устанавливаем deb-пакет командой:

Install deb package

```
sudo wget -O /tmp/to-nft.deb $URL/$RELEASE
sudo dpkg -i /tmp/to-nft.deb
```

Создайте конфигурационный файл для hbf-agent командой:

Configuration file setup

```
mkdir -p /opt/swarm/etc/to-nft
cat <<EOF > /opt/swarm/etc/to-nft/config-agent.yaml
---
graceful-shutdown: 10s

logger:
  level: DEBUG
```

```
fqdn-rules:
  strategy: dns

extapi:
  svc:
    def-daial-duration: 10s
    sgroups:
      dial-duration: 3s
      address: 'tcp://${HBF_SERVER}:${HBF_SERVER_PORT}'
      sync-status:
        interval: 3s #mandatory
        push: true

netlink:
  watcher: #netlink watcher
    linger: 1s

base-rules:
  networks: ['${HBF_SERVER}/32']

dns:
  nameservers: ${DNS_SERVER}
  proto: tcp
  port: 53
  dial-duration: 3s
  read-duration: 5s #default
  write-duration: 5s #default 5s
  retries: 5 #default 1
  retry-timeout: 3s #default 1s

telemetry:
  useragent: "string"
  endpoint: 0.0.0.0:9660
  metrics:
    enable: true
  healthcheck:
    enable: true
EOF
```

Запустите сервис hbf-server.service командой:

Agent service start

```
systemctl enable hbf-agent.service
systemctl start hbf-agent.service
```

rpm

Настроим необходимые переменные окружения командой:

Environment setup

```
export PACKAGE_TYPE=rpm
export URL=https://github.com/H-BF/sgroups/releases/download
export RELEASE=v$VERSION/to-nft-$VERSION-any.$PACKAGE_TYPE
```

Создайте конфигурационный файл для hbf-agent командой:

Configuration file setup

```
mkdir -p /opt/swarm/etc/to-nft
cat <<EOF > /opt/swarm/etc/to-nft/config-agent.yaml
---
graceful-shutdown: 10s

logger:
  level: DEBUG

fqdn-rules:
  strategy: dns

extapi:
  svc:
    def-daial-duration: 10s
    sgroups:
      dial-duration: 3s
      address: 'tcp://${HBF_SERVER}:${HBF_SERVER_PORT}'
      sync-status:
        interval: 3s #mandatory
        push: true

netlink:
  watcher: #netlink watcher
    linger: 1s

base-rules:
  networks: ['${HBF_SERVER}/32']

dns:
  nameservers: ${DNS_SERVER}
  proto: tcp
  port: 53
  dial-duration: 3s
  read-duration: 5s #default
  write-duration: 5s #default 5s
  retries: 5 #default 1
  retry-timeout: 3s #default 1s

telemetry:
  useragent: "string"
  endpoint: 0.0.0.0:9660
```

```
  metrics:
    enable: true
  healthcheck:
    enable: true
EOF
```

Скачиваем и устанавливаем rpm-пакет командой:

Install rpm package

```
sudo wget -O /tmp/to-nft.rpm $URL/$RELEASE
sudo yum localinstall /tmp/to-nft.rpm
```

Запустите сервис hbf-agent.service командой:

Agent service start

```
systemctl enable hbf-agent.service
systemctl start hbf-agent.service
```

source

Перед развертыванием, необходимо создать директории для хранения бинарного файла и файлов конфигурации командой:

Environment setup

```
mkdir -p /opt/swarm/sbin
mkdir -p /opt/swarm/etc/to-nft
```

Клонируйте репозиторий, перейдите в директорию репозитория, переключитесь на тег необходимой версии и создайте необходимые директории командой:

Git setup

```
git clone https://github.com/H-BF/sgroups
cd sgroups
git fetch --tags --all
git checkout tags/v${VERSION}
make to-nft platform=linux/amd64
cp bin/to-nft /opt/swarm/sbin/hbf-agent
```

Создайте конфигурационный файл для hbf-agent командой:

Configuration file setup

```
cat <<EOF > /opt/swarm/etc/to-nft/config-agent.yaml
---
graceful-shutdown: 10s

logger:
  level: DEBUG

fqdn-rules:
  strategy: dns

extapi:
  svc:
    def-daial-duration: 10s
    sgroups:
      dial-duration: 3s
      address: 'tcp://${HBF_SERVER}:${HBF_SERVER_PORT}'
      sync-status:
        interval: 3s #mandatory
        push: true

netlink:
  watcher: #netlink watcher
    linger: 1s

base-rules:
  networks: ['${HBF_SERVER}/32']

dns:
  nameservers: ${DNS_SERVER}
  proto: tcp
  port: 53
  dial-duration: 3s
  read-duration: 5s #default
  write-duration: 5s #default 5s
```

```
  retries: 5 #default 1
  retry-timeout: 3s #default 1s

telemetry:
  useragent: "string"
  endpoint: 0.0.0.0:9660
  metrics:
    enable: true
  healthcheck:
    enable: true
EOF
```

Создайте конфигурационный файл для hbf-server.agent командой:

Agent service setup

```
cat <<EOF > /etc/systemd/system/hbf-agent.service
[Unit]
Description=hbf agent
After=network.target

[Service]
ExecStart=/opt/swarm/sbin/hbf-agent --config=/opt/swarm/etc/to-nft/config-agent.yaml
Restart=always
RestartSec=5
Delegate=yes
KillMode=process
OOMScoreAdjust=-999
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
EOF
```

Запустите сервис hbf-server.service командой:

Agent service start

systemctl enable hbf-agent.service

systemctl start hbf-agent.service

ansible

Перед развертыванием убедитесь, что у вас установлен ansible:

ansible-playbook --version

Далее убедитесь, что вы корректно указали версию, установив переменную VERSION без символа 'v'. Следующим шагом склонируйте репозиторий:

git clone https://github.com/H-BF/ansible

cd ansible

git fetch --tags --all

git checkout tags/v${VERSION}

Требования

Перед выполнением ansible-playbook убедитесь, что текущая версия дистрибутива ОС имеет необходимый пакет linux-headers. Как пример, на Ubuntu/Debian, вы можете проверить это с помощью команды:

apt search linux-headers-$(uname -r)

В случае, если такой пакет существует вывод будет следующим:

Sorting... Done

Full Text Search... Done

linux-headers-5.10.0-26-amd64/oldstable, now 5.10.197-1 amd64 [installed]

  Header files for Linux 5.10.0-26-amd64

Сообщение linux-headers-5.10.0-26-amd64/oldstable, now 5.10.197-1 amd64 [installed] говорит о том, что необходимый пакет заголовков найден. В противном случае необходимо обновить версию ядра до ближайшей версии имеющей пакет linux-headers-$(uname -r)

Использование

Перед запуском плейбука убедитесь, что вы добавили необходимые хосты в env/hosts. Чтобы выполнить плейбук, запустите:

ansible-playbook main.yaml

Удаление

Чтобы удалить определенные компоненты hbf-agent, установите переменную <component>_enabled: false. Если вы удаляете сам hbf-agent, все правила, созданные им в nft, будут автоматически удалены.

Параметры конфигурационного файла

Для настройки агентов требуется использовать конфигурационный файл, который содержит поля, позволяющие настраивать параметры в соответствии с потребностями пользователей. Так же есть альтернативный способ — переменные окружения. Обратите внимание, что переменные окружения являются более приоритетными, чем параметры из файла. Так же

часть параметров можно настроить через файл, а часть через переменные окружения.

Параметры файла config-agent.yaml			
Параметр / Переменная окружения	**Описание**	**Тип**	**Обязательно**
exit-on-success NFT_EXIT_ON_SUCCESS	Завершение процесса, если успешно применилась конфигурация для nft. Значение по умолчанию: false.	Boolean	Нет
continue-on-failure NFT_CONTINUE_ON_FAILURE	В случае ошибки на уровне job, агент не завершает свою работу (идет на цикл перезапуска своих job). Значение по умолчанию: true	Boolean	Нет
logger.level NFT_LOGGER_LEVEL	Уровень логгирования. Допустимые значения: TRACE, DEBUG, INFO, WARN, ERROR, FATAL, PANIC.	String	Нет
graceful-shutdown NFT_GRACEFUL_SHUTDOWN	Определяет период времени, в течение которого процесс агента должен корректно завершиться. По истечению периода, процесс завершиться с кодом 1. Значение по умолчанию: 10s	String	Нет
netns NFT_NETNS	Имя сетевого namespac'a.	String	Нет
netlink.watcher.linger NFT_NETLINK_WATCHER_LINGER	Время, в течение которого из netlink'a ожидаются события. Минимальное значение: 1s.	String	
base-rules.networks NFT_BASE_RULES_NETWORKS	Список IP адресов, к которым всегда должен egress быть доступ. Не рекомендуется устанавливать в качестве значения пустой список.	List	Да
fqdn-rules.strategy NFT_FQDN_RULES_STRATEGY	Способ обработки fqdn-правил. Значение по умолчанию: dns. Допустимые значения: dns, ndpi, combine.	String	Нет
dns.nameservers NFT_DNS_NAMESERVERS	Список доверенных IP-адресов DNS серверов.	List	Да
dns.proto NFT_DNS_PROTO	Протокол L4, используемый DNS сервером. Значение по умолчанию: udp. Допустимые значения: udp, tcp.	String	Да
dns.port NFT_DNS_PORT	Порт, используемый DNS сервером. Значение по умолчанию: 53.	Int	Да
dns.retries NFT_DNS_RETRIES	Количество повторных запросов к DNS серверу при сбое. Значение по	Int	Нет

Параметры файла config-agent.yaml			
Параметр / Переменная окружения	**Описание**	**Тип**	**Обязательно**
	умолчанию: `3`.		
dns.retry-timeout `NFT_DNS_RETRY_TIMEOUT`	Период между повторным запросом DNS сервера при сбое. Значение по умолчанию: `1s`.	String	Нет
dns.dial-duration `NFT_DNS_DIAL_DURATION`	Период времени подключения к DNS серверу. Значение по умолчанию: `3s`.	String	Нет
dns.write-duration `NFT_DNS_WRITE_DURATION`	Максимальное временя ожидания ответа от DNS сервера при запросе резолва FQDN. Значение по умолчанию: `5s`.	String	Нет
dns.read-duration `NFT_DNS_READ_DURATION`	Максимальное временя ожидания ответа от DNS сервера. Значение по умолчанию: `5s`.	String	Нет
extapi.svc.def-dial-duration `NFT_EXTAPI_SVC_DEF_DIAL_DURATION`	Длительность ожидания подключения к службе.	String	Нет
extapi.svc.sgroups.address `NFT_EXTAPI_SVC_SGROUPS_ADDRESS`	IP адрес hbf-сервера.	String	Да
extapi.svc.sgroups.dial-duration `NFT_EXTAPI_SVC_SGROUPS_DIAL_DURATION`	Продолжительность ожидания подключения к hbf-серверу.	String	Нет
extapi.svc.sgroups.sync-status.interval `NFT_EXTAPI_SVC_SGROUPS_SYNC_STATUS_INTERVAL`	Период синхронизации с hbf-сервером. Значение по умолчанию: `10s`.	String	Да
extapi.svc.sgroups.sync-status.push `NFT_EXTAPI_SVC_SGROUPS_SYNC_STATUS_PUSH`	Использовать push модель для синхронизации с hbf-сервером. Значение по умолчанию: `false`.	Boolean	Нет
telemetry.endpoint `NFT_TELEMETRY_ENDPOINT`	IP адрес и порт, для доступа к метрикам. Рекомендуемое значение: `0.0.0.0:9660`.	String	Нет
telemetry.metrics.enable `NFT_TELEMETRY_METRICS_ENABLE`	Включить/Отключить возможность получения метрик. Значение по умолчанию: `true`. Пример получения метрик: `curl 0.0.0.0:9660/metrics`.	Boolean	Нет
telemetry.healthcheck.enable `NFT_TELEMETRY_HEALTHCHECK_ENABLE`	Включить/Отключить возможность получения информации о статусе hbf-агента. Значение по умолчанию: `true`. Пример получения статуса: `curl 0.0.0.0:9660/healthcheck`.	Boolean	Нет

Параметры файла config-agent.yaml			
Параметр / Переменная окружения	**Описание**	**Тип**	**Обязательно**
telemetry.useragent `NFT_TELEMETRY_USERAGENT`	Позволяет устанавливать метку в параметры метрик	String	Нет
telemetry.profile.enable `NFT_TELEMETRY_PROFILE_ENABLE`	Включить/Отключить возможность получения профиля hbf-агента. Значение по умолчанию: `true`. Пример получения профиля: `curl 0.0.0.0:9660/debug/pprof/goroutine?debug=2`.	Boolean	Нет

Мониторинг

Информация о состоянии агента на различных узлах основывается на метриках. В таблице ниже предоставлены доступные метрики и их описания.

Зеленым цветом выделены ключевые метрики.

- потерялось

Название метрики	Тип метрики	Описание
agent_applied_configs	counter	Количество успешно примененных конфигураций
go_gc_duration_seconds	summary	Сводка длительности паузы циклов сборки мусора
go_gc_duration_seconds_sum	counter	Сумма длительности паузы циклов сборки мусора
go_gc_duration_seconds_count	counter	Количество циклов сборки мусора
go_goroutines	gauge	Количество горутин, существующих в данный момент
go_info	gauge	Информация о среде выполнения Go
go_memstats_alloc_bytes	gauge	Количество выделенных и еще используемых байтов
go_memstats_alloc_bytes_total	counter	Общее количество выделенных байтов, даже если они освобождены
go_memstats_buck_hash_sys_bytes	gauge	Количество байтов, используемых хэш-таблицей профилирования
go_memstats_frees_total	counter	Общее количество освобождений
go_memstats_gc_sys_bytes	gauge	Количество байтов, используемых для метаданных системы сборки мусора
go_memstats_heap_alloc_bytes	gauge	Количество выделенных байтов кучи и еще используемых
go_memstats_heap_idle_bytes	gauge	Количество байтов кучи в ожидании использования
go_memstats_heap_inuse_bytes	gauge	Количество используемых байтов кучи
go_memstats_heap_objects	gauge	Количество выделенных объектов
go_memstats_heap_released_bytes	gauge	Количество байтов кучи, освобожденных в ОС
go_memstats_heap_sys_bytes	gauge	Количество байтов кучи, полученных от системы
go_memstats_last_gc_time_seconds	gauge	Количество секунд с 1970 года последней сборки мусора
go_memstats_lookups_total	counter	Общее количество запросов указателей
go_memstats_mallocs_total	counter	Общее количество выделений памяти
go_memstats_mcache_inuse_bytes	gauge	Количество байтов, используемых структурами mcache
go_memstats_mcache_sys_bytes	gauge	Количество байтов, используемых структурами mcache, полученных от системы
go_memstats_mspan_inuse_bytes	gauge	Количество байтов, используемых структурами mspan
go_memstats_mspan_sys_bytes	gauge	Количество байтов, используемых структурами mspan, полученных от системы
go_memstats_next_gc_bytes	gauge	Количество байтов кучи, когда произойдет следующая сборка мусора

go_memstats_other_sys_bytes	gauge	Количество байтов, используемых для других системных выделений
go_memstats_stack_inuse_bytes	gauge	Количество байтов, используемых аллокатором стека
go_memstats_stack_sys_bytes	gauge	Количество байтов, полученных от системы для аллокатора стека
go_memstats_sys_bytes	gauge	Количество байтов, полученных от системы
go_threads	gauge	Количество созданных потоков ОС
healthcheck	gauge	Индикатор проверки состояния процесса (0 или 1)
process_cpu_seconds_total	counter	Общее количество времени процессора, затраченного в секундах
process_max_fds	gauge	Максимальное количество открытых файловых дескрипторов
process_open_fds	gauge	Количество открытых файловых дескрипторов
process_resident_memory_bytes	gauge	Размер резидентной памяти в байтах
process_start_time_seconds	gauge	Время запуска процесса с начала эпохи Unix в секундах
process_virtual_memory_bytes	gauge	Размер виртуальной памяти в байтах
process_virtual_memory_max_bytes	gauge	Максимальное количество виртуальной памяти в байтах
promhttp_metric_handler_errors_total	counter	Общее количество http ошибок, выявленных обработчиком

Настройка TLS

Установка
Настройка TLS (Transport Layer Security) на hbf-агенте обеспечивает шифрование трафика между сервером и клиентом, что повышает безопасность передаваемых данных. В этой документации описан процесс настройки TLS на hbf-агенте, включая использование предоставленного конфигурационного файла. * НЕ ОТСЮДА БЛОК

Прежде чем приступить к настройке TLS, убедитесь, что у вас есть:

1. Установленный hbf-агент
2. Включен и настроен TLS на hbf-сервере
3. Сертификат SSL и соответствующий приватный ключ. Если у вас их нет, вы можете получить их у сертификационного центра (CA) или создать самоподписанный сертификат для тестовых целей.

Шаги по настройке TLS
Создайте файл конфигурации hbf-агента для редактирования:

sudo nano /etc/cmd/sgroups/tls-config.yaml

Далее необходимо настроить секцию для TLS:

Insecure TLS

```
extapi:
  svc:
    authn:
      type: tls
      tls:
        server:
          verify: false
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls tls включен.

tls.key-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

tls.cert-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: true или false. При значении true включен режим проверки

сертификата сервера, при значении false данный режим отключен.ё

Secure TLS

```
extapi:
  svc:
    authn:
      type: tls
      tls:
        server:
          verify: true
          name: "server-name"
          ca-files: ["file1.pem", "file2.pem", ...]
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls tls включен.

tls.key-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

tls.cert-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: true или false. При значении true включен режим проверки сертификата сервера, при значении false данный режим отключен.

name — При включенном режиме проверки сертификата сервера verify: true можно указать имя сервера. Поле не обязательное для заполнения, в случае если имя сервера не будет указано то подлинность будет проверяться по данным сертификата.

ca-files — При включенном режиме проверки сертификата сервера verify: true необходимо перечислить список certificates authorities с указанием относительного или полного пути к файлам.

mTLS

```
extapi:
  svc:
    authn:
      type: tls
      tls:
        key-file: "/etc/ssl/private/key-file.pem"
        cert-file: "/etc/ssl/certs/cert-file.pem"
        server:
          verify: true
          name: "server-name"
          ca-files: ["file1.pem", "file2.pem", ...]
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls tls включен.

key-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — В случае если на сервере стоит настройка verify: cert-required/verify, то в этом случае необходимо установить сертификаты со стороны клиента. Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: true или false. При значении true включен режим проверки сертификата сервера, при значении false данный режим отключен.

name — При включенном режиме проверки сертификата сервера verify: true можно указать имя сервера. Поле не обязательное для заполнения, в случае если имя сервера не будет указано то подлинность будет проверяться по данным сертификата.

ca-files — При включенном режиме проверки сертификата сервера verify: true необходимо перечислить список certificates authorities с указанием относительного или полного пути к файлам.

Nftables

IPSet

Описание

IPSet представляет собой структуру данных, позволяющую хранить и организовывать множество элементов для использования в правилах nftables. Он используется для обработки больших списков элементов с минимальными затратами на ресурсы. В нашем случае IPSet формируется для описания массива IP-адресов, относящихся к FQDN или Security Groups, для дальнейшего использования в описаниях правил.

Наименование параметра	Описание	Значения
IPSet_Name	Наименование IPSet	Должно соответствовать одному из шаблонов: • **NetIPv4-${NAME}** — для описания массивов IP адресов типа v4 • **NetIPv6-${NAME}** — для описания массивов IP адресов типа v6
type	Описывает тип данных	Могут быть установлены следующие значения: • **ipv4_addr** — для описания массивов IP адресов типа v4 • **ipv6_addr** — для описания массивов IP адресов типа v6
flags	Описывает свойства IPSet	Установлены следующие значения: • **constant** — флаг используется если значение элементов в множестве являются постоянными и не могут быть изменены • **interval** — флаг используется для создания диапазона элементов множества

Наименование параметра	Описание	Значения
elements	Указывает массив содержащихся в IPSet элементов подсетей (CIDR)	Значения CIDR, в случае нескольких значений перечисляются через запятую

Шаблон

```
set ${IPSet_Name} {
   type ${type}
   flags ${flags}
   elements ${elements}
}
```

Пример использования

Security Group

```
set NetIPv4-sg-local-example {
   type ipv4_addr
   flags constant,interval
   elements = { 10.168.24.0/23 }
}
```

FQDN

```
set NetIPv4-fqdn-example.com {
   type ipv4_addr
   flags interval
   elements = { 10.10.24.0 }
}
```

Chains

В нашей реализации структуры мы вводим для INPUT и OUTPUT понятие 2-х цепочек. Первая цепочка является точкой входа для пакетов из сетевого стека, в ней указывается хук (input, prerouting, postrouting) и приоритет выполнения, так же эта цепочка используется для маршрутизации в последующие цепочки по принадлежности к той или иной Security Group. Вторая цепочка содержит наборы правил, относящиеся только к конкретной Security Group.

Наименование цепочки	Тип	Описание
INGRESS-INPUT	Ingress	*Первая* цепочка является точкой входа для пакетов из сетевого стека, в ней указывается хук (input) и приоритет выполнения 0 (filter). Так же используется для маршрутизации в последующие цепочки по принадлежности к той или иной Security Group.
INGRESS-INPUT-$sgName	Ingress	*Вторая* цепочка содержит наборы правил, относящиеся только к конкретной Security Group.
EGRESS-POSTROUTING	Egress	*Первая* цепочка является точкой входа для пакетов из сетевого стека, в ней указывается хук (postrouting) и приоритет выполнения 300. Так же используется для маршрутизации в последующие цепочки по принадлежности к той или иной Security Group.
EGRESS-POSTROUTING-$sgName	Egress	*Вторая* цепочка содержит наборы правил, относящиеся только к конкретной Security Group.

INGRESS-INPUT

Описание
Правило перехода в цепочку INGRESS-INPUT-sgName с проверкой что трафик является входящим и предназначен для указанной Security Group.

Параметры

Шаблон параметра	Структура параметра	Значение	Описание
$ConntrackState	ct state	established,related	Оптеделяет правило для обработки пакетов, удовлетворяющих условиям установленного и связанного состояния соединения.
$CtVerdict		accept	$CtVerdict — указывает на принятие (accept) пакетов по указанным условиям. *Подробнее: Verdict statement*
$BaseRules			*Base Rules — набор правил, которые прописываются статично из конфиграционнного файла агента для того что бы всегда был доступ до высококритичных сервисов таких как HBF и DNS.* *Подробнее: Config Base Rules*
$RuleType	ip		Указатель на трафик типа IP
$DstSgroup	daddr	@${IPSet({sgName})}	Значение типа string, не должно содержать в себе пробелов
$sgName			Название Security Group

Шаблон параметра	Структура параметра	Значение	Описание
$Counter	counter	packets 0 bytes 0	Счетчик, учитывает количество пройденных пакетов с количеством байтов переданной информации в рамках указанной цепочки правил
$PolicyVerdict	policy	drop	*Policy $Verdict устанавливается для цепочек с целью установки базового правила, которое будет применено к пакету если установленное правило не удовлетворили условия. По умолчанию drop.* *Подробнее: Verdict statement*
$Verdict		goto	*Так как данное правило используется для проверки типа трафика то переход на другую цепочку правил происходит только с помощью goto.* *Подробнее: Verdict statement*
$Hook	hook	input	Приоритет выполнения цепочки характеризующий стадию прохождения трафика
$HookPriority	priority	0	Приоритет выполнения цепочки одного типа

Шаблон

```
chain INGRESS-INPUT {
   type filter $Hook $HookPriority; $PolicyVerdict;
   $ConntrackState $Counter $CtVerdict
   $BaseRules
   # **********
   $RuleType $DstSgroup $Counter $Verdict INGRESS-INPUT-$sgName
   # **********
   $Counter
}
```

Пример использования

```
chain INGRESS-INPUT {
   type filter hook input priority 0; policy drop;
   ct state established,related counter packets 0 bytes 0 accept
   ip saddr { 1.1.1.1, 2.2.2.2} accept
   # **********
   ip daddr @NetIPv4-exampleSG counter packets   0 bytes 0 goto INGRESS-INPUT-exampleSG
   # **********
   counter packets 0 bytes 0
}
```

```
table inet main-1705582480 {

   chain INGRESS-INPUT {
      type filter hook input priority filter; policy drop;
      ct state established,related counter packets 0 bytes 0 accept
      ip saddr { 1.1.1.1, 2.2.2.2} accept
      # ******
      ip daddr @NetIPv4-no-routed counter packets   0 bytes 0 goto INGRESS-INPUT-no-routed
      ip daddr @NetIPv4-exampleSG counter packets   0 bytes 0 goto INGRESS-INPUT-
exampleSG
      counter packets 0 bytes 0
   }

   chain INGRESS-INPUT-no-routed {
      # ******
      counter packets 0 bytes 0 accept
   }

   chain INGRESS-INPUT-exampleSG {
      # ******
      counter packets 0 bytes 0 accept
   }

}
```

EGRESS-POSTROUTING

Описание
Правило перехода в цепочку EGRESS-POSTROUTING-$sgName с проверкой что трафик является исходящим и предназначен для указанной Security Group.

Параметры

Шаблон параметра	Структура параметра	Значение	Описание
$ConntrackState	`ct state`	`established,related`	Оптеделяет правило для обработки пакетов, удовлетворяющих условиям установленного и связанного состояния соединения.
$CtVerdict		`accept`	$CtVerdict — указывает на принятие (accept) пакетов по указанным условиям. *Подробнее:* *Verdict statement*
$BaseRules			*Base Rules — набор правил, которые прописываются статично из конфиграционнного файла агента для того что бы всегда был доступ до высококритичных сервисов таких как HBF и DNS.* *Подробнее:* *Config Base Rules*
$RuleType	`ip`		Указатель на трафик типа IP

Шаблон параметра	Структура параметра	Значение	Описание
$SrcSgroup	`saddr`	`@${IPSet({sgName})}`	Значение типа string, не должно содержать в себе пробелов
$sgName			Название Security Group
$Counter	`counter`	`packets 0 bytes 0`	Счетчик, учитывает количество пройденных пакетов с количеством байтов переданной информации в рамках указанной цепочки правил
$PolicyVerdict	`policy`	`drop`	*Policy $Verdict устанавливается для цепочек с целью установки базового правила, которое будет применено к пакету если установленное правило не удовлетворили условия. По умолчанию drop.* *Подробнее: Verdict statement*
$Verdict		`goto`	*Так как данное правило используется для проверки типа трафика то переход на другую цепочку правил происходит только с помощью goto.* *Подробнее: Verdict statement*
$Hook	`hook`	input	Приоритет выполнения цепочки характерезующий стадию прохождения трафика
$HookPriority	`priority`	0	Приоритет выполнения цепочки одного типа

Шаблон

```
chain EGRESS-POSTROUTING {
   type filter $Hook $HookPriority; $PolicyVerdict;
   $ConntrackState $Counter $CtVerdict
   $BaseRules
   # **********
   $RuleType $SrcSgroup $Counter $Verdict EGRESS-POSTROUTING-$sgName
   # **********
   $Counter
}
```

Пример использования

```
chain EGRESS-POSTROUTING {
   type filter hook postrouting priority 300; policy drop;
   ct state established,related counter packets 0 bytes 0 accept
   ip daddr { 1.1.1.1, 2.2.2.2} accept
   # **********
   ip saddr @NetIPv4-exampleSG counter packets 0 bytes 0 goto EGRESS-POSTROUTING-
exampleSG
```

```
    # **********
    counter packets 0 bytes 0
}


table inet main-1705582480 {

  chain EGRESS-POSTROUTING {
      type filter hook postrouting priority 300; policy drop;
      ct state established,related counter packets 0 bytes 0 accept
      ip daddr { 1.1.1.1, 2.2.2.2} accept
      # **********
      ip saddr @NetIPv4-exampleSG counter packets 0 bytes 0 goto EGRESS-POSTROUTING-
exampleSG
      ip saddr @NetIPv4-no-routed counter packets 0 bytes 0 goto EGRESS-POSTROUTING-no-
routed
      counter packets 0 bytes 0
  }

  chain EGRESS-POSTROUTING-no-routed {
      # ******
      counter packets 0 bytes 0 accept
  }

  chain EGRESS-POSTROUTING-exampleSG {
      # ******
      counter packets 0 bytes 0 accept
  }
```

Verdict statement

Значение	Описание
accept	Терминирующие правило которое разрещает трафик, попавший под это условие, и завершает обработку пакета в текущей таблице.
drop	Терминирующие правило которое запрещает трафик, попавший под это условие, и завершает обработку пакета в текущей таблице.
goto *chain*	Переход на другую цепочку в рамках указанного правила. После завершения правил в этой цепочке обратно в ту же цепочку трафик попасть не сможет.
jump *chain*	Переход на другую цепочку в рамках указанного правила. После завершения правил в этой цепочке трафик вернется в исходную цепочку.

Icmp-дескрипторы

IPv4

Значение	Дескриптор
0	Echo Reply

Значение	Дескриптор
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench (Deprecated)
5	Redirect
6	Alternate Host Address (Deprecated)
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request (Deprecated)
16	Information Reply (Deprecated)
17	Address Mask Request (Deprecated)
18	Address Mask Reply (Deprecated)
19	Reserved (for Security)
20-29	Reserved (for Robustness Experiment)
30	Traceroute (Deprecated)
31	Datagram Conversion Error (Deprecated)
32	Mobile Host Redirect (Deprecated)
33	IPv6 Where-Are-You (Deprecated)
34	IPv6 I-Am-Here (Deprecated)
35	Mobile Registration Request (Deprecated)
36	Mobile Registration Reply (Deprecated)
37	Domain Name Request (Deprecated)
38	Domain Name Reply (Deprecated)
39	SKIP (Deprecated)
40	Photuris
41	ICMP messages utilized by experimental mobility protocols such as Seamoby
42	Extended Echo Request
43	Extended Echo Reply
44-252	Unassigned
253	RFC3692-style Experiment 1
254	RFC3692-style Experiment 2
255	Reserved

IPv6

Значение	Дескриптор
0	Reserved

Значение	Дескриптор
1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem
5-99	Unassigned
100	Private experimentation
101	Private experimentation
102-126	Unassigned
127	Reserved for expansion of ICMPv6 error messages
128	Echo Request
129	Echo Reply
130	Multicast Listener Query
131	Multicast Listener Report
132	Multicast Listener Done
133	Router Solicitation
134	Router Advertisement
135	Neighbor Solicitation
136	Neighbor Advertisement
137	Redirect Message
138	Router Renumbering
139	ICMP Node Information Query
140	ICMP Node Information Response
141	Inverse Neighbor Discovery Solicitation Message
142	Inverse Neighbor Discovery Advertisement Message
143	Version 2 Multicast Listener Report
144	Home Agent Address Discovery Request Message
145	Home Agent Address Discovery Reply Message
146	Mobile Prefix Solicitation
147	Mobile Prefix Advertisement
148	Certification Path Solicitation Message
149	Certification Path Advertisement Message
150	ICMP messages utilized by experimental mobility protocols such as Seamoby
151	Multicast Router Advertisement
152	Multicast Router Solicitation
153	Multicast Router Termination
154	FMIPv6 Messages
155	RPL Control Message
156	ILNPv6 Locator Update Message
157	Duplicate Address Request
158	Duplicate Address Confirmation
159	MPL Control Message
160	Extended Echo Request
161	Extended Echo Reply
162-199	Unassigned
200	Private experimentation

Значение	Дескриптор
201	Private experimentation
202-254	Unassigned
255	Reserved for expansion of ICMPv6 informational messages

Config Base Rules

Описание

Base Rules — набор правил, которые прописываются статично из конфиграционнного файла агента для того что бы всегда был доступ до высококритичных сервисов таких как HBF и DNS.

Параметры

Шаблон параметра	Структура параметра	Значение	Описание
${SrcCIDR}	`saddr {CIDR}`	Массив подсетей	Список сетей в которые разрешаем трафик
${DstCIDR}	`daddr {CIDR}`	Массив подсетей	Список сетей в которые разрешаем трафик
${RuleType}	`ip`		Описывает, что принимает трафик типа ip
${Verdict}	`accept`	Не параметризированный ** Так как данное правило используется для проверки типа трафика то переход на другую цепочку правил происходит только с помощью goto.* *Подробнее: Verdict statement*	Вердикт политики по пакетам данных

Конфигурационный файл

```
base-rules:
  networks:
    - '1.1.1.1'
    - '2.2.2.2'
```

Шаблон

```
chain EGRESS-POSTROUTING {
   ${RuleType} ${DstCIDR} ${Verdict}
   # **********
}

chain INGRESS-INPUT {
   ${RuleType} ${SrcCIDR} ${Verdict}
   # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING {
   ip daddr { 1.1.1.1, 2.2.2.2} accept
   # **********
}

chain INGRESS-INPUT {
```

Правило	Значение приоритета
Sgroup to Sgroup (icmp) (legacy)	-300
Sgroup to Sgroup (tcp|udp) (legacy)	-200
Sgroup to Sgroup (icmp) (ingress/egress)	-100
Sgroup to Sgroup (tcp|udp) (ingress/egress)	0
Sgroup to FQDN (tcp|udp)	100
Sgroup to CIDR (icmp) (ingress/egress)	200
Sgroup to CIDR (tcp|udp) (ingress/egress)	300

```
   ip saddr { 1.1.1.1, 2.2.2.2} accept
   # **********
}
```

Приоритет правил

Приоритет правил в nftables определяет порядок в котором правила применяются к пакетам или сетевым потокам. Чем ниже значение преоритета, тем выше преоритет имеет правило. Значение приоритета могут варьироваться от -32768 до 32767, однако обычно они ограничиваются диапазоном от -300 до 300 для удобства управления и понимания.
Ниже представлена птаблица преодитета правил в nftables:

HBF-сервер

Установка

Перед развертыванием, пользователь должен решить, где hbf-server будет хранить данные. Предоставляется два варианта для выбора:

- In Memory.
- Postgres: В этом случае необходимо перед запуском hbf-server выполнить миграцию базы данных.

Независимо от выбранного метода установки hbf-server, необходимо сконфигурировать следующие переменные окружения:
HBF_SERVER - IP-адрес hbf-server.

Setup HBF_SERVER

export HBF_SERVER=0.0.0.0

HBF_SERVER_PORT — порт, используемый hbf-server.

Setup HBF_SERVER_PORT

export HBF_SERVER_PORT=9650

VERSION — версия hbf-server.

Setup VERSION

export VERSION=1.9.1

SG_STORAGE_TYPE — тип хранилища для hbf-server. Допустимые значени: internal или postgres.

В случае выбора типа хранилища "In Memory" установите переменную окружения командой:

Setup SG_STORAGE_TYPE

export SG_STORAGE_TYPE=internal

В случае выбора типа хранилища "Postgres" установите переменную окружения командой:

Setup SG_STORAGE_TYPE

```
export SG_STORAGE_TYPE=postgres
```

SG_STORAGE_POSTGRES_URL — URL для подключения к Postgres базе данных. Для корректного формирования воспользуйтесь командой:

Setup SG_STORAGE_POSTGRES_URL

```
cat <<EOF > setup.sh
#!/bin/bash
SG_DB_USER="user"
SG_DB_PASSWORD="password"
SG_DB_URL="localhost:5432"
SG_DB_NAME="swarm"

SG_STORAGE_POSTGRES_URL="postgres://${SG_DB_USER}:${SG_DB_PASSWORD}@${
SG_DB_URL}/${SG_DB_NAME}?sslmode=disable"

export SG_STORAGE_POSTGRES_URL=$SG_STORAGE_POSTGRES_URL
EOF
sh setup.sh
```

docker

Перед развертыванием убедитесь, что у вас установлен **docker**:

Get docker version

```
docker -v
```

Клонируем репозиторий, переходим в директорию репозитория и переключаемся на тег необходимой версии командой:

Git setup

```
git clone https://github.com/H-BF/sgroups
cd sgroups
git fetch --tags --all
git checkout tags/v${VERSION}
```

Создайте конфигурационный файл для hbf-server командой:

Configuration file setup

```
cat <<EOF > config-server.yaml
---
logger:
  # log level
  level: INFO

metrics:
  # enable api metrics
  enable: true

healthcheck:
  # enables|disables health check handler
  enable: true

server:
  # server endpoint
  endpoint: tcp://${HBF_SERVER}:${HBF_SERVER_PORT}
  # graceful shutdown period
  graceful-shutdown: 30s
EOF
```

Собираем образ hbf-server командой:

```
docker build -f docker/Dockerfile.server --tag sgroups:v$(cat VERSION) .
```

В случае выбора типа хранилища "In Memory" запустите hbf-server командой:

```
docker run \
-d \
-v ./config-server.yaml:/opt/swarm/etc/sgroups/config-server.yaml \
--name hbf-server \
--entrypoint "/app/bin/sgroups" \
--env SG_STORAGE_TYPE \
sgroups:v$(cat VERSION) -config /opt/swarm/etc/sgroups/config-server.yaml
```

В случае выбора типа хранилища "Postgres" запустите hbf-server командой:

```
sh setup.sh
docker run \
-d \
-v ./config-server.yaml:/opt/swarm/etc/sgroups/config-server.yaml \
--name hbf-server \
--entrypoint "/app/bin/sgroups" \
--env SG_STORAGE_TYPE \
--env SG_STORAGE_POSTGRES_URL \
sgroups:v$(cat VERSION) -config /opt/swarm/etc/sgroups/config-server.yaml
```

deb

Устанавливаем дополнительные переменные окружения командой:

Environment setup
export PACKAGE_TYPE=deb
export URL=https://github.com/H-BF/sgroups/releases/download
export RELEASE=v$VERSION/sgroups-$VERSION-any.$PACKAGE_TYPE

Скачиваем и устанавливаем deb-пакет командой:

Install deb package

sudo wget -O /tmp/sgroups.deb $URL/$RELEASE
sudo dpkg -i /tmp/sgroups.deb

Создайте конфигурационный файл для hbf-server.service командой:

Server service setup

cat <<EOF > /etc/systemd/system/hbf-server.service
[Unit]
Description=sgroups
Documentation=TODO:
After=network.target

[Service]
EnvironmentFile=/opt/swarm/etc/sgroups/env
ExecStart=/opt/swarm/sbin/sgroups --config=/opt/swarm/etc/sgroups/config.yaml
Restart=always
RestartSec=5
Delegate=yes
KillMode=process
OOMScoreAdjust=-999
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
EOF

Создайте конфигурационный файл для hbf-server командой:

Setup server service

cat <<EOF > /opt/swarm/etc/sgroups/config.yaml

```
---
logger:
  # log level
  level: INFO

metrics:
  # enable api metrics
  enable: true

healthcheck:
  # enables|disables health check handler
  enable: true

server:
  # server endpoint
  endpoint: tcp://${HBF_SERVER}:${HBF_SERVER_PORT}
  # graceful shutdown period
  graceful-shutdown: 30s
EOF
```

В случае выбора типа хранилища "In Memory" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:

Environment file setup

```
mkdir -p /opt/swarm/etc/sgroups
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
EOF
```

В случае выбора типа хранилища "Postgres" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:

Environment file setup

```
mkdir -p /opt/swarm/etc/sgroups
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
SG_STORAGE_POSTGRES_URL=${SG_STORAGE_POSTGRES_URL}
EOF
```

Запустите сервис hbf-server.service командой:

Server service start

```
systemctl enable hbf-server.service
```

systemctl start hbf-server.service

rpm

Настроим необходимые переменные окружения командой:

bash

```
export PACKAGE_TYPE=rpm
export URL=https://github.com/H-BF/sgroups/releases/download
export RELEASE=v$VERSION/sgroups-$VERSION-any.$PACKAGE_TYPE
```

Скачиваем и устанавливаем rpm-пакет командой:

Install rpm package

```
sudo wget -O /tmp/sgroups.rpm $URL/$RELEASE
sudo yum localinstall -i /tmp/sgroups.rpm
```

Создайте конфигурационный файл для hbf-server.service командой:

Server service setup

```
cat <<EOF > /etc/systemd/system/hbf-server.service
[Unit]
Description=sgroups
Documentation=TODO:
After=network.target

[Service]
EnvironmentFile=/opt/swarm/etc/sgroups/env
ExecStart=/opt/swarm/sbin/sgroups --config=/opt/swarm/etc/sgroups/config.yaml
Restart=always
RestartSec=5
Delegate=yes
KillMode=process
OOMScoreAdjust=-999
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
EOF
```

Создайте конфигурационный файл для hbf-server командой:

Setup server service

```
cat <<EOF > /opt/swarm/etc/sgroups/config.yaml
---
logger:
  # log level
  level: INFO

metrics:
  # enable api metrics
  enable: true

healthcheck:
  # enables|disables health check handler
  enable: true

server:
  # server endpoint
  endpoint: tcp://${HBF_SERVER}:${HBF_SERVER_PORT}
  # graceful shutdown period
  graceful-shutdown: 30s
EOF
```

В случае выбора типа хранилища "In Memory" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:

Environment file setup

```
mkdir -p /opt/swarm/etc/sgroups
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
EOF
```

В случае выбора типа хранилища "Postgres" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:
Environment file setup

```
mkdir -p /opt/swarm/etc/sgroups
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
SG_STORAGE_POSTGRES_URL=${SG_STORAGE_POSTGRES_URL}
EOF
```

Запустите сервис hbf-server.service командой:

Server service start

```
systemctl enable hbf-server.service
```

systemctl start hbf-server.service

source

Перед развертыванием, необходимо создать директории для хранения бинарного файла и файлов конфигурации командой:

Environment setup

```
mkdir -p /opt/swarm/sbin
mkdir -p /opt/swarm/etc/to-nft
```

Клонируем репозиторий, переходим в директорию репозитория, переключаемся на тег необходимой версии и Создайте необходимые директории командой:

Git setup

```
git clone https://github.com/H-BF/sgroups
cd sgroups
git fetch --tags --all
git checkout tags/v${VERSION}
make sg-service platform=linux/amd64
cp bin/sgroups /opt/swarm/sbin/hbf-server
```

Создайте конфигурационный файл для hbf-server командой:

Configuration file setup

```
cat <<EOF > /opt/swarm/etc/sgroups/config-server.yaml
---
logger:
  # log level
  level: INFO

metrics:
  # enable api metrics
  enable: true

healthcheck:
  # enables|disables health check handler
  enable: true

server:
  # server endpoint
```

```
  endpoint: tcp://${HBF_SERVER}:${HBF_SERVER_PORT}
  # graceful shutdown period
  graceful-shutdown: 30s
EOF
```

Создайте конфигурационный файл для hbf-server.service командой:

Server service setup

```
cat <<EOF > /etc/systemd/system/hbf-server.service
[Unit]
Description=sgroups
Documentation=TODO:
After=network.target

[Service]
EnvironmentFile=/opt/swarm/etc/sgroups/env
ExecStart=/opt/swarm/sbin/hbf-server --config=/opt/swarm/etc/sgroups/config-server.yaml
Restart=always
RestartSec=5
Delegate=yes
KillMode=process
OOMScoreAdjust=-999
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity

[Install]
WantedBy=multi-user.target
EOF
```

В случае выбора типа хранилища "In Memory" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:

Environment file setup

```
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
export SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
EOF
```

В случае выбора типа хранилища "Postgres" необходимо настроить файл /opt/swarm/etc/sgroups/env командой:

Environment file setup

```
cat <<EOF > /opt/swarm/etc/sgroups/env
#!/bin/bash
SG_STORAGE_TYPE=${SG_STORAGE_TYPE}
```

SG_STORAGE_POSTGRES_URL=${SG_STORAGE_POSTGRES_URL}
EOF

Запустите сервис hbf-server.service командой:
Server service start
systemctl enable hbf-server.service
systemctl start hbf-server.service

Параметры конфигурационного файла

Для настройки агентов требуется использовать конфигурационный файл, который содержит поля, позволяющие настраивать параметры в соответствии с потребностями пользователей. Так же есть альтернативный способ — переменные окружения. Обратите внимание, что переменные окружения являются более приоритетными, чем параметры из файла. Так же часть параметров можно настроить через файл, а часть через переменные окружения.

Параметры файла config-server.yaml

Параметр	Описание	Тип	Обязательно
logger.level `SG_LOGGER_LEVEL`	Уровень логгирования. Допустимые значения: `TRACE`, `DEBUG`, `INFO`, `WARN`, `ERROR`, `FATAL`, `PANIC`.	String	Нет
server.endpoint `SG_SERVER_ENDPOINT`	IP адрес и порт hbf-сервер. Рекомендуемое значение: `0.0.0.0:9650`.	String	Да
server.graceful-shutdow `SG_SERVER_GRACEFUL_SHUTDOW`	Определяет период времени, в течение которого процесс сервера должен корректно завершиться. По истечению периода, процесс завершиться с кодом 1. Значение по умолчанию: `10s`	String	Нет
telemetry.metrics.enable `SG_METRICS_ENABLE`	Включить/Отключить возможность получения метрик. Значение по умолчанию: `true`. Пример получения метрик: `curl 0.0.0.0:9650/metrics`.	Boolean	Нет
healthcheck.enable `SG_HEALTHCHECK_ENABLE`	Включить/Отключить возможность получения информации о статусе hbf-сервера. Значение по умолчанию: `true`. Пример получения статуса: `curl 0.0.0.0:9650/healthcheck`.	Boolean	Нет
storage.type `SG_STORAGE_TYPE`	Подключаемый тип базы данных. Значение по умолчанию: `internal`. Допустимые значения: `internal`, `postgres`.	String	Нет
storage.postgres.url `SG_STORAGE_POSTGRES_URL`	URL для подключения к базе данных Postgres. При использовании `storage.type:postgres`, данный параметр должен иметь корректный URL подключения. В ином случае, в параметре нет необходимости.	String	Нет

Миграция

В качестве инструмента для миграции базы данных использует goose. Когда в качестве хранилища используется Postgres, goose запускается перед началом работы hbf-server'a. Более подробную информацию о goose можно найти здесь.

Перед запуском goose необходимо настроить скрипт миграции командой:

```
cat <<EOF > migration.sh
#!/bin/bash
SG_DB_USER="user"
SG_DB_PASSWORD="password"
SG_DB_URL="localhost:5432"
SG_DB_NAME="swarm"

SG_STORAGE_POSTGRES_URL="postgres://${SG_DB_USER}:${SG_DB_PASSWORD}@${
SG_DB_URL}/${SG_DB_NAME}?sslmode=disable"

export SG_STORAGE_POSTGRES_URL=$SG_STORAGE_POSTGRES_URL

exec ./bin/goose postgres $SG_STORAGE_POSTGRES_URL up
EOF
```

source

Сборка

Перед сборкой необходимо склонировать репозиторий

```
export VERSION=1.9.1
git clone https://github.com/H-BF/sgroups
cd sgroups
git fetch --tags --all
git checkout tags/v${VERSION}
make .install-goose
```

После сборки, скомпилированный бинарный файл будет доступен в папке ./bin/goose в корне репозитория (не путать с /bin, расположенным в корне файловой системы).

По умолчанию файлы миграции расположены в папке ./internal/registry/sgroups/pg/migrations, но пользователь может указать и другой путь, определив переменную окружения PG_MIGRATIONS.

docker

Сборка
Перед развертыванием убедитесь, что у вас установлен **docker**:

Get docker version

docker -v

Клонируем репозиторий, переходим в директорию репозитория и переключаемся на тег необходимой версии командой:

```
export VERSION=1.9.1
git clone https://github.com/H-BF/sgroups
cd sgroups
git fetch --tags --all
git checkout tags/v${VERSION}
```

Собираем образ *goose* командой:

docker build -f docker/Dockerfile.goose --tag goose:v$(cat VERSION) .

По умолчанию файлы миграции расположены в папке ./internal/registry/sgroups/pg/migrations, но пользователь может указать и другой путь, определив переменную окружения PG_MIGRATIONS.

Используемые переменные		
Название	**Описание**	**Значение по умолчанию**
SG_STORAGE_POSTGRES_URL	Переменная необходимая для подключения к БД	
PG_MIGRATIONS	Переменная, содержащая путь к файлам миграций	"./internal/registry/sgroups/pg/migrations"
SG_DB_USER	Имя пользователя БД	
SG_DB_PASSWORD	Пароль пользователя БД	
SG_DB_URL	Адрес базы данных (HOST:PORT)	
SG_DB_NAME	Имя БД	swarm

Запуск

Для выполнения миграции базы данных с помощью Docker необходимо выполнить команду:

```
docker run -d -t --name=goose-migration --rm \
-v "./migration.sh:/app/migration.sh" \
--entrypoint /app/migration.sh \
goose:$(cat VERSION)
```

Мониторинг

Информация о состоянии сервера основывается на метриках. В таблице ниже предоставлены доступные метрики и их описания.

Зеленым цветом выделены ключевые метрики. *ДУБЛЬ

Metric Name	Metric Type	Description
go_gc_duration_seconds	summary	Сводка длительности пауз циклов сборки мусора
go_gc_duration_seconds_count	counter	Сводка длительности пауз циклов сборки мусора
go_gc_duration_seconds_sum	counter	Сумма длительности паузы циклов сборки мусора
go_goroutines	gauge	Количество текущих горутин
go_info	gauge	Информация о среде выполнения
go_memstats_alloc_bytes	gauge	Количество выделенных и все еще используемых байтов
go_memstats_alloc_bytes_total	counter	Общее количество выделенных байтов, даже если они были освобождены
go_memstats_buck_hash_sys_bytes	gauge	Количество байтов, используемых хэш-таблицей профилирования
go_memstats_frees_total	counter	Общее количество "освобожденных" объектов кучи
go_memstats_gc_sys_bytes	gauge	Количество байтов, используемых для метаданных системы сборки мусора
go_memstats_heap_alloc_bytes	gauge	Количество выделенных и все еще используемых байтов кучи
go_memstats_heap_idle_bytes	gauge	Количество байтов кучи в ожидании использования
go_memstats_heap_inuse_bytes	gauge	Количество байтов кучи, используемых в данный момент
go_memstats_heap_objects	gauge	Количество выделенных объектов
go_memstats_heap_released_bytes	gauge	Количество байтов кучи, освобожденных в ОС
go_memstats_heap_sys_bytes	gauge	Количество байтов кучи, полученных от системы
go_memstats_last_gc_time_seconds	gauge	Количество секунд с 1970 года последней сборки мусора

Metric Name	Metric Type	Description
go_memstats_lookups_total	counter	Общее количество поисков указателей
go_memstats_mallocs_total	counter	Общее количество выделений памяти
go_memstats_mcache_inuse_bytes	gauge	Количество байтов, используемых структурами mcache
go_memstats_mcache_sys_bytes	gauge	Количество байтов, используемых структурами mcache, полученных из системы
go_memstats_mspan_inuse_bytes	gauge	Количество байтов, используемых структурами mspan
go_memstats_mspan_sys_bytes	gauge	Количество байтов, используемых структурами mspan, полученных из системы
go_memstats_next_gc_bytes	gauge	Количество байтов кучи при следующей сборке мусора
go_memstats_other_sys_bytes	gauge	Количество байтов, используемых для других системных выделений
go_memstats_stack_inuse_bytes	gauge	Количество байтов, используемых выделителем стека
go_memstats_stack_sys_bytes	gauge	Количество байтов, полученных от системы для выделителя стека
go_memstats_sys_bytes	gauge	Количество байтов, полученных от системы
go_threads	gauge	Количество созданных ОС потоков
healthcheck	gauge	Индикатор проверки состояния процесса (0 или 1)
process_cpu_seconds_total	counter	Общее количество времени CPU пользователя и системы в секундах
process_max_fds	gauge	Максимальное количество открытых дескрипторов файлов
process_open_fds	gauge	Количество открытых дескрипторов файлов
process_resident_memory_bytes	gauge	Размер резидентной памяти в байтах
process_start_time_seconds	gauge	Время запуска процесса с начала эпохи Unix в секундах
process_virtual_memory_bytes	gauge	Размер виртуальной памяти в байтах
process_virtual_memory_max_bytes	gauge	Максимальный объем доступной виртуальной памяти в байтах
promhttp_metric_handler_requests_in_flight	counter	Количество обрабатываемых запросов в моменте на ручке /metrics
promhttp_metric_handler_requests_total	counter	Котичество запросов на ручку /metrics
server_grpc_connections	gauge	Количество подключенных на данный момент агентов
server_grpc_response_time_sum	histogram	Суммарная гистограмма времени ответа по всем методам

Metric Name	Metric Type	Description
server_grpc_response_time_count	histogram	Гистограмма времени ответа по каждому методу
server_grpc_response_time_bucket	histogram	Гистограмма времени ответа
server_grpc_methods_started	counter	Количество заходов в метод
server_grpc_methods_finished	counter	Количество выходов из метод
server_grpc_messages	counter	Количество полученных и отправленных сообщений
nftables_up	counter	Удачен ли сбор данных для метрик (1 или 0)
nftables_counter_bytes	counter	Количество байтов на табличном каунтере
nftables_counter_packets	counter	Количество пакетов на табличном каунтере
nftables_table_chains	counter	Количество цепочек в таблице
nftables_chain_rules	counter	Количество количество правил в цепочке
nftables_rule_bytes	counter	Количество байтов у правила
nftables_rule_packets	counter	Количество пакетов у правила

Для подключения или отключения сбора метрик необходимо настроить следующие поля metrics.enable, healthcheck.enable и profile.enable установив значение true - чтобы включить или false - чтобы выключить (по умолчанию true).

```
telemetry:
  useragent: "string"
  nft-collector:
    min-frequency: 1s
  endpoint: 127.0.0.1:5000
  metrics:
    enable: true
  healthcheck:
    enable: true
  profile:
    enable: true
```

Настройка TLS * ЛУЧШЕ ГРУППИРОВАТЬ

МОНИТОРИНГ, ТЛС, НАСТРОЙКА, ЛУЧШЕ ГРУППИРОВАТЬ ЧТО БЫ ПОЛУЧИЛОСЬ

- ПРИМЕР: НАСТРОЙКА TLS
- – АГЕНТ
- - СЕРВЕР
- – ТЕРРАФОРМ

Установка

Настройка TLS (Transport Layer Security) на hbf-сервере обеспечивает шифрование трафика между сервером и клиентом, что повышает безопасность передаваемых данных. В этой документации описан процесс настройки TLS на hbf-сервере, включая использование предоставленного конфигурационного файла.

Прежде чем приступить к настройке TLS, убедитесь, что у вас есть:

Установленный hbf-сервер
Сертификат SSL и соответствующий приватный ключ. Если у вас их нет, вы можете получить их у сертификационного центра (CA) или создать самоподписанный сертификат для тестовых целей.
Шаги по настройке TLS
Создайте файл конфигурации hbf-сервера для редактирования:

sudo nano /etc/cmd/to-nft/internal/tls-config.yaml

Далее необходимо настроить секцию для TLS:

Insecure TLS

```
authn:
  type: tls
  tls:
    key-file: "/etc/ssl/private/key-file.pem"
    cert-file: "/etc/ssl/certs/cert-file.pem"
    client:
      verify: skip
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls

tls включен.

key-file — Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: skip, cert-required или verify. При значении skip сертификат клиента не проверяется.

Secure TLS

```
authn:
  type: tls
  tls:
    key-file: "/etc/ssl/private/key-file.pem"
    cert-file: "/etc/ssl/certs/cert-file.pem"
    client:
      verify: cert-required
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls tls включен.

key-file — Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: skip, cert-required или verify. При значении cert-required от клиента требуется наличие сертификатов, но со стороны сервена данные сертификаты не проверяются.

mTLS

```
authn:
  type: tls
  tls:
    key-file: "/etc/ssl/private/key-file.pem"
    cert-file: "/etc/ssl/certs/cert-file.pem"
    client:
      verify: verify
      ca-files: ["file1.pem", "file2.pem", ...]
```

type — Допустимые значени: none или tls. При значении none tls отключен, при значении tls

tls включен.

key-file — Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

verify — Допустимые значени: skip, cert-required или verify. При значени verify включается режим mTLS, когда сертификат клиента необходим и происходит его проверка.

ca-files — При включенном режиме проверки сертификата сервера verify: verify необходимо перечислить список certificates authorities с указанием относительного или полного пути к файлам.

Описание базы данных

Ниже приводится схема и описание таблиц базы данных, созданных для стандартного использования HBF-Server.

HBF-Server поддерживает PostgreSQL версии 14.8

Поскольку HBF-Server взаимодействует с этой базой данных самостоятельно, конечному пользователю не нужно беспокоиться о ее структуре и о том как хранятся данные.

Схема базы данных

На диаграмме ниже представлен визуальный обзор базы данных HBF-Server и связей между таблицами. В приведенном ниже Обзоре Таблиц, содержатся дополнительные сведения о таблицах и столбцах базы данных.

Обзор таблиц (сущности)

В этом разделе представлен обзор всех таблиц, созданных для стандартного использования HBF-Server. С последующим детальным описанием, что находится в каждой таблице.

Название таблицы	Описание	Соответствующие области взаимодействия интерфейса (API)
tbl_network	таблица tbl_network хранит информацию о IP Subnets с уникальным названием, CIDR и ссылкой на SG к сети которой она принадлежит	• Отобразить список доступных сетей (Networks) • Отобразить список доступных сетей

Название таблицы	Описание	Соответствующие области взаимодействия интерфейса (API)
		(Networks) связанных с SG • Отобразить SG по IP или CIDR • Внести изменения в БД
tbl_sg	таблица tbl_sg хранит информацию о Security Groups (SG) с уникальным названием, правилом применяемым для входящих или исходящих пакетов, также возможностью включить логирование	• Отобразить список Security Groups (SG) • Внести изменения в БД
tbl_ie_sg_sg_rule	таблица tbl_ie_sg_sg_rule хранит информацию SG-SG правил для входящего и исходящего траффика с сетевым транспортным протоколами и диапазоном портов	• Отобразить список IE-SG-SG правил для входящего и исходящего траффика • Внести изменения в БД
tbl_ie_sg_sg_icmp_rule	таблица tbl_ie_sg_sg_icmp_rule хранит информацию SG-SG правил для входящего и исходящего траффика через сетевой протокол ICMP	Отобразить список IE-SG-SG-ICMP правил для входящего и исходящего траффика Внести изменения в БД
tbl_cidr_sg_rule	таблица tbl_cidr_sg_rule хранит информацию CIDR-SG правил для входящего и исходящего траффика с сетевым транспортным протоколом, бесклассовой междоменной маршрутизацией (CIDR) и диапазоном портов	• Отобразить список CIRD-SG правил для входящего и исходящего траффика • Внести изменения в БД
tbl_cidr_sg_icmp_rule	таблица tbl_cidr_sg_icmp_rule хранит информацию CIDR-SG правил для входящего и исходящего траффика с сетевым протоколом ICMP, бесклассовой междоменной маршрутизацией (CIDR)	• Отобразить список CIRD-SG правил для входящего и исходящего траффика • Внести изменения в БД
tbl_fqdn_rule	таблица tbl_fqdn_rule хранит информацию SG-to-FQDN правил с сетевым транспортным протоколом и диапазоном портов	• Отобразить список полных доменных имен (FQDN) • Внести изменения в БД
tbl_sg_icmp_rule	таблица tbl_sg_icmp_rule хранит информацию SG:ICMP правил	• Отобразить список правил SG:ICMP ограниченных по типу SG • Внести изменения в БД

Название таблицы	Описание	Соответствующие области взаимодействия интерфейса (API)
tbl_sg_rule	таблица tbl_sg_rules хранит информацию о правилах виртуального файрволла который можно настраивать для того чтобы контролировать входящий и выходящий трафик	• Отобразить список SG правил ограниченных по условиям from -> to • Внести изменения в БД
tbl_sg_sg_icmp_rule	таблица tbl_sg_sg_icmp_rule хранит информацию SG-SG:ICMP правил	• Отобразить список правил SG-SG:ICMP ограниченных по типу SG from -> to • Внести изменения в БД
tbl_sync_status	в таблице tbl_sync_status хранится информация об изменениях внесенных пользователем (дата последнего успешного изменения и кол-во изменённых строк)	• Отобразить статус последнего успешного обновления БД

Подробное описание таблиц

Ниже приведены конкретные поля в каждой из таблиц, созданных для стандартного использования HBF-Server

tbl_network

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
sg	int(8)	YES	FK		внешний ключ к таблице tbl_sg.id
name	cname		ALT		• длина значения не должна превышать 256 символов • значения должно начинаться и заканчиваться символами без пробелов • значение должно быть уникальным
network	cidr				• значение от 7 до 19 байт пример `192.168.100.128/25` • сетевые интервалы не должны пересекаться

Ключи

Имя ключа	Тип	Поля
Alternative key	Simple Key	cname

tbl_sg

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
name	cname		ALT		• длина значения не должна превышать 256 символов • значения должно начинаться и заканчиватьс я символами без пробелов • значение должно быть уникальным
logs	bool			false	
trace	bool			false	
default_action	chain_default_action			‘DROP’::chain_default_action	одно из двух значений: • ACCEPT • DROP

Ключи

Имя ключа	Тип	Поля
Alternative key	Simple Key	cname

tbl_ie_sg_sg_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
proto	proto		ALT		одно из двух значений: • tcp • udp
sg	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
sg_local	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
traffic	traffic		ALT		одно из двух значений: • ingress

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
					• egress
ports	sg_rule_ports[]	YES			• должно быть указано значение порта исходящего либо входящего трафика • значение должно находиться в интервале от 1 до 65535 • интервалы введённых значений портов для исходящего трафика не должны пересекаться
logs	bool				
trace	bool				
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• proto • sg • sg_local • traffic

tbl_ie_sg_sg_icmp_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
sg	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
sg_local	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
traffic	traffic		ALT		одно из двух значений "ingress" или "egress"
ip_v	ip_family		ALT		одно из двух значений "IPv6" или "IPv4"
types	icmp_types				массив из smallint[] кодов типа ICMP

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
logs	Boolean			false	
trace	Boolean			false	
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• sg • sg_local • traffic • ip_v

tbl_cidr_sg_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
proto	proto		ALT		одно из двух значений: • tcp • udp
cidr	cidr		ALT		значение cidr (диапазон ip адресов) в рамках одного правила (proto, sg, traffic) не должны пересекаться
sg	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
traffic	traffic		ALT		• ingress • egress
ports	sg_rule_ports[]	YES			• должно быть указано значение порта исходящего либо входящего трафика • значение должно находиться в интервале от 1 до 65535

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
					• интервалы введённых значений портов для исходящего трафика не должны пересекаться
logs	bool				
trace	bool				
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• proto • cidr • sg • traffic

tbl_cidr_sg_icmp_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
cidr	cidr		ALT		значение cidr (диапазон ip адресов) в рамках одного правила (ip_v, sg, traffic) не должны пересекаться
sg	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
traffic	traffic		ALT		• ingress • egress
ip_v	ip_family		ALT		одно из двух значений: • IPv6 • IPv4
types	icmp_types				массив из smallint[] кодов типа ICMP

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
logs	bool				
trace	bool				
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• ip_v • cidr • sg • traffic

tbl_fqdn_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
sg_from	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
fqdn_to	fqdn		ALT		• длина значения не должна превышать 256 символов • значение начинается со строки, которая содержит один или более символов, являющихся буквами нижнего регистра, цифрами, символом . или символами _ и - (кроме первого символа, который не может быть - или _), и должна быть длиной от 1 до 62 символов • затем может следовать любое количество строк, начинающихся с символа ., за которым идет один символ, являющийся буквой нижнего регистра, цифрой,

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
					символом _ или символом -, и длина строки от 0 до 62 символов
proto	proto		ALT		одно из двух значений: • tcp • udp
ports	sg_rule_ports[]	YES			• должно быть указано значение порта исходящего либо входящего трафика • значение должно находиться в интервале от 1 до 65535 • интервалы введённых значений портов для исходящего трафика не должны пересекаться
logs	bool			false	
ndpi_protocols	citext				• количество элементов в массиве (наименований протоколов) не должно превышать 255 • значение элемента (наименование протокола) не должно начинаться или заканчиваться пробелом и не должно быть пустым
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• sg_from • fqdn_to

Имя ключа	Тип	Поля
		• proto

tbl_sg_icmp_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
ip_v	ip_family		ALT		одно из двух значений: • IPv6 • IPv4
types	icmp_types				массив из smallint[] кодов типа ICMP
sg	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
logs	bool				
trace	bool				
action	rule_action				одно из двух значений: • ACCEPT • DROP

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• ip_v • sg

tbl_sg_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
sg_from	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
sg_to	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
proto	proto		ALT		одно из двух значений: • tcp • udp
ports	sg_rule_ports[]	YES			• должно быть указано значение порта исходящего либо входящего трафика

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
					• значение должно находиться в интервале от 1 до 65535 • интервалы введённых значений портов для исходящего трафика не должны пересекаться
logs	bool			false	
action	rule_action				одно из двух значений: • ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• sg_from • sg_to • proto

tbl_sg_sg_icmp_rule

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
ip_v	ip_family		ALT		одно из двух значений: • IPv6 • IPv4
types	icmp_types				массив из smallint[] кодов типа ICMP
sg_from	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
sg_to	int(8)		FK/ALT		внешний ключ к таблице tbl_sg.id
logs	bool				
trace	bool				
action	rule_action				одно из двух значений:

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
					• ACCEPT • DROP
priority	smallint			0	• значения должны находиться в интервале от -32768 до 32767 • чем ниже значение приоритета, тем выше приоритет имеет правило • необязательное поле для заполнения, если значение не указать будет использоваться значение по умолчанию

Ключи

Имя ключа	Тип	Поля
Alternative key	Compound Key	• ip_v • sg_from • sg_to

tbl_sync_status

Поле	Тип	Null	Ключ	По умолчанию	Дополнительно
id	int(8)		PRI		auto_increment
total_affected_rows	int(8)				при любой процедуре (удаление/добавление/редактирование) данных в таблицах, tbl_network, tbl_sg, tbl_fqdn_rule, tbl_sg_rule, tbl_sg_icmp_rule, tbl_sg_sg_icmp_rule, будет учтена сумма всех изменённых строк
updated_at	timstamptz	YES			дата изменения

API

POST /v1/sync

Networks

Ресурс Networks представляет собой введенную нами абстракцию, которая позволяет определять группы IP-адресов или подсетей, доступных для управления Host Based NGFW. Эти подсети затем могут быть связаны с конкретными группами безопасности для логического разделения и использоваться в правилах для разрешения или блокирования доступа к определенным ресурсам в вашей сети.

Входные параметры

- networks[] — Массив/Список подсетей типа IP.
- networks[].name — название подсети.
- networks[].network — объект содержащий CIDR подсети
- networks[].network.CIDR — Подсеть типа IP.
- syncOp — Поле определяющее действие с данными из запроса.

название	обязательность	тип данных	значение по умолчанию
networks[]	да	Object[]	
networks[].name	да	String	
networks[].network	да	Object	
networks[].network.CIDR	да	String	
syncOp	да	Enum("Delete", "Upsert", "FullSync")	

Ограничения

networks.networks[].name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

networks.networks[].network.CIDR:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

Пример использования

```
curl '127.0.0.1:9007/v1/sync' \
--header 'Content-Type: application/json' \
--data '{
   "networks": {
      "networks": [{
         "name": "nw-1",
         "network": {
```

```
            "CIDR": "10.0.0.0/24"
          }
        }]
    },
    "syncOp": "Upsert"
}'
```

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5

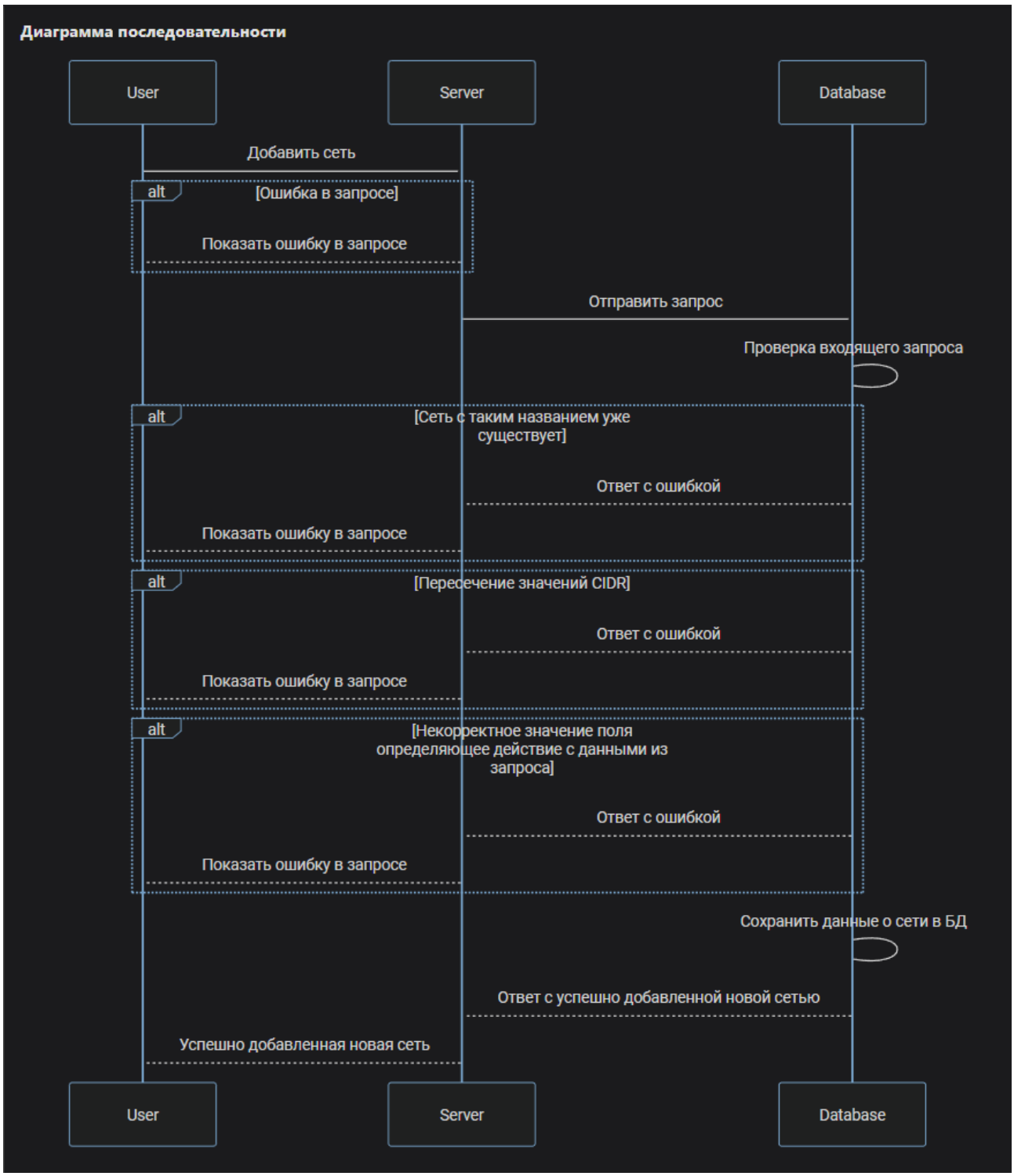


Диаграмма последовательности
User
Server
Database
Добавить сеть
alt
[Ошибка в запросе]
Показать ошибку в запросе
Отправить запрос
Проверка входящего запроса
alt
[Сеть с таким названием уже существует]
Ответ с ошибкой
Показать ошибку в запросе
alt
[Пересечение значений CIDR]
Ответ с ошибкой
Показать ошибку в запросе
alt
[Некорректное значение поля определяющее действие с данными из запроса]
Ответ с ошибкой
Показать ошибку в запросе
Сохранить данные о сети в БД
Ответ с успешно добавленной новой сетью
Успешно добавленная новая сеть
User
Server
Database

POST /v2/list-security-groups

Этот метод отображает список сетей (networks) и действия по умолчанию, в соответствии с указанным списком имен Security Groups.

Входные параметры

sgNames[] - Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
sgNames[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/list-security-groups' \
--header 'Content-Type: application/json' \
--data '{
    "sgNames": ["sg-example"]
}'
```

Выходные параметры

- $node.groups[] — Структура, содержащая описание создаваемых правил.
- $node.groups[].name - Security Group, с которой устанавливаются правила взаимодействия.
- $node.groups[].logs — Включить/отключить логирование.
- $node.groups[].trace — Включить/отключить трассировку.
- $node.groups[].networks — Массив/Список подсетей типа IP.
- $node.groups[].defaultAction — Структура, содержащая описание создаваемых правил типа ICMP.

название	**тип данных**
$node.groups[]	Object[]
$node.groups[].name	String
$node.groups[].logs	Boolean
$node.groups[].trace	Boolean
$node.groups[].networks	Object[]
$node.groups[].defaultAction	String

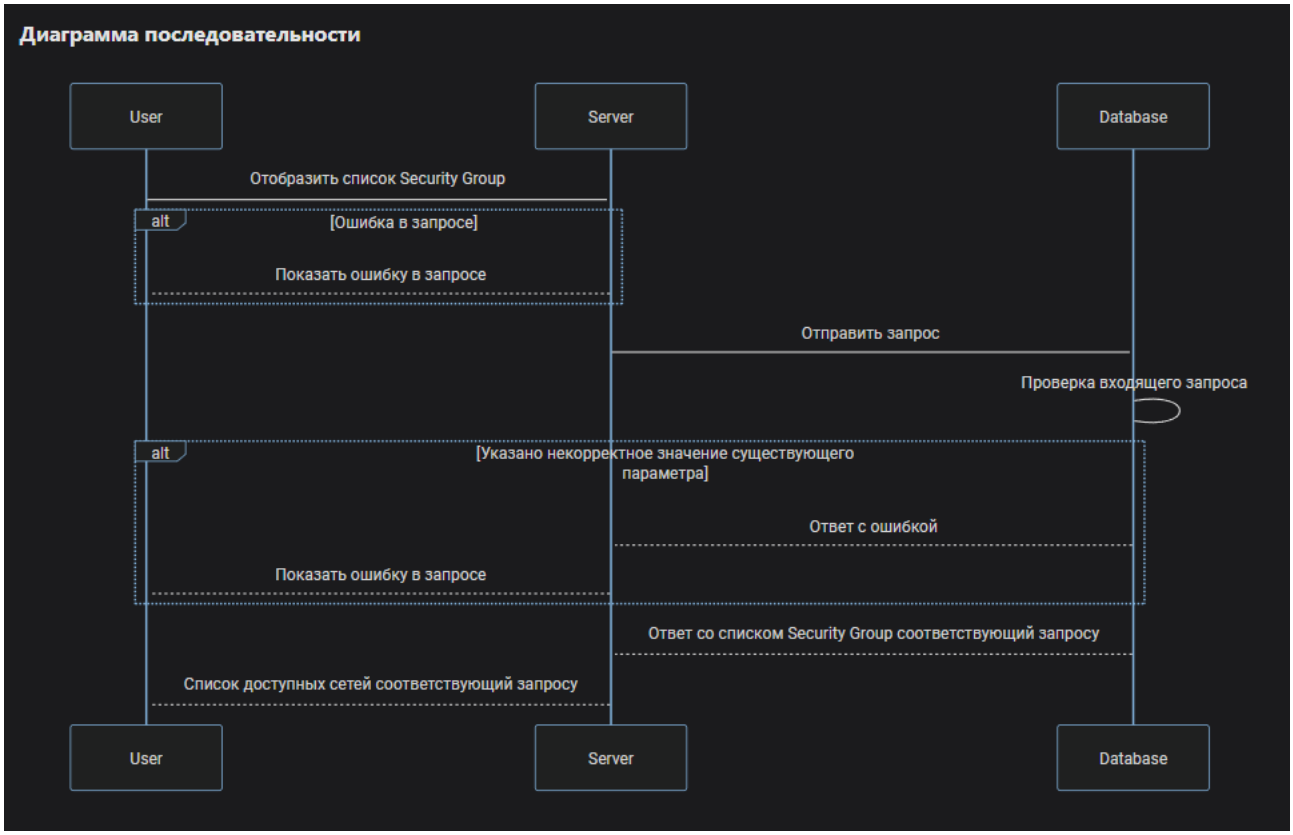
Пример ответа


```
{
   "groups": [{
      "name": "sg-example",
      "logs": "true",
      "trace": "true",
      "networks": ["network-example"],
      "defaultAction": "DROP"
   }]
}
```


Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



GET /v2/{address}/sg

Этот метод отображает Security Group по IP или CIDR входящей в нее подсети.

Входные параметры

- {address} - Подсеть типа IP.

название	обязательность	тип данных	Значение по умолчанию
{address}	да	String	

Пример использования

curl '127.0.0.1:9007/v2/10.150.0.224/sg' \
--header 'Content-Type: application/json'

Выходные параметры

- $node.name — Security Group, с которой устанавливаются правила взаимодействия.
- $node.logs — Включить/отключить логирование.
- $node.trace — Включить/отключить трассировку.
- $node.networks — Массив/Список имен подсетей
- $node.defaultAction — Действие по умолчанию.

название	тип данных
$node.name	String
$node.logs	Boolean
$node.trace	Boolean
$node.networks	Object[]
$node.defaultAction	String

Пример ответа

```
{
   "name": "sg-example",
   "logs": false,
   "trace": false,
   "network": ["network-example"],
   "defaultAction": "DROP"
}
```

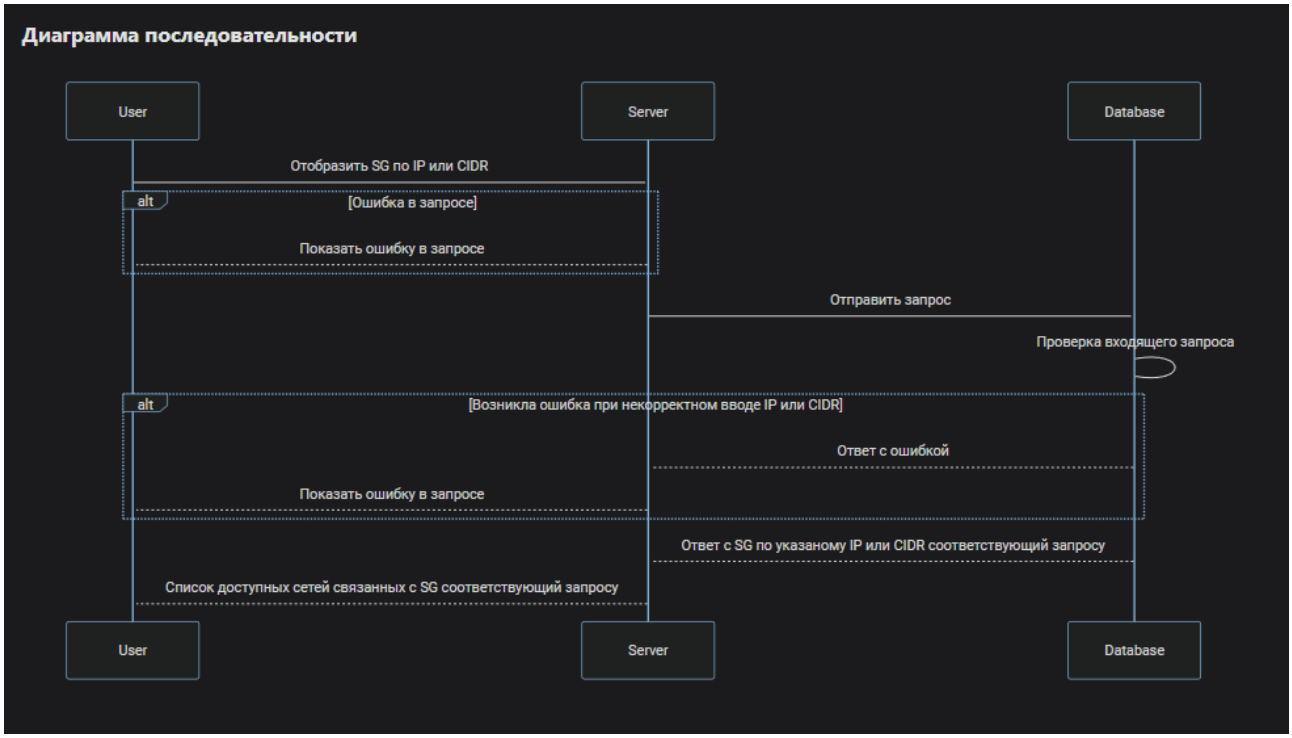
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/list-networks

Этот метод отображает список подсетей (networks) и их IP, в соответствии с указанным списком имен подсетей (networks).

Входные параметры

- networkNames[] — Массив/Список имен подсетей

Входные параметры

- `networkNames[]` — Массив/Список имен подсетей

название	обязательность	тип данных	Значение по умолчанию
networkNames[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/list-networks' \
--header 'Content-Type: application/json' \
--data '{
   "networkNames": ["network-example"]
}'
```

Выходные параметры

- $node.networks[] — Структура, содержащая описание создаваемых правил.
- $node.networks[].name — Имя подсети
- $node.networks[].network — Структура, содержащая описание сети
- $node.networks[].network.CIDR — Массив/Список подсетей типа IP.

название	тип данных
$node.networks[]	Object[]
$node.networks[].name	String
$node.networks[].network	Object
$node.networks[].network.CIDR	String

Пример ответа


```
{
   "networks": [{
      "network": "network-example",
      "ICMP": {
            "CIDR": "10.150.0.220/32"
         },
   }]
}
```

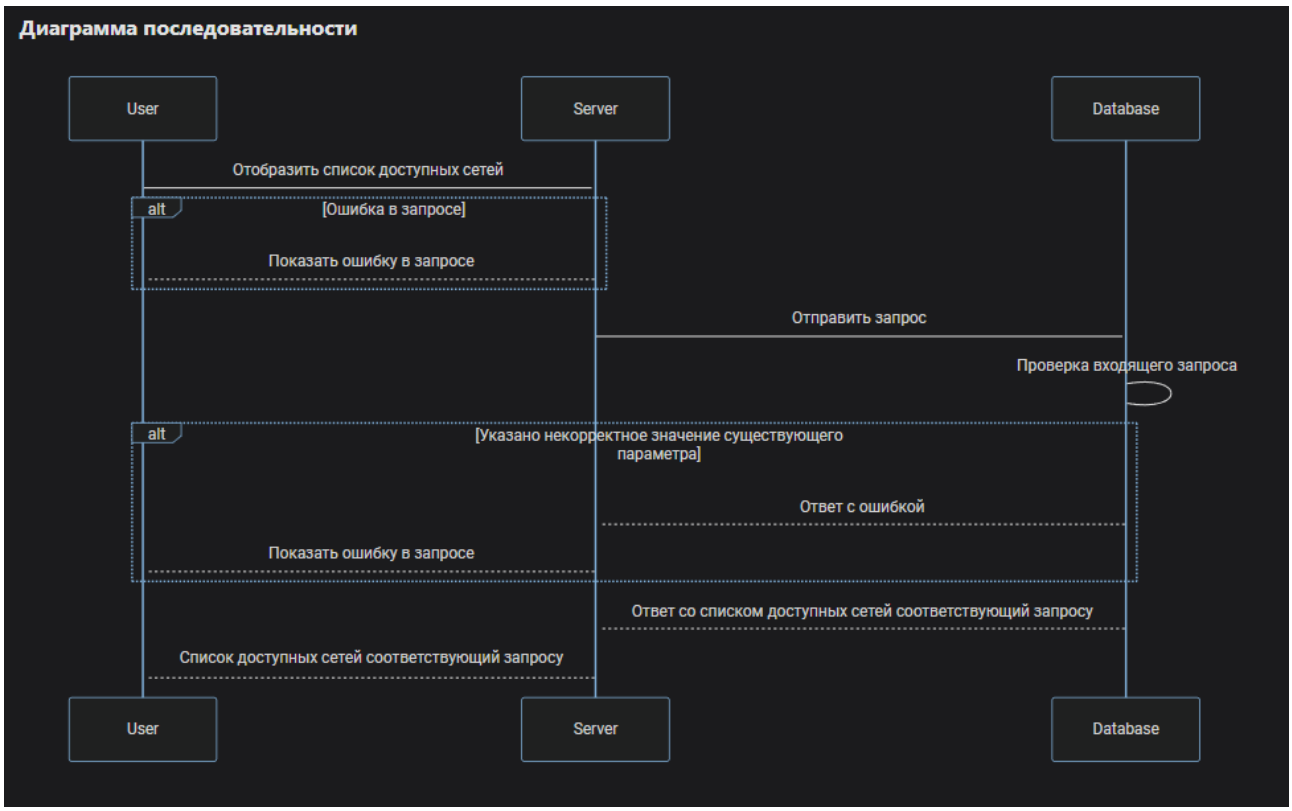

Возможные ошибки API

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



GET /v2/sg/{sgName}/subnets

Этот метод отображает список подсетей по указанной Security Group.

Входные параметры

- {sgName} — Security Group, с которой устанавливаются правила взаимодействия.

название	обязательность	тип данных	Значение по умолчанию
{sgName}	да	String	

Пример использования

```
curl '127.0.0.1:9007/v2/sg/sg-example/subnets' \
--header 'Content-Type: application/json'
```

Выходные параметры

- $node.networks[] — Массив/Список имен подсетей
- $node.networks[].name — Имя подсети
- $node.networks[].network — Структура, содержащая описание сети
- $node.networks[].network.CIDR — Подсеть типа IP.

название	тип данных
$node.networks[]	Object[]
$node.networks[].name	String
$node.networks[].network	Object
$node.networks[].network.CIDR	String

Пример ответа

```
{
   "networks": [{
      "name": "network-example",
      "network": {
            "CIDR": "10.150.0.222/32"
         },
   }]
}
```

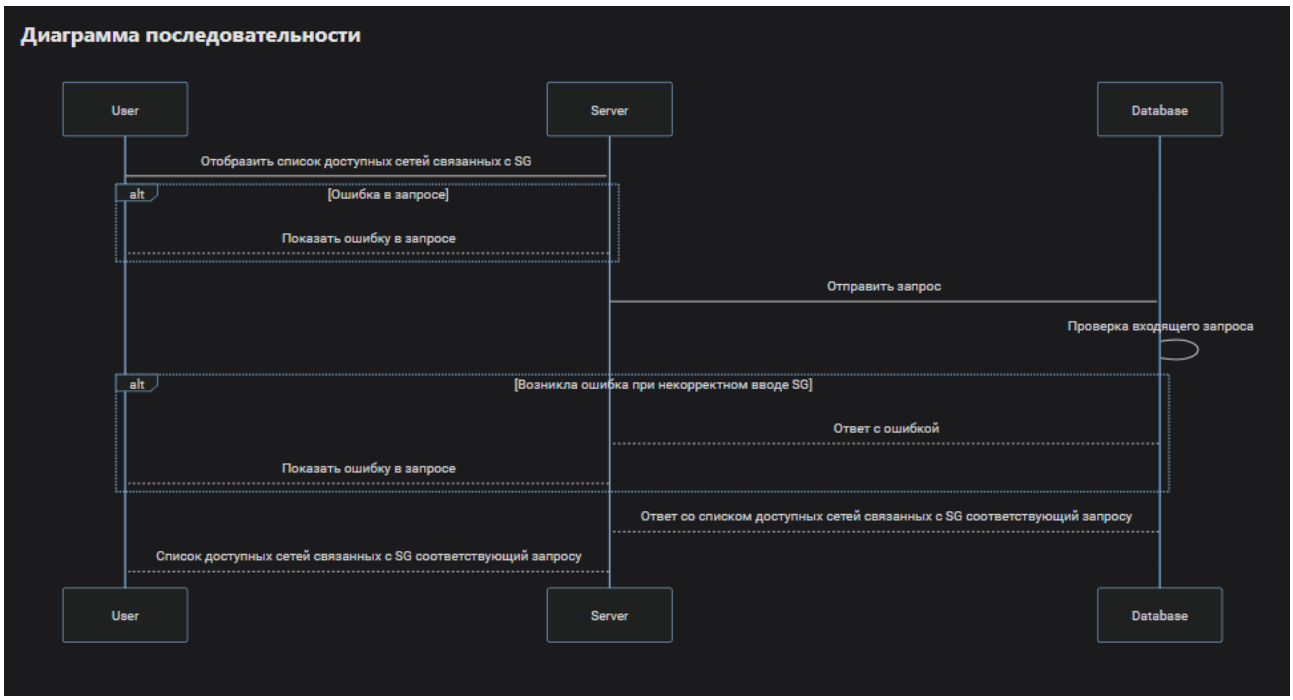
Возможные ошибки API

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/sg-sg-icmp-rules

Этот метод отображает список Security Group to Security Group правил, в соответствии с указанным списком Security Groups.

Входные параметры

- sgFrom[] — Список, содержащий названия Security Group(s).
- sgTo[] — Список, содержащий названия Security Group(s).

название	**обязательность**	**тип данных**	**Значение по умолчанию**
sgFrom[]	да	Object[]	
sgTo[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/sg-sg-icmp-rules' \
--header 'Content-Type: application/json' \
--data '{
    "sgFrom": ["sg-example"],
    "sgTo": ["sg-example-2"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].sgFrom — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].sgTo — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types[] — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].sgFrom	String
$node.rules[].sgTo	String
$node.rules[].logs	Boolean
$node.rules[].trace	Boolean
$node.rules[].ICMP	Object
$node.rules[].ICMP.IPv	String
$node.rules[].ICMP.Types[]	Object[]
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

Пример ответа

```
{
   "rules": [{
      "sgFrom": "sg-example",
      "sgTo": "sg-example-2",
      "logs": "true",
      "trace": "true",
      "ICMP": {
            "IPv": "IPv4",
            "Types": [0, 8]
         },
      "action": "ACCEPT",
      "priority": {
            "some": -300
         }
   }]
}
```

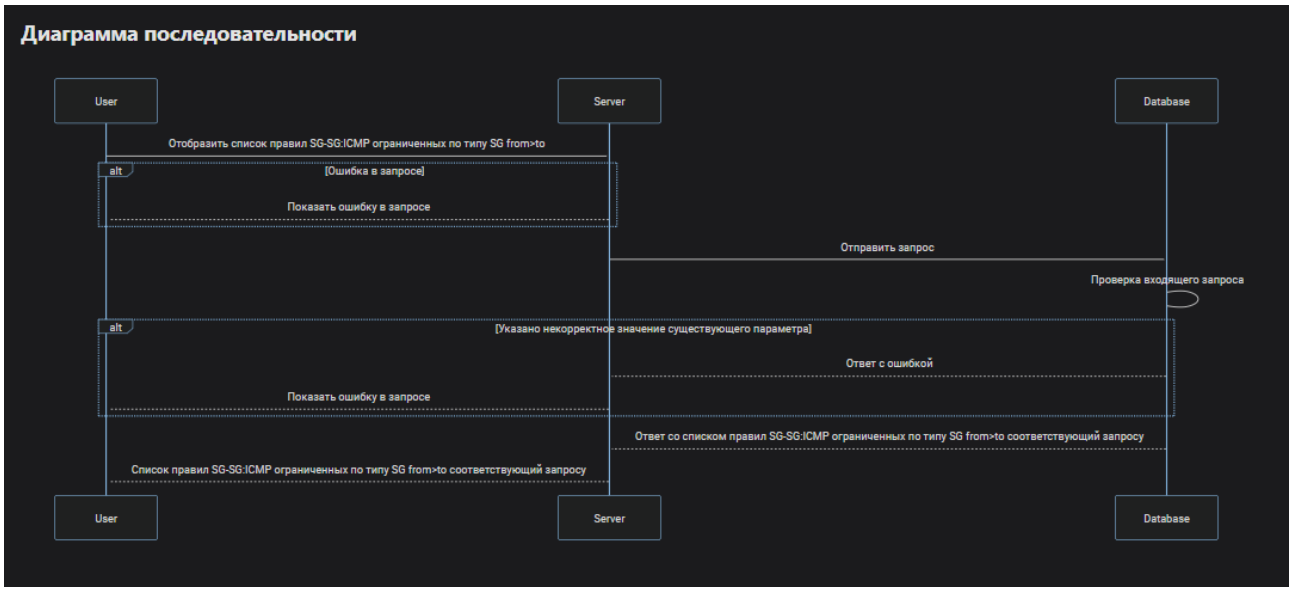

Возможные ошибки API

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/sg-icmp-rules

Этот метод отображает список Security Group правил, в соответствии с указанным списком Security Groups.

Входные параметры

- SG[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
SG[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/sg-icmp-rules' \
--header 'Content-Type: application/json' \
--data '{
    "SG": ["sg-example"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types[] — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].SG	String
$node.rules[].logs	Boolean
$node.rules[].trace	Boolean
$node.rules[].ICMP	Object

название	тип данных
$node.rules[].ICMP.IPv	String
$node.rules[].ICMP.Types[]	Object[]
$node.rules[].action	String

Пример ответа


```
{
   "rules": [{
      "SG": "sg-example",
      "logs": "true",
      "trace": "true",
      "ICMP": {
            "IPv": "IPv4",
            "Types": [0, 8]
         },
      "action": "ACCEPT",
   }]
}
```

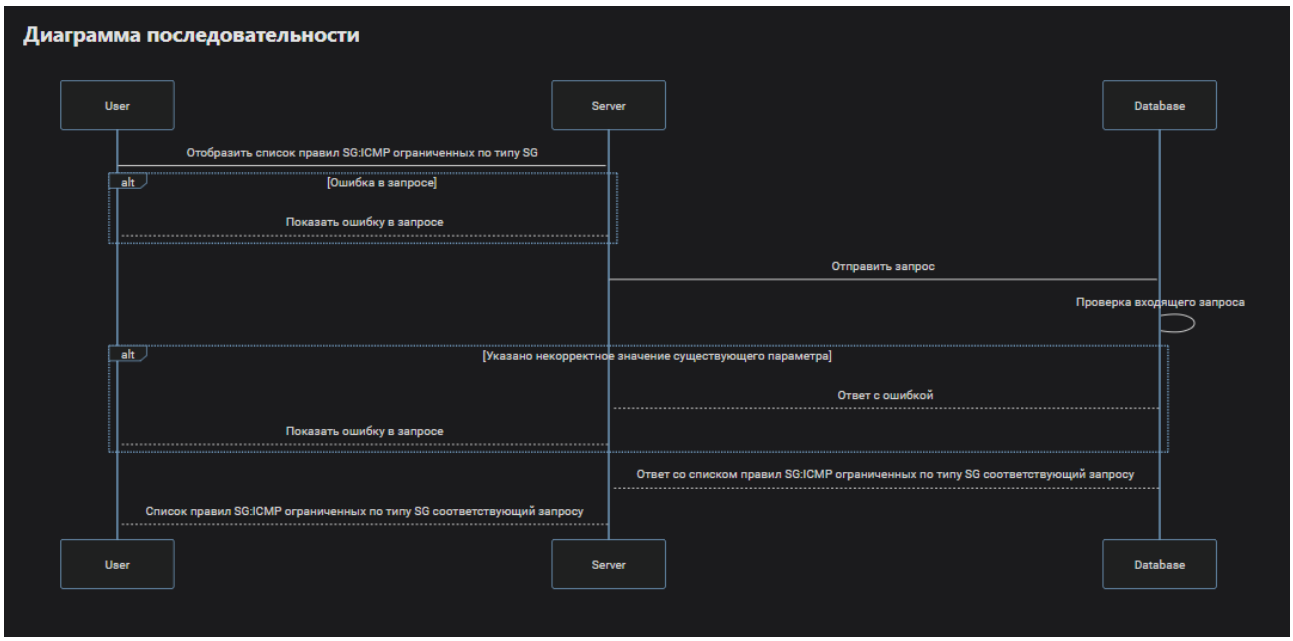

Возможные ошибки API

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/sg-sg-rules

Этот метод отображает список Security Group to Security Group правил, в соответствии с указанным списком Security Groups.

Входные параметры

- sgFrom[] — Список, содержащий названия Security Group(s).
- sgTo[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
sgFrom[]	да	Object[]	
sgTo[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/sg-sg-rules' \
--header 'Content-Type: application/json' \
--data '{
    "sgFrom": ["sg-example"],
    "sgTo": ["sg-example-2"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].sgFrom — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].sgTo — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].sgFrom	String
$node.rules[].sgTo	String
$node.rules[].logs	Boolean
$node.rules[].ports	Object[]
$node.rules[].ports.d	String
$node.rules[].ports.s	String
$node.rules[].transport	String
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

Пример ответа


```
{
   "rules": [{
      "sgFrom": "sg-example",
      "sgTo": "sg-example-2",
      "logs": "true",
      "transport": "TCP",
      "ports": [{
            "d": "5000",
            "s": ""
         }],
      "action": "ACCEPT",
      "priority": {
            "some": -200
         }
   }]
}
```

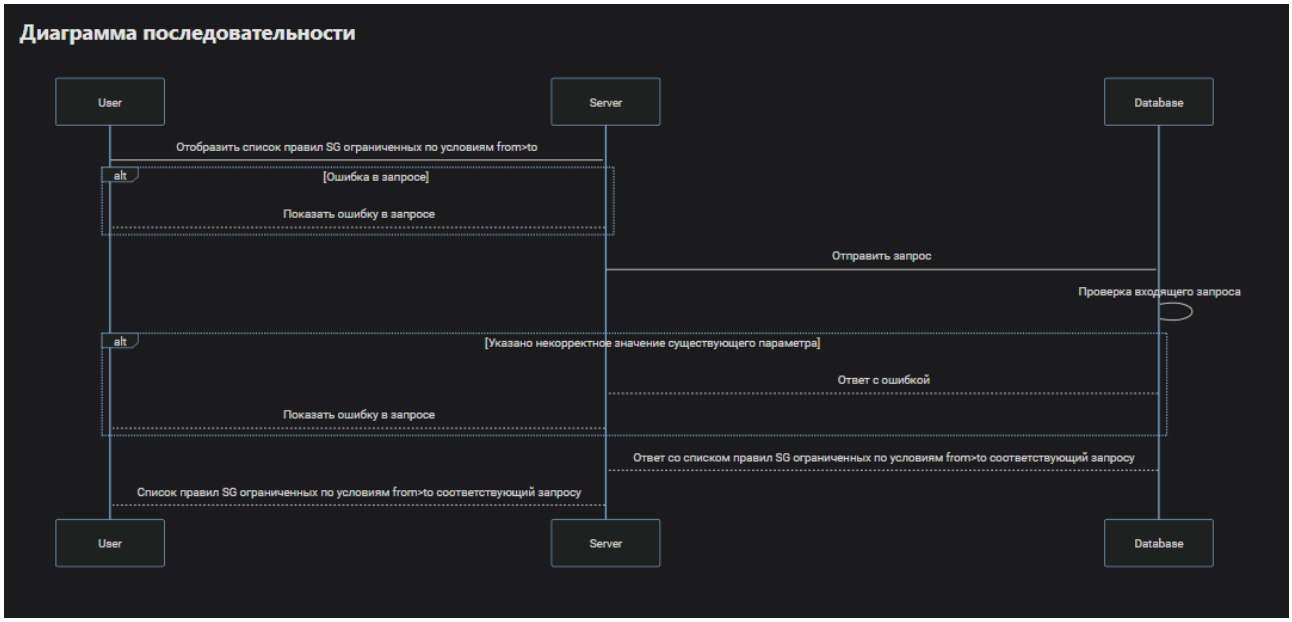
Возможные ошибки API

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/fqdn-rules

Этот метод отображает список Security Group to FQDN правил, в соответствии с указанным списком Security Groups.

Входные параметры

- sgFrom[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию

sgFrom[]	да	Object[]	

Ограничения

sgFrom[]:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

Пример использования

```
curl '127.0.0.1:9007/v2/fqdn-rules' \
--header 'Content-Type: application/json' \
--data '{
   "sgFrom": ["sg-example"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].FQDN — Полное доменное имя (FQDN), для которого применяется данное правило.
- $node.rules[].sgFrom — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].protocols — Список протоколов L7 уровня модели OSI.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].FQDN	String
$node.rules[].sgFrom	String
$node.rules[].logs	Boolean
$node.rules[].ports	Object[]
$node.rules[].ports[].d	String
$node.rules[].ports[].s	String
$node.rules[].protocols	Object[]
$node.rules[].transport	String
$node.rules[].action	String
$node.rules[].priority	Object

название	тип данных
$node.rules[].priority.some	Integer

Пример ответа

```
{
   "rules": [{
      "FQDN": "example.com",
      "sgFrom": "sg-example",
      "logs": "true",
      "ports": [{
            "d": "7800",
            "s": ""
         }],
      "protocols": ["ssh"],
      "transport": "TCP",
      "action": "ACCEPT",
      "priority": {
            "some": 100
         }
   }]
}
```

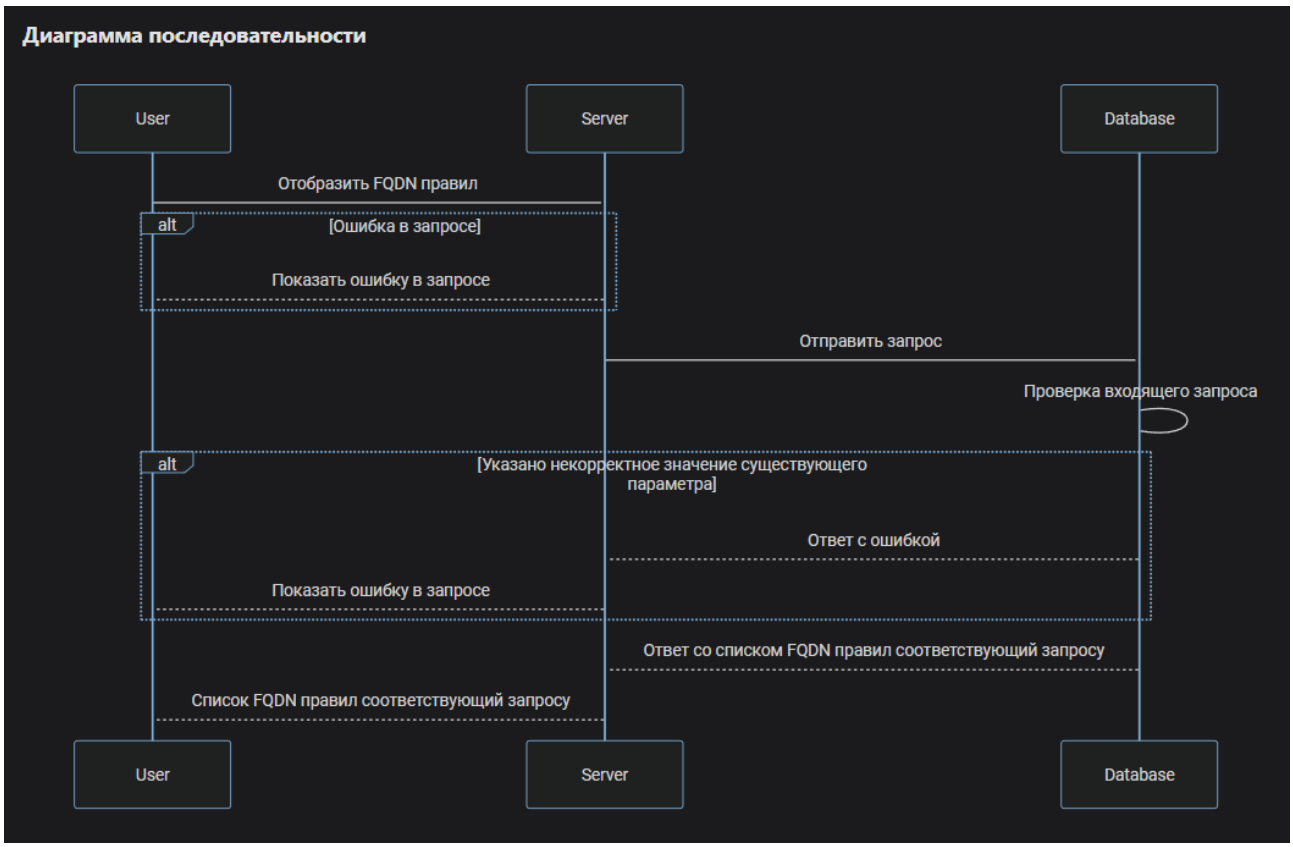
Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5

Ошибка в указанных данных

- HTTP code: 500
- gRPC code: INTERNAL
- gRPC number: 13



POST /v2/ie-cidr-sg-rules

Этот метод отображает список Security Group to CIDR правил, в соответствии с указанным списком Security Groups.

Входные параметры

- SG[] - Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
SG[]	да	Object[]	

Ограничения

SG[]:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

Пример использования

```
curl '127.0.0.1:9007/v2/ie-cidr-sg-rules' \
--header 'Content-Type: application/json' \
--data '{
   "SG": ["sg-example"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].CIDR — Подсеть типа IP.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].CIDR	String
$node.rules[].SG	String
$node.rules[].logs	Boolean
$node.rules[].ports	Object[]
$node.rules[].ports[].d	String
$node.rules[].ports[].s	String
$node.rules[].trace	Boolean
$node.rules[].traffic	String
$node.rules[].transport	String
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

Пример ответа

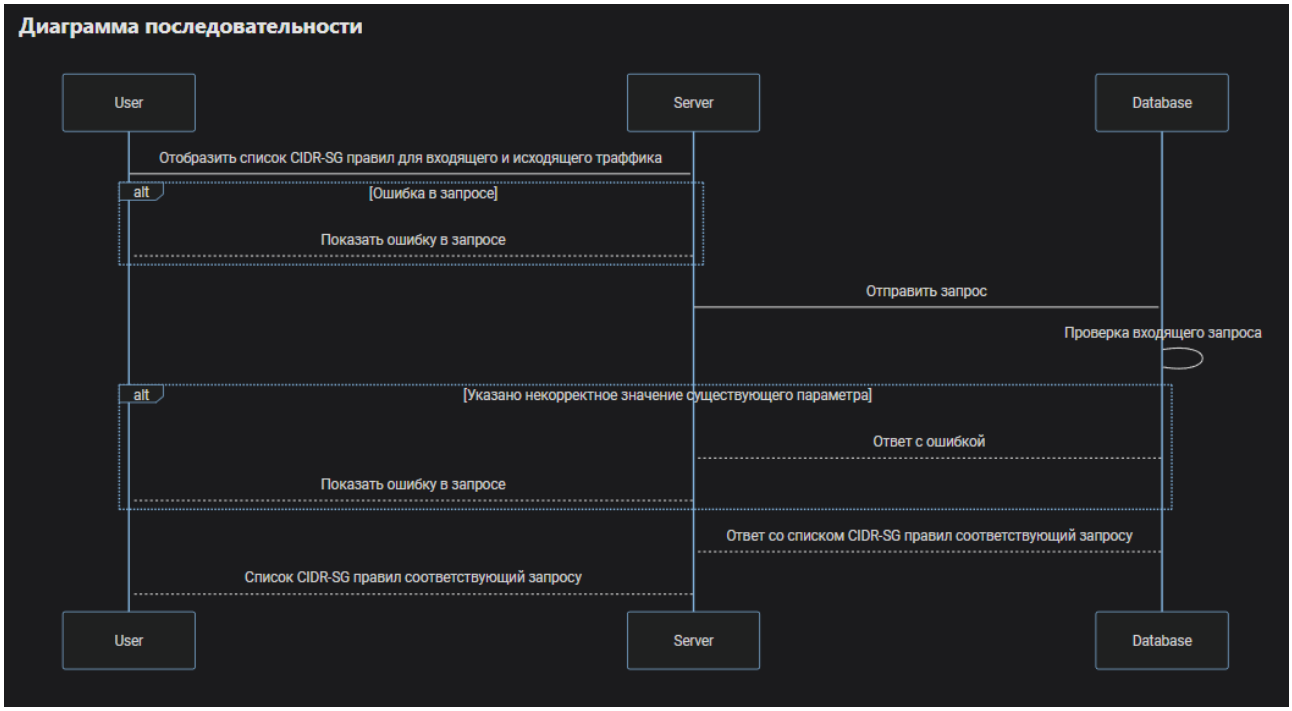

```
{
   "rules": [{
      "CIDR": "10.0.0.0/24",
      "SG": "sg-example",
      "logs": "true",
      "ports": [{
```

```
            "d": "7800",
            "s": ""
        }],
      "trace": "true",
      "traffic": "ingress",
      "transport": "TCP",
      "action": "ACCEPT",
      "priority": {
            "some": 300
        }
   }]
}
```


Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST /v2/ie-cidr-sg-icmp-rules

Этот метод отображает список Security Group to CIDR правил, в соответствии с указанным

списком Security Groups.

Входные параметры

- SG[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
SG[]	да	Object[]	

Ограничения

SG[]:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

Пример использования

```
curl '127.0.0.1:9007/v2/ie-cidr-sg-icmp-rules' \
--header 'Content-Type: application/json' \
--data '{
    "SG": ["sg-example"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].CIDR — Подсеть типа IP.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types[] — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].CIDR	String
$node.rules[].SG	String
$node.rules[].logs	Boolean
$node.rules[].ICMP	Object
$node.rules[].ICMP.IPv	String
$node.rules[].ICMP.Types[]	Object[]
$node.rules[].trace	Boolean
$node.rules[].traffic	String
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

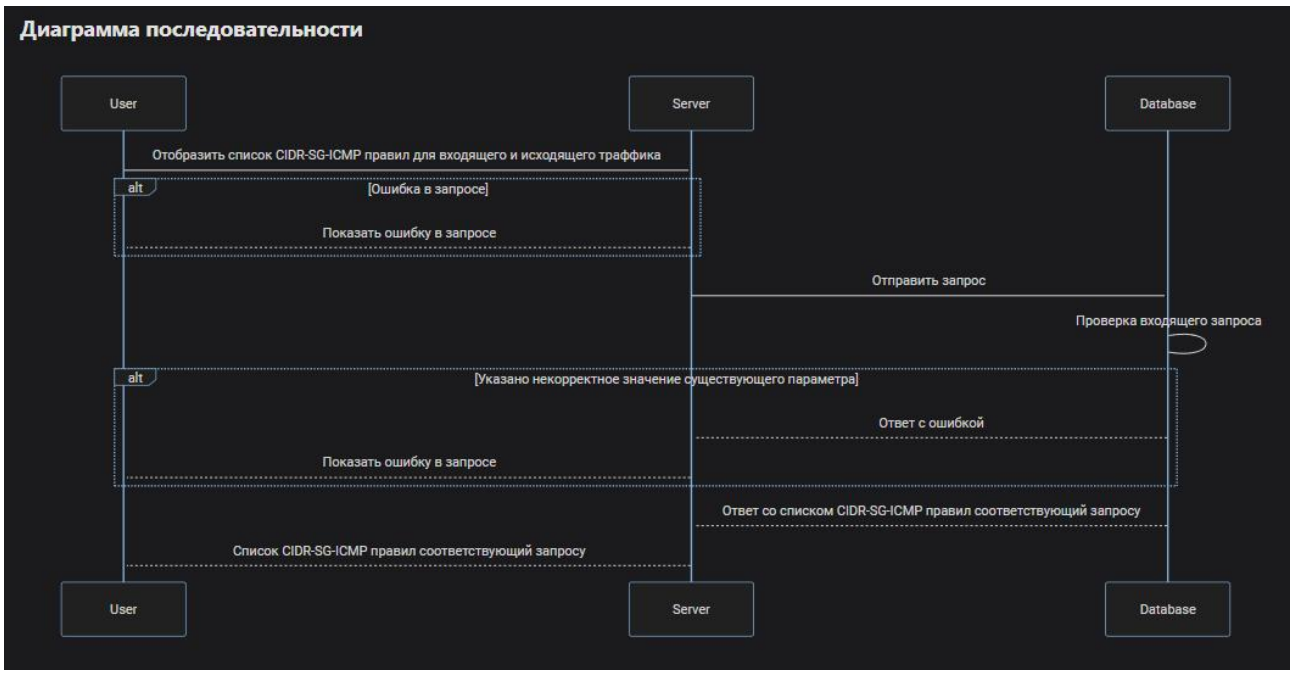
Пример ответа


```
{
   "rules": [{
      "CIDR": "10.0.0.0/24",
      "SG": "sg-example",
      "logs": "true",
      "ICMP": {
            "IPv": "IPv4",
            "Types": [0, 8]
         },
      "trace": "true",
      "traffic": "ingress",
      "action": "ACCEPT",
      "priority": {
            "some": 200
         }
   }]
}
```


Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST v2/ie-sg-sg-rules

Этот метод отображает список Security Group to Security Group правил, в соответствии с указанным списком Security Groups и типом траффика.

Входные параметры

- SG[] — Список, содержащий названия Security Group(s).
- sgLocal[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по умолчанию
SG[]	да	Object[]	
sgLocal[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/ie-sg-sg-rules' \
--header 'Content-Type: application/json' \
--data '{
   "SG": ["sg-example"],
   "sgLocal": ["sg-example-2"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.

- $node.rules[].sgLocal — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].SG	String
$node.rules[].sgLocal	String
$node.rules[].logs	Boolean
$node.rules[].trace	Boolean
$node.rules[].ports	Object[]
$node.rules[].ports[].s	String
$node.rules[].ports[].d	String[]
$node.rules[].traffic	String
$node.rules[].transport	String
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

Пример ответа

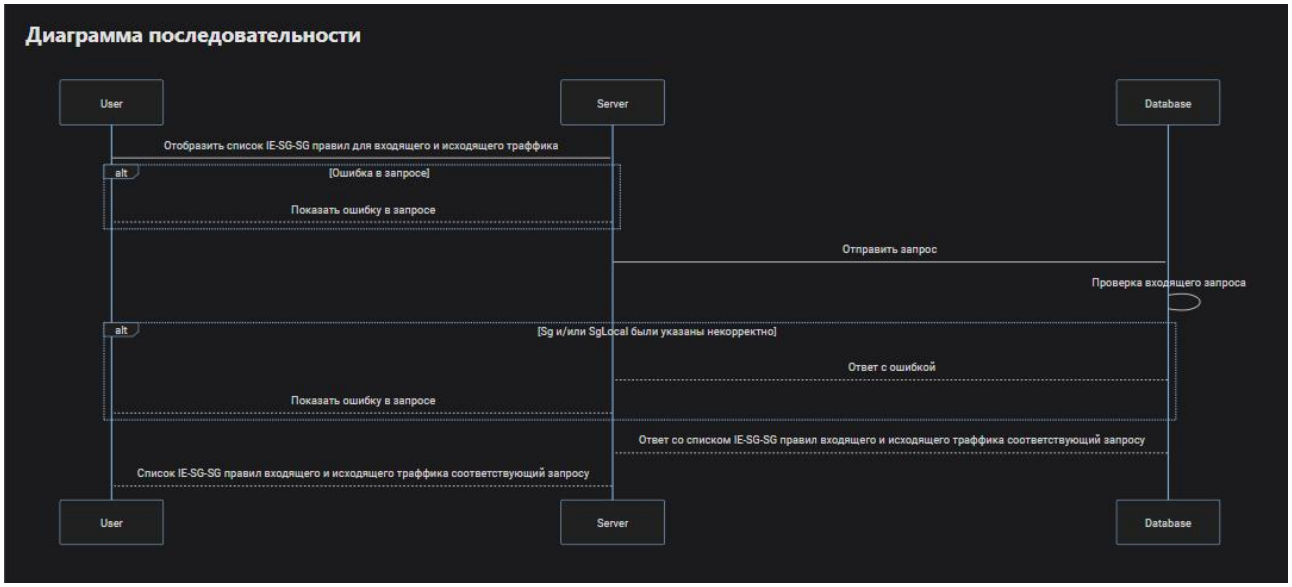

```
{
   "rules": [{
      "SG": "sg-example",
      "sgLocal": "sg-example-2",
      "logs": "true",
      "trace": "true",
      "ports": [{
            "d": "7800",
            "s": "4446"
         }],
      "traffic": "ingress",
      "transport": "TCP",
      "action": "ACCEPT",
      "priority": {
            "some": 0
```

```
        }
    }]
}
```

Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



POST v2/ie-sg-sg-icmp-rules

Этот метод отображает список Security Group to Security Group правил, в соответствии с указанным списком Security Groups и типом траффика.

Входные параметры

- SG[] — Список, содержащий названия Security Group(s).
- sgLocal[] — Список, содержащий названия Security Group(s).

название	обязательность	тип данных	Значение по

			умолчанию
SG[]	да	Object[]	
sgLocal[]	да	Object[]	

Пример использования

```
curl '127.0.0.1:9007/v2/ie-sg-sg-icmp-rules' \
--header 'Content-Type: application/json' \
--data '{
   "SG": ["sg-example"],
   "sgLocal": ["sg-example-2"]
}'
```

Выходные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].sgLocal — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types[] — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.

название	тип данных
$node.rules[]	Object[]
$node.rules[].SG	String
$node.rules[].sgLocal	String
$node.rules[].logs	Boolean
$node.rules[].trace	Boolean
$node.rules[].ICMP	Object
$node.rules[].ICMP.IPv	String
$node.rules[].ICMP.Types[]	Object[]
$node.rules[].traffic	String
$node.rules[].action	String
$node.rules[].priority	Object
$node.rules[].priority.some	Integer

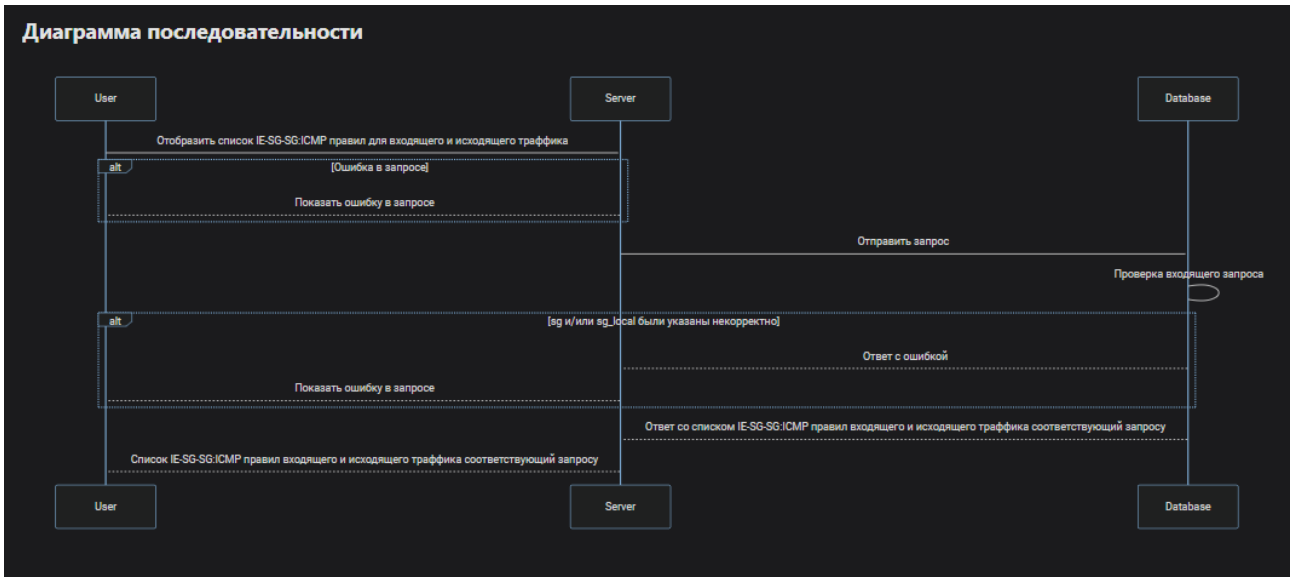
Пример ответа


```
{
   "rules": [{
      "SG": "sg-example",
      "sgLocal": "sg-example-2",
      "logs": "true",
      "trace": "true",
      "traffic": "ingress",
      "ICMP": {
            "IPv": "IPv4",
            "Types": [0, 8]
         },
      "action": "ACCEPT",
      "priority": {
            "some": 100
         }
   }]
}
```


Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



GET /v2/sync/status

Этот метод отображает дату последнего успешного изменения данных.

Входные параметры

Пример использования

```
curl '127.0.0.1:9007/v2/sync/status' \
--header 'Content-Type: application/json'
```

Выходные параметры

- $node.updatedAt — Дата последнего успешного изменения данных

название	тип данных
$node.updatedAt	String

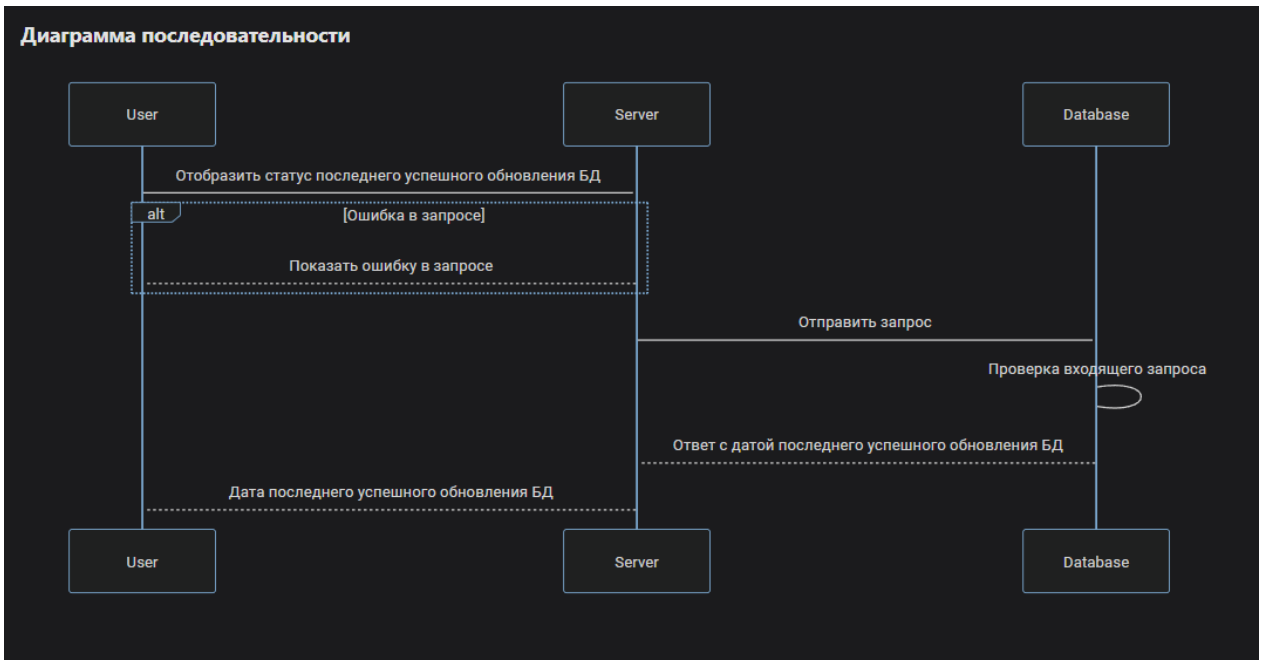
Пример ответа

```
{
   "updatedAt": "2023-11-21T17:02:30.717786Z"
}
```

Возможные ошибки API

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



Terraform

Установка провайдера

bin

Перед развертыванием убедитесь, что у вас установлен terraform:

terraform -v

Далее установкой убедитесь, что вы корректно указали версию, установив переменную VERSION без символа 'v', а также переменные OS и ARCH.

Environment setup

```
export VERSION=1.9.1
export OS=linux
export ARCH=amd64
export GIT=https://github.com/H-BF/sgroups/releases/download/v${VERSION}
export RELEASE_NAME=terraform-provider-sgroups
export PLUGIN_PATH=~/.terraform.d/plugins/registry.terraform.io/sgroups
export
PROVIDER_PATH=${PLUGIN_PATH}/${VERSION}/${OS}_${ARCH}/${RELEASE_NAME
}_v${VERSION}
mkdir -p ${PLUGIN_PATH}/${VERSION}/${OS}_${ARCH}
```

Install provider

```
wget -O ${PROVIDER_PATH} ${GIT}/${RELEASE_NAME}
chmod +x ${PROVIDER_PATH}
```

Terraform setup

```
cat <<EOF >> ~/.terraformrc
plugin_cache_dir = "${HOME}/.terraform.d/plugin-cache"
disable_checkpoint = true
EOF
```

source

Перед развертыванием убедитесь, что вы корректно указали версию, установив переменную VERSION без символа 'v', а также переменные OS и ARCH.

Environment setup

```
export VERSION=1.9.1
export OS=linux
export ARCH=amd64
export RELEASE_NAME=terraform-provider-sgroups
```

```
export PLUGIN_PATH=~/.terraform.d/plugins/registry.terraform.io/sgroups
export
PROVIDER_PATH=${PLUGIN_PATH}/${VERSION}/${OS}_${ARCH}/${RELEASE_NAME
}_v${VERSION}
mkdir -p ${PLUGIN_PATH}/${VERSION}/${OS}_${ARCH}
```

Build provider

```
git clone https://github.com/H-BF/sgroups
cd sgroups
make sgroups-tf-v2
cp bin/${RELEASE_NAME} ${PROVIDER_PATH}
chmod +x ${PROVIDER_PATH}
```

Terraform setup

```
cat <<EOF >> ~/.terraformrc
plugin_cache_dir = "${HOME}/.terraform.d/plugin-cache"
disable_checkpoint = true
EOF
```

Запуск

После установки провайдера, пользователь может приступать к описанию собственных sgroups. В качестве отправной точки, воспользуемся готовым шаблоном.

Install terraform-spec-template

git clone https://github.com/H-BF/swarm-spec-template
cd swarm-spec-template

Следующим шагом настроим файл providers.tf:

- Убедитесь, что вы корректно настроили backend, который будет хранить актуальный terraform-state.
- Убедитесь, что вы корректно указали версию провайдер в секции required_providers.
- Убедитесь, что вы корректно указали IP адресс и порт hbf-server'a. Либо укажите данное значение через переменную окружения SGROUPS_ADDRESS.
- Убедитесь, что вы корректно указали период времени ожидания подключения к серверу. Либо укажите данное значение через переменную окружения SGROUPS_DIAL_DURATION.

Далее убедимся в корректной настройке файл main.tf:

- Убедитесь, что вы корректно настроили параметр source. Важной часть параметра, является то, куда ссылается

Воспользуйтесь предоставленными в репозитории sgroups для проверки работоспособности провайдера. Для этого выполните команду (важно устанавливать флаг --parallelism=1):

Run terraform plan

terraform plan --parallelism=1

Результатом выполнения команды, должен быть список ресурсов, которые описаны в директории spec/.

Следующим шагом будет описание собственных sgroups опираясь, на примеры из документации. Вы можете создавать любую иерархичность в директорию spec/, поскольку при описании сетевых политик sgroups, они ссылаются на имена sgroups, а не на пути расположения файлов до sgroups.

Для того, чтобы применить описанные правила, выполните команду (важно устанавливать флаг --parallelism=1):

Run terraform plan

terraform apply --auto-approve --parallelism=1

Настройка TLS

Установка

Настройка TLS (Transport Layer Security) на hbf-агенте (terraform) обеспечивает шифрование трафика между сервером и клиентом, что повышает безопасность передаваемых данных. В этой документации описан процесс настройки TLS на hbf-агенте (terraform), включая использование предоставленного конфигурационного файла.

Прежде чем приступить к настройке TLS, убедитесь, что у вас есть:

- Установленный hbf-агент (terraform)
- Включен и настроен TLS на hbf-сервере
- Сертификат SSL и соответствующий приватный ключ. Если у вас их нет, вы можете получить их у сертификационного центра (CA) или создать самоподписанный сертификат для тестовых целей.

Шаги по настройке TLS

Создайте файл конфигурации hbf-агента (terraform) для редактирования:

sudo nano /etc/cmd/sgroups-tf-v2/internal/provider/tls-config.tf

Далее необходимо настроить секцию для TLS:

Insecure TLS

```
provider "sgroups" {
  authn = {
    tls = {
      cert = {}
      server_verify = {}
    }
  }
}
```

В случае если сертификат клиента не проверяются, то значения для cert и server_verify можно не указывать.

Secure TLS

provider "sgroups" {

```
  authn = {
    tls = {
      cert = {
        key_file = "/etc/ssl/private/key-file.pem"
        cert_file = "/etc/ssl/certs/cert-file.pem"
      }
      server_verify = {}
    }
  }
}
```

Для подключения secure TLS требуется наличие сертификата, и необходимо указать значения в key-file и cert-file актуальных сертификатов и ключей.

key-file — Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

mTLS

```
provider "sgroups" {
  authn = {
    tls = {
      cert = {
        key_file = "/etc/ssl/private/key-file.pem"
        cert_file = "/etc/ssl/certs/cert-file.pem"
      }
      server_verify = {
        server_name = "server-name"
        root_ca_files = ["file1.pem", "file2.pem", ...]
      }
    }
  }
}
```

Для подключения mTLS требуется наличие сертификата, и необходимо указать значения в key-file и cert-file актуальных сертификатов и ключей.

key-file — Необходимо указать полный путь /etc/ssl/private/key-file.pem или относительный путь ./../key-file.pem с названием файла ключа.

cert-file — Необходимо указать полный путь /etc/ssl/certs/cert-file.pem или относительный путь ./../cert-file.pem с названием файла сертификата.

server_name — При включенном режиме проверки сертификата сервера mTLS можно указать имя сервера. Поле не обязательное для заполнения, в случае если имя сервера не будет указано

то подлинность будет проверяться по данным сертификата.

root_ca_files — При включенном режиме проверки сертификата сервера mTLS необходимо перечислить список certificates authorities с указанием относительного или полного пути к файлам.

Конфигурация ресурсов

Networks

Ресурс Networks представляет собой введенную нами абстракцию, которая позволяет определять группы IP-адресов или подсетей, доступных для управления Host Based NGFW. Эти подсети затем могут быть связаны с конкретными группами безопасности для логического

разделения и использоваться в правилах для разрешения или блокирования доступа к определенным ресурсам в вашей сети.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

название параметра	**описание**	**тип данных**	**значение по умолчанию**
name	Имя Security Group	String	
cidrs[]	Список CIDR, связанных с Security Group	String[]	[]

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

cidrs[]:

- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

Пример использования

name: sg-example

cidrs:

- 10.0.0.0/24

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

название	**обязательность**	**тип данных**	**доп. описание**
items	да	Object	Список ресурсов создаваемые terraform.
items.key	да	String	Уникальный ключ блока items.
items.key.name	да	string	Имя подсети.
items.key.cidr	да	string	Подсеть типа IP.

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.key:

- Имя ключа должно совпадать с значением из поля name.

items.key.name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.key.cidr:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

Пример использования

```
resource "sgroups_networks" "networks" {
   items = {
      key = {
         name = "nw-1"
         cidr = "10.0.0.0/24"
      }
   }
}
```

API

Входные параметры

- networks[] — Массив/Список подсетей типа IP.
- networks[].name — название подсети.
- networks[].network — объект содержащий CIDR подсети

- networks[].network.CIDR — Подсеть типа IP.
- syncOp — Поле определяющее действие с данными из запроса.

название	**обязательность**	**тип данных**	**значение по умолчанию**
networks[]	да	Object[]	
networks[].name	да	String	
networks[].network	да	Object	
networks[].network.CIDR	да	String	
syncOp	да	Enum("Delete", "Upsert", "FullSync")	

Ограничения

networks.networks[].name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

networks.networks[].network.CIDR:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

Пример использования

```
curl '127.0.0.1:9007/v1/sync' \
--header 'Content-Type: application/json' \
--data '{
```

```
  "networks": {
    "networks": [{
      "name": "nw-1",
      "network": {
        "CIDR": "10.0.0.0/24"
      }
    }]
  },
  "syncOp": "Upsert"
}'
```

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

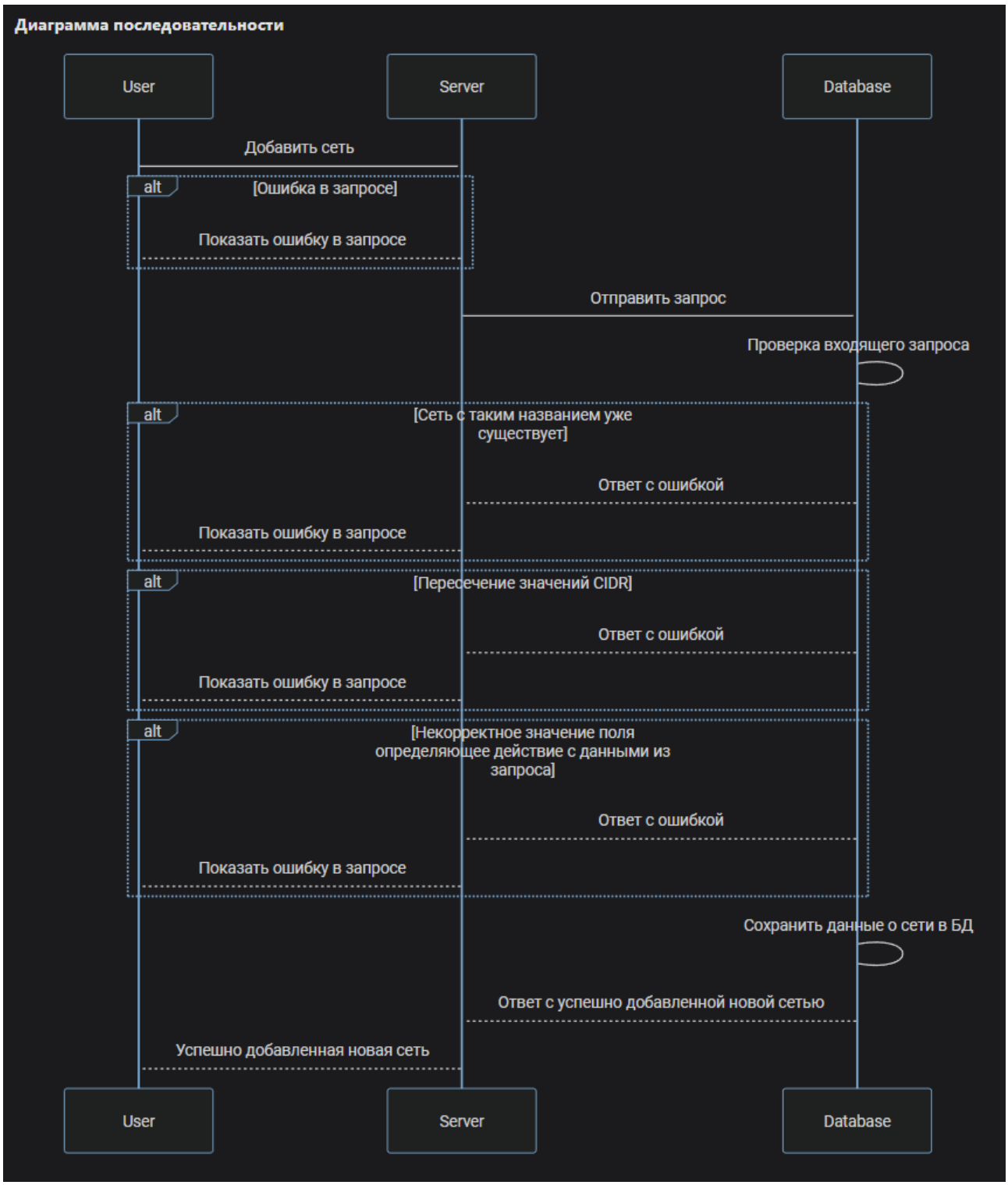
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

В этом разделе мы покажем, как ресурс networks, созданный с помощью Terraform/API, внедряется в настройки nftables. В контексте HBF мы интерпретируем ресурс networks как проекцию на поле elements ресурса IPSet, который функционирует в рамках инструмента nftables, обеспечивая более широкие возможности для управления подсетями.

$IPSet_Name — Наименование IPSet
$type — Описывает тип данных
$flags — Описывает свойства IPSet.
$elements — Указывает массив содержащихся в IPSet элементов подсетей (CIDR)

шаблон параметра	**структура параметра**	**значения**
$IPSet_Name	`^NetIPv[4|6]-.*`	Примеры значений: • **NetIPv4-sg-example** — для описания массивов IP адресов типа v4 • **NetIPv6-sg-example** — для описания массивов IP адресов типа v6
$type	`type`	Могут быть установлены следующие значения: • **ipv4_addr** — для описания массивов IP адресов типа v4 • **ipv6_addr** — для описания массивов IP адресов типа v6
$flags	`flags`	Установлены следующие значения: • **constant** — флаг используется если значение элементов в множестве являются постоянными и не могут быть изменены • **interval** — флаг используется для создания диапазона элементов множества
$elements	`elements = {}`	Значения CIDR, в случае нескольких значений перечисляются через запятую

Шаблон

```
set $IPSet_Name {
   $type
   $flags
   $elements
}
```

Пример использования

```
set NetIPv4-sg-example {
   type ipv4_addr
   flags constant,interval
   elements = { 10.0.0.0/24 } <- networks
}
```

Security Groups

Ресурс Groups представляет собой введенную нами абстракцию, которая позволят объединить подсети в логические группы и применять к ним единые правила сетевого взаимодействия.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного

использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

Далее везде в документе подразумевается что все места, содержащие переменную $transport подразумевается одно из двух значений: icmp, icmp6.

название параметра	**описание**	**тип данных**	**значение по умолчанию**
name	Название Security Group	String	
cidrs[]	Список CIDR, связанных с Security Group	String[]	[]
default_rules	Структура, описывающая правила по умолчанию, для пакетов не соответствующих ни одному из установленных правил в цепочке.	Object	
default_rules.access	Структура, описывающая взаимодействие с пакетами не соответствующими ни одному из установленных правил в таблице.	Object	
default_rules.access.default.logs	Включить/отключить логирование.	Boolean	false
default_rules.access.default.trace	Включить/отключить трассировку.	Boolean	false
default_rules.access.default.action	Определяет действие по умолчанию для пакетов, не соответствующих ни одному из установленных правил в цепочке.	Enum("ACCEPT", "DROP")	ACCEPT
default_rules.access.$transport	Структура, описывающая взаимодействие с ICMP-трафиком по умолчанию. Для обработки ICMP-трафика добавляется соответствующее правило в начало цепочки.	Object	
default_rules.access.$transport.logs	Включить/отключить логирование.	Boolean	false
default_rules.access.$transport.trace	Включить/отключить трассировку.	Boolean	false
default_rules.access.$transport.type[]	Список, определяющий допустимые типы ICMP запросов.	Integer[]	[]
default_rules.access.$transport.action	Действие для пакетов в сформированных правил в цепочке.	Enum("ACCEPT", "DROP")	ACCEPT

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

cidrs[]:

- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

default_rules.access.icmp.type:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

default_rules.access.icmp6.type:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

```
name: sg-example
cidrs:
  - 10.0.0.0/24
  - 11.0.0.0/24
default_rules:
  access:
    default:
      logs: true
      trace: true
      action: ACCEPT
```

```
icmp:
  action: DROP
  logs: true
  trace: true
  type: [0, 8]

icmp6:
  action: DROP
  logs: true
  trace: true
  type: [0, 8]
```

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

Далее везде в документе подразумевается что все места, содержащие переменную $transport (Протокол L3/L4 уровня модели OSI.) подразумевается одно из двух значений: icmp, icmp6.

название	обязательность	тип данных	доп. описание
items	да	Object	Список ресурсов создаваемые terraform ресурсом.
items.key	да	String	Уникальный ключ блока items.
items.key.name	да	String	Имя Security Group.
items.key.networks	да	String[]	Список содержащий имена подсетей которые включены в указанную группу безопасности. Т.к. описывается имплементация Terrafom module в Terraform resource то имя Networks совпадает с CIDR хотя им не является.
items.key.logs	нет	Boolean	Включить/отключить логирование.
items.key.trace	нет	Boolean	Включить/отключить трассировку.
items.key.default_action	да	Enum("DROP", "ACCEPT")	Определяет действие по умолчанию для пакетов, не соответствующих ни одному

название	обязательность	тип данных	доп. описание
			из установленных правил в цепочке.
items.key.$transport	да	Object	Структура, описывающая взаимодействие с ICMP-трафиком по умолчанию. Для обработки ICMP-трафика добавляется соответствующее правило в начало цепочки.
items.key.$transport.logs	нет	Boolean	Включить/отключить логирование.
items.key.$transport.trace	нет	Boolean	Включить/отключить трассировку.
items.key.$transport.type[]	да	String[]	Список, определяющий допустимые типы ICMP запросов.
items.key.$transport.action	да	Enum("ACCEPT", "DROP")	Действие для пакетов в сформированных правил в цепочке.

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.key:

- Имя ключа должно совпадать с значением из поля name.

items.key.name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.key.networks:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

items.key.icmp.type:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

items.key.icmp6.type:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

```
resource "sgroups_groups" "groups" {
   items = {
      key = {
         name          = "sg-example"
         networks      = ["10.0.0.0/24", "11.0.0.0/24"]
         logs          = true
         trace         = true
         default_action = "ACCEPT"
         icmp = {
            logs = true
            trace = true
            type = [0, 8]
            action = "DROP"
         }
         icmp6 = {
            logs = true
            trace = true
            type = [0, 8]
            action = "DROP"
         }
      }
   }
}
```

API

Приведеные ранее примеры создания Security Group с правилами по умолчанию для ICMP с использованием Terraform Module и Terraform Resource преобразуются в два API запроса.

Входные параметры

- groups.grous[] — Структура, содержащая описание создаваемых Security Group
- groups.groups[].defaultAction — представляет действие по умолчанию в конце цепочек для SG
- groups.groups[].logs — Включить/отключить логирование.
- groups.groups[].name — Security Group относительно которой рассматриваются правила.
- groups.groups[].networks — Имена подсетей
- groups.groups[].trace — Включить/отключить трассировку.
- sgIcmpRules.rules[] — Структура, содержащая описание создаваемых правил.
- sgIcmpRules.rules[].ICMP — Структура, содержащая описание создаваемых правил

типа ICMP.

- sgIcmpRules.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- sgIcmpRules.rules[].ICMP.Types — Список, определяющий допустимые типы ICMP запросов.
- sgIcmpRules.rules[].SG — Security Group относительно которой рассматриваются правила.
- sgIcmpRules.rules[].logs — Включить/отключить логирование.
- sgIcmpRules.rules[].trace — Включить/отключить трассировку.
- sgIcmpRules.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- syncOp — Поле определяющее действие с данными из запроса.

название	обязательность	тип данных	Значение по умолчанию	API request	
				groups	sgIcmpRules
groups.groups[]	да	Object[]		✔	
groups.groups[].defaultAction	да	Enum("ACCEPT", "DROP")		✔	
groups.groups[].logs	нет	Boolean	false	✔	
groups.groups[].name	да	String		✔	
groups.groups[].networks	нет	String[]	[]	✔	
groups.groups[].trace	нет	Boolean	false	✔	
sgIcmpRules.rules[]	да	Object[]			✔
sgIcmpRules.rules[].ICMP	да	Object			✔
sgIcmpRules.rules[].ICMP.IPv	да	Enum("IPv4", "IPv6")			✔
sgIcmpRules.rules[].ICMP.Types	нет	String[]	[]		✔
sgIcmpRules.rules[].SG	да	String			✔
sgIcmpRules.rules[].logs	нет	Boolean	false		✔
sgIcmpRules.rules[].trace	нет	Boolean	false		✔
sgIcmpRules.rules[].action	да	Enum("UNDEF", "ACCEPT", "DROP")			✔
syncOp	да	Enum("Delete", "Upsert", "FullSync")		✔	✔

Ограничения

groups.groups[].name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

groups.groups[].networks[]:

- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

sgIcmpRules.rules[].SG:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

$node.rules[].type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.
- Пример использования

Пример использования

Security Groups	ICMP

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
   "groups": {
      "groups": [{
         "defaultAction": "ACCEPT",
         "logs": true,
         "name": "sg-example",
         "networks":                ["10.0.0.0/24",
"11.0.0.0/24"],
         "trace": true
      }]
   },
   "syncOp": "Upsert"
}'
```

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
   "sgIcmpRules": {
      "rules": [{
         "ICMP": {
            "IPv": "IPv4",
            "Types": [0,8]
         },
         "SG": "sg-example",
         "logs": true,
         "trace": true,
         "action": "DROP",
      },
      {
         "ICMP": {
            "IPv": "IPv6",
            "Types": [0,8]
         },
         "SG": "sg-example",
         "logs": true,
         "trace": true,
         "action": "DROP",
      }]
   },
   "syncOp": "Upsert"
}'
```

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

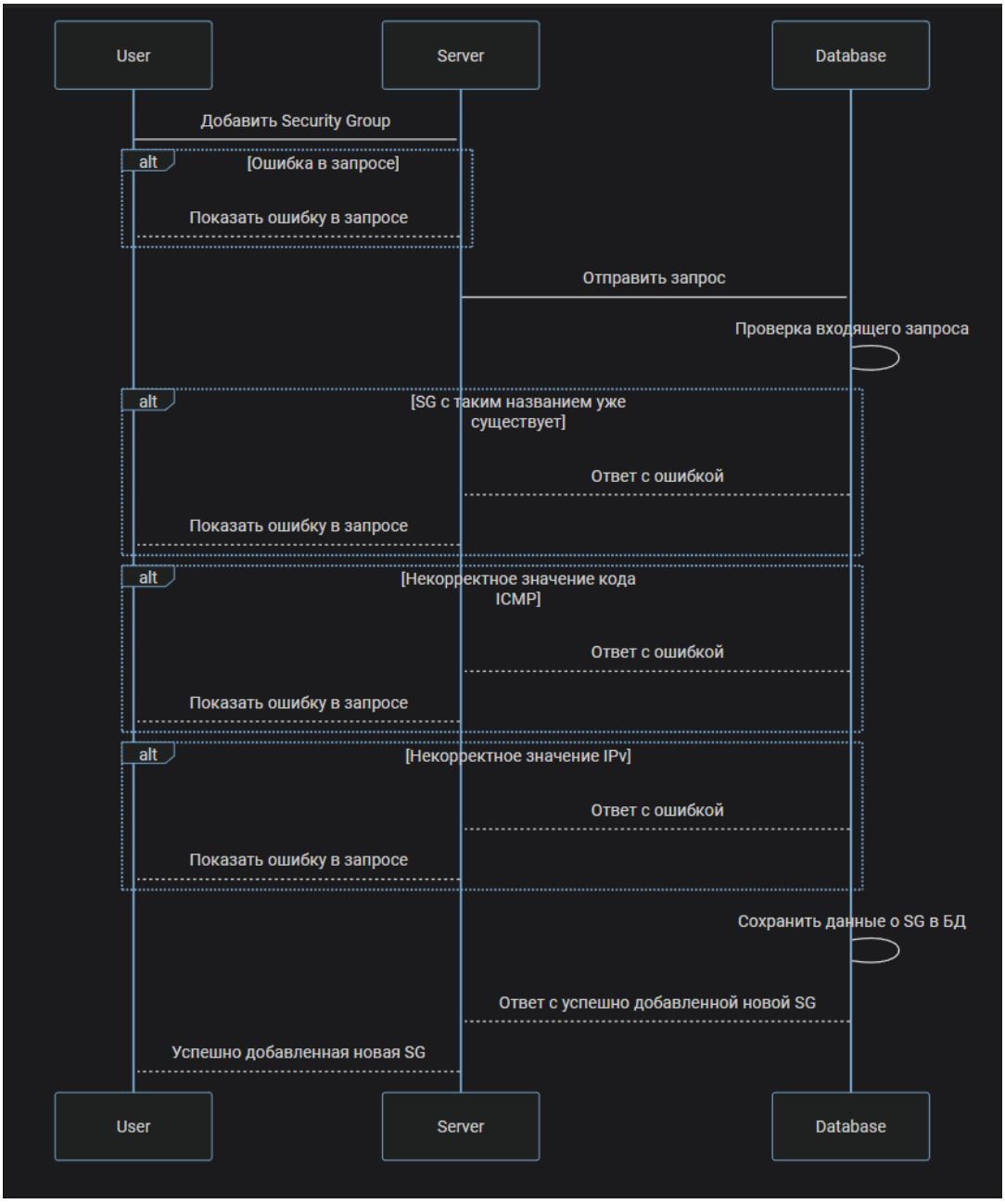
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

В этом разделе мы покажем, как ресурс groups, созданный с помощью Terraform/API, внедряется в настройки nftables. В контексте HBF мы интерпретируем ресурс groups как проекцию на ресурсы IPSet и Chains которые функционируют в рамках инструмента nftables, обеспечивая более широкие возможности для управления подсетями и правилами.

- $Trace — Включить/отключить трассировку.
- $Transport — Протокол L3/L4 уровня модели OSI.
- $TypeList — Список, определяющий допустимые типы ICMP запросов.
- $Log — Включить/отключить логирование.
- $Counter — Счетчик количества байтов и пакетов.
- $Verdict — Результат применения правила, определяющий действие, которое будет применено к пакету.

шаблон параметра	**структура параметра**	**значение**	**общие**	**icmp**
$Trace	`nftrace set`	• **1** — трассировка включена • **0** — трассировка выключена	✔	✔
$Transport	`icmp`			✔
$TypeList	`type {}`	Набор целочисленных значений от 0 до 255		✔
$Log	`log`	`level debug flags ip options`	✔	✔
$Counter	`counter`	`packets 0 bytes 0`	✔	✔
$Verdict	`accept`	*$Verdict определяет действие, которое будет применено к пакету в соответсвии с правилом. Это поле может принимать значение* `accept` *или* `drop` *в зависимости от указанного в правиле.* *Подробнее: Verdict statement*	✔	✔

Шаблон

```
chain INGRESS-INPUT-sgName {
   # **********
   $Trace $Transport $TypeList $Counter $Log $Verdict    //ICMP
   # **********
   $Trace $Counter $Log $Verdict                    //Общие
}

chain EGRESS-POSTROUTING-sgName {
   # **********
   $Trace $Transport $TypeList $Counter $Log $Verdict    //ICMP
   # **********
   $Trace $Counter $Log $Verdict                    //Общие
}
```

Пример использования

```
chain INGRESS-INPUT-sg-example {
   # **********
   nftrace set 1 icmp    type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options drop
//ICMP
```

```
    nftrace set 1 icmpv6 type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options drop
//ICMP
    # **********
    nftrace set 1 counter packets 0 bytes 0 log level debug flags ip options accept                    //Общие
}


chain EGRESS-POSTROUTING-sg-example {
    # **********
    nftrace set 1 icmp    type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options drop
//ICMP
    nftrace set 1 icmpv6 type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options drop
//ICMP
    # **********
    nftrace set 1 counter packets 0 bytes 0 log level debug flags ip options accept                    //Общие
}
```

Sgroup to Sgroup

Данный тип правил управляет обменом данными между различными группами безопасности. Он автоматически создает два правила на хостах: одно для исходящего трафика от инициирующей стороны и другое для входящего трафика от группы безопасности, к которой предоставлен доступ.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

Далее везде в документе подразумевается что все места, содержащие переменную $transport иогут принимать одно из четырех значений: tcp, udp, icmpIPv4, icmpIPv6.

- rules — Структура, содержащая описание создаваемых правил.
- rules.s2s[] — Список правил, описывающий сетевое взаимодействие текущей Security Group с внешней Security Group.
- rules.s2s[].sgroupSet[] — Список, содержащий названия Security Group(s).
- rules.s2s[].access.$transport — Протокол L3/L4 уровня модели OSI.
- rules.s2s[].access.$transport.action — Действие для пакетов в сформированных правил в цепочке.
- rules.s2s[].access.$transport.priority — Поле определяющее порядок применения правил

в цепочке.

- rules.s2s[].access.$transport.log — Включить/отключить логирование.
- rules.s2s[].access.$transport.trace — Включить/отключить трассировку.
- rules.s2s[].access.$transport.ports[].description — Формальное текстовое описание.
- rules.s2s[].access.$transport.ports[].ports_to[] — Набор открытых портов получателя
- rules.s2s[].access.$transport.ports[].ports_from[] — Набор открытых портов отправителя.
- rules.s2s[].access.$transport.types[].description — Формальное текстовое описание.
- rules.s2s[].access.$transport.types[].type[] — Список, определяющий допустимые типы ICMP запросов.

Области применения полей относительно используемого протокола

название параметра	тип данных	значение по умолчанию	transport*		
			TCP	UDP	ICMP
rules	Object[]	[]	✔	✔	✔
rules.s2s	Object		✔	✔	✔
rules.s2s.sgroupSet[]	String[]		✔	✔	✔
rules.s2s.access.$transport	Object[]		✔	✔	✔
rules.s2s[].access.$transport.action	Enum("ACCEPT", "DROP")		✔	✔	✔
rules.s2s[].access.$transport.priority	String		✔	✔	✔
rules.s2s[].access.$transport.log	Boolean	false	✔	✔	✔
rules.s2s[].access.$transport.trace	Boolean	false	✔	✔	✔
rules.s2s.access.$transport.ports[].description	String	""	✔	✔	✔
rules.s2s.access.$transport.ports[].ports_to[]	Integer[]	null	✔	✔	
rules.s2s.access.$transport.ports[].ports_from[]	Integer[]	null	✔	✔	
rules.s2s[].access.$transport.types[].type[]	Integer[]	null			✔

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

rules.s2s[].sgroupSet:

- Длина значения элемента не должна превышать 256 символов.
- Значение элемента должно начинаться и заканчиваться символами без пробелов.

- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.

rules.s2s[].access.$transport.priority:

- Значения должны находиться в интервале от -32768 до 32767

rules.s2s[].access.$transport.ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

rules.s2s[].access.$transport.ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

rules.s2s[].access.$transport.types[].type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP	UDP	ICMP4	ICMP6

name: sg-local-example
rules:
  s2s:
  - sgroupSet:
  - sg-example
  access:
  tcp:
  action: ACCEPT
  priority: -200
  logs: true
  trace: true
  ports:
  - description: ""
  ports_from:
  - 64231
  ports_to:
  - 443
  - 80
	name: sg-local-example
	rules:
	  s2s:
	  - sgroupSet:
	  - sg-example
	  access:
	  udp:
	  action: ACCEPT
	  priority: -200
	  logs: true
	  trace: true
	  ports:
	  - description: ""
	  ports_from:
	  - 64231
	  ports_to:
	  - 443
	  - 80
		name: sg-local-example
		rules:
		  s2s:
		  - sgroupSet:
		  - sg-example
		  access:
		  icmpIPv4:
		  action: ACCEPT
		  priority: -300
		  logs: true
		  trace: true
		  types:
		  - description: ""
		  Types:
		  - 0
		  - 8
			name: sg-local-example
			rules:
			  s2s:
			  - sgroupSet:
			  - sg-example
			  access:
			  icmpIPv6:
			  action: ACCEPT
			  priority: -300
			  logs: true
			  trace: true
			  types:
			  - description: ""
			  Types:
			  - 0
			  - 8

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

- items — Список ресурсов создаваемые terraform ресурсом.
- items.$ruleName — Уникальное имя создаваемого ресурса.
- items.$ruleName.transport — Протокол L3/L4 уровня модели OSI.
- items.$ruleName.sg_from — Security Group, с которой устанавливаются правила взаимодействия.
- items.$ruleName.sg_to — Security Group, с которой устанавливаются правила взаимодействия.

- items.$ruleName.ports — Блок описывающий набор пар портов (src-dst).
- items.$ruleName.ports[].d — Набор открытых портов получателя
- items.$ruleName.ports[].s — Набор открытых портов отправителя.
- items.$ruleName.logs — Включить/отключить логирование.
- items.$ruleName.trace — Включить/отключить трассировку.
- items.$ruleName.ip_v — Версия IP для ICMP (IPv4 или IPv6).
- items.$ruleName.type — Список, определяющий допустимые типы ICMP запросов.
- items.$ruleName.action — Действие для пакетов в сформированных правил в цепочке.
- items.$ruleName.priority — Поле определяющее порядок применения правил в цепочке.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
items	да	Object		✔	✔	✔
items.$ruleName	да	Object		✔	✔	✔
items.$ruleName.transport	да	Enum("TCP", "UDP")		✔	✔	
items.$ruleName.sg_from	да	String		✔	✔	✔
items.$ruleName.sg_to	да	String		✔	✔	✔
items.$ruleName.ports	нет	Object[]	null	✔	✔	
items.$ruleName.ports[].d	нет	String	""	✔	✔	
items.$ruleName.ports[].s	нет	String	""	✔	✔	
items.$ruleName.logs	нет	Boolean	false	✔	✔	✔
items.$ruleName.trace	нет	Boolean	false	✔	✔	✔
items.$ruleName.ip_v	да	Enum("IPv4", "IPv6")				✔
items.$ruleName.type	да	String[]	null			✔
items.$ruleName.action	да	Enum("ACCEPT", "DROP")		✔	✔	✔
items.$ruleName.priority	нет	Integer		✔	✔	✔

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.$ruleName:

- Форма ruleName должна быть организована в соответствии с определенной последовательностью, которую нужно соблюдать "${transport}:sg-

local(${sg_local})sg(${sg})".

items.$ruleName.sg_from:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.$ruleName.sg_to:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.$ruleName.ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

items.$ruleName.ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

items.$ruleName.type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

items.$ruleName.priority:

- Значения должны находиться в интервале от -32768 до 32767

Пример использования

TCP	UDP

resource "sgroups_rules" "rules" {
  items = {
  "tcp:sg(sg-local-example)sg(sg-example)" = {
  sg_from = "sg-local-example"
  sg_to = "sg-example"
  logs = true
  trace = true
  transport = "tcp"
  ports = [{
  d = "443,80"
  s = "64231"
  }]
  action = "ACCEPT"
  priority = -200
  }
  }
}
	resource "sgroups_rules" "rules" {
	  items = {
	  "udp:sg(sg-local-example)sg(sg-example)"
	= {
	  sg_from = "sg-local-example"
	  sg_to = "sg-example"
	  logs = true
	  trace = true
	  transport = "udp"
	  ports = [{
	  d = "443,80"
	  s = "64231"
	  }]
	  action = "ACCEPT"
	  priority = -200
	  }
	  }
	}

ICMP4	ICMP6

resource "sgroups_icmp_rules" "rules" {
  items = {
  "sg(sg-local-example)sg(sg-
example)icmp4" => {
  sg_from = "sg-local-example"
  sg_to = "sg-example"
  logs = true
  trace = true
  ip_v = "IPv4"
  type = [0,8]
  action = "ACCEPT"
  priority = -300
  }
  }
}
	resource "sgroups_icmp_rules" "rules" {
	  items = {
	  "sg(sg-local-example)sg(sg-example)icmp4"
	=> {
	  sg_from = "sg-local-example"
	  sg_to = "sg-example"
	  logs = true
	  trace = true
	  ip_v = "IPv4"
	  type = [0,8]
	  action = "ACCEPT"
	  priority = -300
	  }
	  }
	}

API

Далее везде в документе подразумевается что все места, содержащие переменную $node,

могут принять одно из двух значений: sgRules, sgSgIcmpRules.

Входные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].sgFrom — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].sgTo — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.
- syncOp — Поле определяющее действие с данными из запроса.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
$node.rules[]	да	Object[]	null	✔	✔	✔
$node.rules[].sgFrom	да	String		✔	✔	✔
$node.rules[].sgTo	да	String		✔	✔	✔
$node.rules[].logs	нет	Boolean	false	✔	✔	✔
$node.rules[].trace	нет	Boolean	false	✔	✔	✔
$node.rules[].ports	нет	Object[]	null	✔	✔	
$node.rules[].ports[].d	нет	String	null	✔	✔	
$node.rules[].ports[].s	нет	String	null	✔	✔	
$node.rules[].transport	нет	Enum("TCP", "UDP")	TCP	✔	✔	

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
$node.rules[].ICMP	да	Object				✔
$node.rules[].ICMP.IPv	да	Enum("IPv4", "IPv6")				✔
$node.rules[].ICMP.Types	нет	String[]	[]			✔
$node.rules[].action	да	Enum("UNDEF", "ACCEPT", "DROP")		✔	✔	✔
$node.rules[].priority	нет	Object		✔	✔	✔
$node.rules[].priority.some	нет	Integer		✔	✔	✔
syncOp	да	Enum("Delete", "Upsert", "FullSync")		✔	✔	✔

Ограничения

sgSgRules.rules[].sgFrom:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

sgSgRules.rules[].sgTo:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

sgSgRules.rules[].ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

sgSgRules.rules[].ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

sgSgRules.rules[].ICMP.Types[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP	UDP

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
"sgRules": {
"rules": [{
"sgFrom": "sg-local-example",
"sgTo": "sg-example",
"logs": true,
"trace": true,
"ports": [{
"d": "443,80",
"s": "64231"
}],
"transport": "TCP",
"action": "ACCEPT",
"priority": {
"some": -200
}
}]
},
"syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	"sgRules": {
	"rules": [{
	"sgFrom": "sg-local-example",
	"sgTo": "sg-example",
	"logs": true,
	"trace": true,
	"ports": [{
	"d": "443,80",
	"s": "64231"
	}],
	"transport": "UDP",
	"action": "ACCEPT",
	"priority": {
	"some": -200
	}
	}]
	},
	"syncOp": "Upsert"
	}'

ICMP4	ICMP6

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
"sgSgIcmpRules": {
"rules": [{
"sgFrom": "sg-local-example",
"sgTo": "sg-example",
"logs": true,
"trace": true
"ICMP": {
"IPv": "IPv4",
"Types": [0,8]
},
"action": "ACCEPT",
"priority": {
"some": -300
}
}]
},
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	"sgSgIcmpRules": {
	"rules": [{
	"sgFrom": "sg-local-example",
	"sgTo": "sg-example",
	"logs": true,
	"trace": true
	"ICMP": {
	"IPv": "IPv6",
	"Types": [0,8]
	},
	"action": "ACCEPT",
	"priority": {
	"some": -300
	}
	}]
	},

"syncOp": "Upsert" }'	"syncOp": "Upsert" }'

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

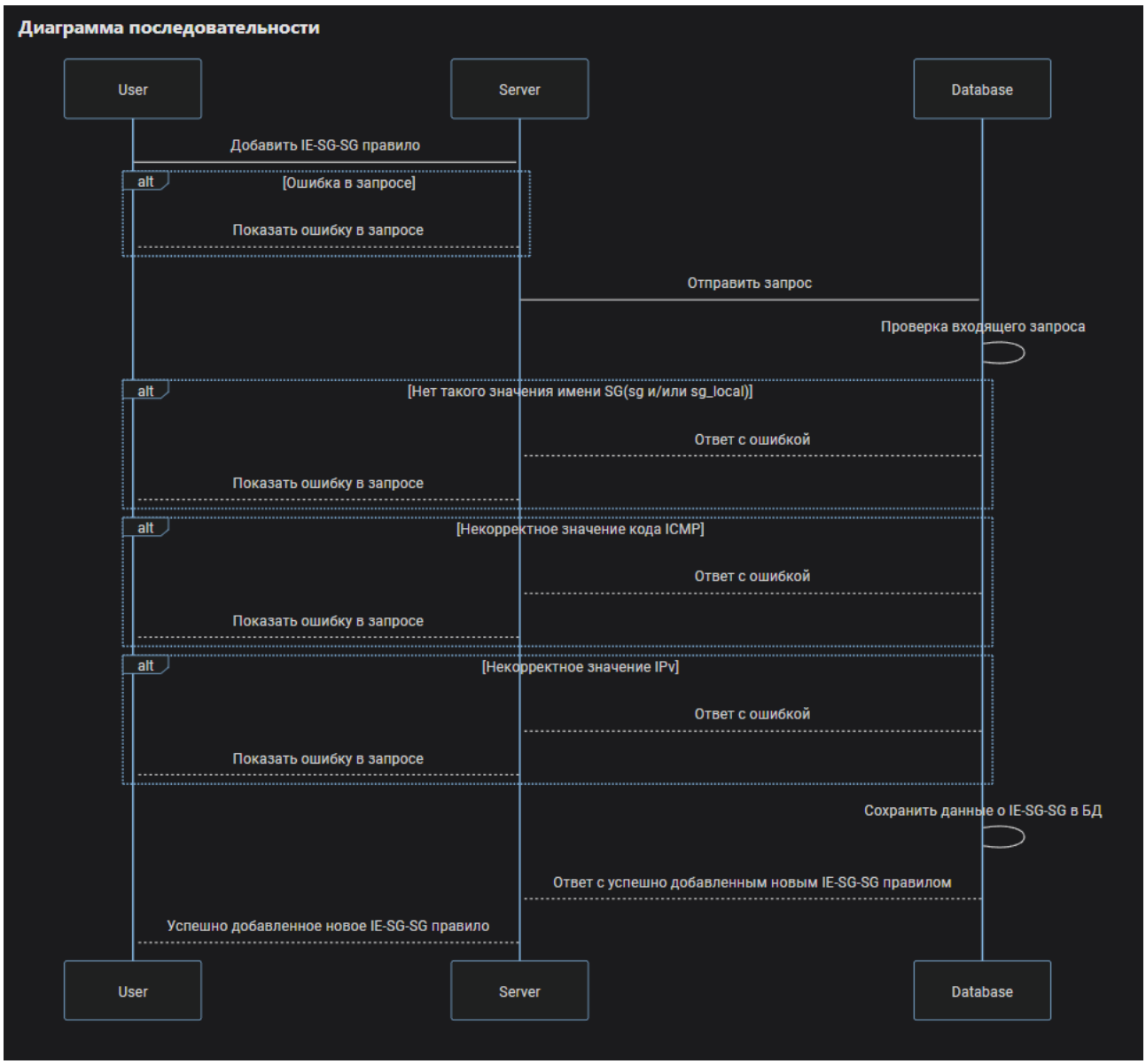
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

В этом разделе мы покажем, как правила фильтрации трафика, созданные с помощью Terraform и API, внедряются в настройки nftables. Это позволяет легко интегрировать сложные правила безопасности прямо в вашу систему фильтрации трафика.

- $Trace — Включить/отключить трассировку.
- $SrcSgroup — Security Group, с которой устанавливаются правила взаимодействия.
- $DstGroup — Security Group, с которой устанавливаются правила взаимодействия.
- $Transport — Протокол L3/L4 уровня модели OSI.
- $NftRuleType — Характеристика описывающая, что принимается трафик типа ip.

- $IcmpTypeList — Список, определяющий допустимые типы ICMP запросов.
- $SrcPorts — Набор открытых портов отправителя.
- $DstPorts — Набор открытых портов получателя
- $NftCounter — Счетчик количества байтов и пакетов.
- $Log — Включить/отключить логирование.
- $NftRuleVerdict — Результат применения правила, определяющий действие, которое будет применено к пакету.

Области применения полей относительно используемого протокола

шаблон параметра	структура параметра	значения	transport*		
			TCP	UDP	ICMP
$Trace	`nftrace set`	• **1** - трассировка включена • **0** - трассировка выключена	✔	✔	✔
$SrcSgroup	`saddr`	`@${IPSet(sgName)}`	✔	✔	✔
$DstSgroup	`daddr`	`@${IPSet(sgName)}`	✔	✔	✔
$Transport	`tcp | udp | icmp`		✔	✔	✔
$NftRuleType	`ip`		✔	✔	✔
$IcmpTypeList	`type {}`	Набор целочисленных значений от 0 до 255			✔
$SrcPorts	`sport {}`	Набор целочисленных значений от 0 до 65535	✔	✔	
$DstPorts	`dport {}`	Набор целочисленных значений от 0 до 65535	✔	✔	
$NftCounter	`counter`	`packets 0 bytes 0`	✔	✔	✔
$Log	`log`	`level debug flags ip options`	✔	✔	✔
$NftRuleVerdict	`accept`	*$NftRuleVerdict определяет действие, которое будет применено к пакету в соответсвии с правилом. Это поле может принимать значение* `accept` *или* `drop` *в зависимости от указанного в правиле.* *Подробнее: Verdict statement*	✔	✔	✔

Пример использования

TCP

Шаблон

```
chain INGRESS-INPUT-sgName {
   # **********
   ${Trace} ${RuleType} ${SrcSgroup} ${Transport} ${SrcPorts} ${DstPorts} ${Counter}
${Log} ${Verdict}
   # **********
```

```
}

chain EGRESS-POSTROUTING-sgName {
   # **********
   ${Trace} ${RuleType} ${DstSgroup} ${Transport} ${SrcPorts} ${DstPorts} ${Counter}
${Log} ${Verdict}
   # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-example-to {
   # **********
   nftrace set 1 ip saddr NetIPv4-sg-example-from tcp dport { 80, 443 } sport { 64231 } counter
packets 0 bytes 0 log level debug flags ip options accept
   nftrace set 1 ip saddr NetIPv6-sg-example-from tcp dport { 80, 443 } sport { 64231 } counter
packets 0 bytes 0 log level debug flags ip options accept
   # **********
}

chain EGRESS-POSTROUTING-sg-example-from {
   # **********
   nftrace set 1 ip daddr NetIPv4-sg-example-to tcp dport { 80, 443 } sport { 64231 } counter
packets 0 bytes 0 log level debug flags ip options accept
   nftrace set 1 ip daddr NetIPv6-sg-example-to tcp dport { 80, 443 } sport { 64231 } counter
packets 0 bytes 0 log level debug flags ip options accept
   # **********
}
```

UDP

Шаблон

```
chain INGRESS-INPUT-sgName {
   # **********
   ${Trace} ${RuleType} ${SrcSgroup} ${Transport} ${SrcPorts} ${DstPorts} ${Counter}
${Log} ${Verdict}
   # **********
}

chain EGRESS-POSTROUTING-sgName {
   # **********
   ${Trace} ${RuleType} ${DstSgroup} ${Transport} ${SrcPorts} ${DstPorts} ${Counter}
${Log} ${Verdict}
   # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-example-to {
   # **********
```

```
    nftrace set 1 ip saddr NetIPv4-sg-example-from udp dport { 80, 443 } sport { 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr NetIPv6-sg-example-from udp dport { 80, 443 } sport { 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}

chain EGRESS-POSTROUTING-sg-example-from {
    # **********
    nftrace set 1 ip daddr NetIPv4-sg-example-to udp dport { 80, 443 } sport { 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr NetIPv6-sg-example-to udp dport { 80, 443 } sport { 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

ICMP

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    ${Trace} ${RuleType} ${SrcSgroup} ${Transport} ${TypeList} ${Counter} ${Log} ${Verdict}
    # **********
}

chain EGRESS-POSTROUTING-sgName {
    # **********
    ${Trace} ${RuleType} ${DstSgroup} ${Transport} ${TypeList} ${Counter} ${Log} ${Verdict}
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-example-to {
    # **********
    nftrace set 1 ip saddr NetIPv4-v4-sg-example-from icmp type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr NetIPv6-v6-sg-example-from icmp type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}

chain EGRESS-POSTROUTING-sg-example-from {
    # **********
    nftrace set 1 ip daddr NetIPv4-sg-example-to icmp type { 0, 8 } counter packets 0 bytes 0 log level debug flags ip options accept
```

```
    nftrace set 1 ip daddr NetIPv6-sg-example-to icmp type { 0, 8 } counter packets 0 bytes 0 log
level debug flags ip options accept
    # **********
}
```

Sgroup to Sgroup (I/E)

Ресурс Security Group to Security Group представляет собой введенную нами абстракцию, которая обеспечивает гибкое управление и контроль сетевого трафика между разными группами безопасности, используя протоколы TCP, UDP и ICMP. Этот ресурс позволяет точно настраивать, какой трафик может передаваться между группами, обеспечивая таким образом высокий уровень защиты и контроля в сетевой инфраструктуре.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

Далее везде в документе подразумевается что все места, содержащие переменную $traffic, могут принять одно из двух значений: Ingress Egress. Аналогчично для $transport подразумевается одно из четырех значений: tcp, udp, icmpIPv4, icmpIPv6

- rules — Структура, содержащая описание создаваемых правил.
- rules.$traffic[] — Поле описывающий направление трафика.
- rules.$traffic[].sgroupSet[] — Список, содержащий названия Security Group(s).
- rules.$traffic[].access.$transport — Протокол L3/L4 уровня модели OSI.
- rules.$traffic[].access.$transport.action — Действие для пакетов в сформированных правил в цепочке.
- rules.$traffic[].access.$transport.priority — Поле определяющее порядок применения правил в цепочке.
- rules.$traffic[].access.$transport.log — Включить/отключить логирование.
- rules.$traffic[].access.$transport.trace — Включить/отключить трассировку.
- rules.$traffic[].access.$transport.ports[].description — Формальное текстовое описание.
- rules.$traffic[].access.$transport.ports[].ports_to[] — Набор открытых портов получателя
- rules.$traffic[].access.$transport.ports[].ports_from[] — Набор открытых портов отправителя.
- rules.$traffic[].access.$transport.types[].description — Формальное текстовое описание.
- rules.$traffic[].access.$transport.types[].type[] — Список, определяющий допустимые типы ICMP запросов.

Области применения полей относительно используемого протокола

название параметра	тип данных	значение по умолчанию	transport*		
			TCP	UDP	ICMP
rules	Object[]	[]	✔	✔	✔
rules.$traffic[]	Object[]		✔	✔	✔
rules.$traffic[].sgroupSet[]	String[]		✔	✔	✔
rules.$traffic[].access.$transport	Object		✔	✔	✔
rules.$traffic[].access.$transport.action	Enum("ACCEPT", "DROP")		✔	✔	✔
rules.$traffic[].access.$transport.priority	String		✔	✔	✔
rules.$traffic[].access.$transport.log	Boolean	false	✔	✔	✔
rules.$traffic[].access.$transport.trace	Boolean	false	✔	✔	✔
rules.$traffic[].access.$transport.ports[]	Object[]	""	✔	✔	✔
rules.$traffic[].access.$transport.ports[].description	String	""	✔	✔	✔
rules.$traffic[].access.$transport.ports[].ports_to[]	Integer[]	null	✔	✔	
rules.$traffic[].access.$transport.ports[].ports_from[]	Integer[]	null	✔	✔	
rules.$traffic[].access.$transport.types[]	Object[]	""	✔	✔	✔
rules.$traffic[].access.$transport.types[].description	String	""	✔	✔	✔
rules.$traffic[].access.$transport.types[].type[]	Integer[]	null			✔

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

rules.$traffic[].sgroupSet:

- Длина значения элемента не должна превышать 256 символов.
- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.

rules.$traffic[].access.$transport.priority:

- Значения должны находиться в интервале от -32768 до 32767

rules.$traffic[].access.$transport.ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

rules.$traffic[].access.$transport.ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

rules.$traffic[].access.$transport.types[].type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP

Ingress	Egress

name: "sg-local-example"
rules:
  ingress:
  - sgroupSet:
  - "sg-example"
  access:
  tcp:
  action: ACCEPT
  priority: 0
  logs: true
  trace: true
  ports:
  - description: "example"
  ports_from:
  - 64231
  ports_to:
  - 443
  - 80
	name: "sg-local-example"
	rules:
	  egress:
	  - sgroupSet:
	  - "sg-example"
	  access:
	  tcp:
	  action: ACCEPT
	  priority: 0
	  logs: true
	  trace: true
	  ports:
	  - description: "example"
	  ports_from:
	  - 64231
	  ports_to:
	  - 443
	  - 80

UDP

Ingress	Egress

name: "sg-local-example"
rules:
  ingress:
  - sgroupSet:
  - "sg-example"
  access:
  udp:
  action: ACCEPT
  priority: 0
  logs: true
  trace: true
  ports:
  - description: "example"
  ports_from:
  - 64231
  ports_to:
  - 443
  - 80
	name: "sg-local-example"
	rules:
	  egress:
	  - sgroupSet:
	  - "sg-example"
	  access:
	  udp:
	  action: ACCEPT
	  priority: 0
	  logs: true
	  trace: true
	  ports:
	  - description: "example"
	  ports_from:
	  - 64231
	  ports_to:
	  - 443
	  - 80

ICMP4

Ingress	Egress

name: "sg-local-example"
rules:
  ingress:
  - sgroupSet:
  - "sg-example"
  access:
  icmpIPv4:
  action: ACCEPT
  priority: -100
  logs: true
  trace: true
  types:
  - description: "example"
  type:
  - 0
  - 8
	name: "sg-local-example"
	rules:
	  egress:
	  - sgroupSet:
	  - "sg-example"
	  access:
	  icmpIPv4:
	  action: ACCEPT
	  priority: -100
	  logs: true
	  trace: true
	  types:
	  - description: "example"
	  type:
	  - 0
	  - 8

ICMP6

Ingress	Egress

name: "sg-local-example"
rules:
  ingress:
  - sgroupSet:
  - "sg-example"
  access:
  icmpIPv6:
  action: ACCEPT
  priority: -100
  logs: true
  trace: true
  types:
  - description: "example"
  type:
  - 0
  - 8
	name: "sg-local-example"
	rules:
	  egress:
	  - sgroupSet:
	  - "sg-example"
	  access:
	  icmpIPv6:
	  action: ACCEPT
	  priority: -100
	  logs: true
	  trace: true
	  types:
	  - description: "example"
	  type:
	  - 0
	  - 8

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

- items — Список ресурсов создаваемые terraform ресурсом.
- items.$ruleName — Уникальное имя создаваемого ресурса.
- items.$ruleName.traffic — Поле описывающий направление трафика.
- items.$ruleName.transport — Протокол L3/L4 уровня модели OSI.
- items.$ruleName.sg — Security Group, с которой устанавливаются правила взаимодействия.
- items.$ruleName.sg_local — Security Group относительно которой рассматриваются правила.
- items.$ruleName.ports — Блок описывающий набор пар портов (src-dst).
- items.$ruleName.ports[].d — Набор открытых портов получателя
- items.$ruleName.ports[].s — Набор открытых портов отправителя.
- items.$ruleName.logs — Включить/отключить логирование.
- items.$ruleName.trace — Включить/отключить трассировку.
- items.$ruleName.ip_v — Версия IP для ICMP (IPv4 или IPv6).
- items.$ruleName.type — Список, определяющий допустимые типы ICMP запросов.
- items.$ruleName.action — Действие для пакетов в сформированных правил в цепочке.
- items.$ruleName.priority — Поле определяющее порядок применения правил в цепочке.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
items	да	Object		✔	✔	✔
items.$ruleName	да	Object		✔	✔	✔
items.$ruleName.traffic	да	Enum("ingress", "egress")		✔	✔	✔
items.$ruleName.transport	да	Enum("TCP", "UDP")		✔	✔	
items.$ruleName.sg	да	String		✔	✔	✔
items.$ruleName.sg_local	да	String		✔	✔	✔
items.$ruleName.ports	нет	Object[]	null	✔	✔	
items.$ruleName.ports[].d	нет	String	""	✔	✔	
items.$ruleName.ports[].s	нет	String	""	✔	✔	
items.$ruleName.logs	нет	Boolean	false	✔	✔	✔
items.$ruleName.trace	нет	Boolean	false	✔	✔	✔
items.$ruleName.ip_v	да	Enum("IPv4", "IPv6")				✔
items.$ruleName.type	да	String[]	null			✔
items.$ruleName.action	да	Enum("ACCEPT", "DROP")		✔	✔	✔
items.$ruleName.priority	нет	Integer		✔	✔	✔

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.$ruleName:

- Форма ruleName должна быть организована в соответствии с определенной последовательностью, которую нужно соблюдать "${transport}:sg-local(${sg_local})sg(${sg})${traffic}".

items.$ruleName.sg:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.$ruleName.sg_local:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.$ruleName.ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

items.$ruleName.ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

items.$ruleName.type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

items.$ruleName.priority:

- Значения должны находиться в интервале от -32768 до 32767

Пример использования

TCP

Ingress	Egress

resource "sgroups_ie_rules" "rules" {
  items = {
  "tcp:sg-local(sg-local-example)sg(sg-example)ingress" = {
  traffic = "ingress"
  logs = true
  trace = true
  sg_local = "sg-local-example"
  sg = "sg-example"
  transport = "tcp"
  ports = [{
  d = "80"
  s = ""
  }]
  action = "ACCEPT"
  priority = 0
  }
  }
}
	resource "sgroups_ie_rules" "rules" {
	  items = {
	  "tcp:sg-local(sg-local-example)sg(sg-example)egress" = {
	  traffic = "egress"
	  logs = true
	  trace = true
	  sg_local = "sg-local-example"
	  sg = "sg-example"
	  transport = "tcp"
	  ports = [{
	  d = "80"
	  s = ""
	  }]
	  action = "ACCEPT"
	  priority = 0
	  }
	  }
	}

UDP

Ingress	Egress

resource "sgroups_ie_rules" "rules" {
  items = {
  "udp:sg-local(sg-local-example)sg(sg-example)ingress" = {
  traffic = "ingress"
  logs = true
  trace = true
  sg_local = "sg-local-example"
  sg = "sg-example"
  transport = "udp"
  ports = [{
  d = "80"
  s = ""
  }]
  action = "ACCEPT"
  priority = 0
  }
  }
}
	resource "sgroups_ie_rules" "rules" {
	  items = {
	  "udp:sg-local(sg-local-example)sg(sg-example)egress" = {
	  traffic = "egress"
	  logs = true
	  trace = true
	  sg_local = "sg-local-example"
	  sg = "sg-example"
	  transport = "udp"
	  ports = [{
	  d = "80"
	  s = ""
	  }]
	  action = "ACCEPT"
	  priority = 0
	  }
	  }
	}

ICMP4

Ingress	Egress

resource "sgroups_ie_icmp_rules" "rules" {
  items = {
  "icmp4:sg-local(sg-local-example)sg(sg-example)ingress" = {
  traffic = "ingress"
  logs = true
  trace = true
  sg_local = "sg-local-example"
  sg = "sg-example"
  ip_v = "IPv4"
  type = [0,8]
  action = "ACCEPT"
  priority = -100
  }
  }
}
	resource "sgroups_icmp_rules" "rules" {
	  items = {
	  "icmp4:sg-local(sg-local-example)sg(sg-example)egress" = {
	  traffic = "egress"
	  logs = true
	  trace = true
	  sg_local = "sg-local-example"
	  sg = "sg-example"
	  ip_v = "IPv4"
	  type = [0,8]
	  action = "ACCEPT"
	  priority = -100
	  }
	  }
	}

ICMP6

Ingress	Egress

resource "sgroups_ie_icmp_rules" "rules" {
  items = {
  "icmp6:sg-local(sg-local-example)sg(sg-example)ingress" = {
  traffic = "ingress"
  logs = true
  trace = true
  sg_local = "sg-local-example"
  sg = "sg-example"
  ip_v = "IPv6"
  type = [0,8]
  action = "ACCEPT"
  priority = -100
  }
  }
}
	resource "sgroups_icmp_rules" "rules" {
	  items = {
	  "icmp6:sg-local(sg-local-example)sg(sg-example)egress" = {
	  traffic = "egress"
	  logs = true
	  trace = true
	  sg_local = "sg-local-example"
	  sg = "sg-example"
	  ip_v = "IPv6"
	  type = [0,8]
	  action = "ACCEPT"
	  priority = -100
	  }
	  }
	}

API

Далее везде в документе подразумевается что все места, содержащие переменную $node, могут принять одно из двух значений: sgSgRules, ieSgSgIcmpRules.

Входные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].SG — Security Group, с которой устанавливаются правила взаимодействия.
- $node.rules[].sgLocal — Security Group относительно которой рассматриваются правила.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.
- syncOp — Поле определяющее действие с данными из запроса.

Области применения полей относительно используемого протокола

название	**обязательность**	**тип данных**	**значение по умолчанию**	**transport***		
				TCP	**UDP**	**ICMP**
$node.rules[]	да	Object[]	null	✔	✔	✔
$node.rules[].SG	да	String		✔	✔	✔
$node.rules[].sgLocal	да	String		✔	✔	✔
$node.rules[].logs	нет	Boolean	false	✔	✔	✔
$node.rules[].trace	нет	Boolean	false	✔	✔	✔
$node.rules[].ports	нет	Object[]	null	✔	✔	

$node.rules[].ports[].d	нет	String	null	✔	✔	
$node.rules[].ports[].s	нет	String	null	✔	✔	
$node.rules[].traffic	да	Enum("Ingress", "Egress")		✔	✔	✔
$node.rules[].transport	нет	Enum("TCP", "UDP")	TCP	✔	✔	
$node.rules[].ICMP	да	Object				✔
$node.rules[].ICMP.IPv	да	Enum("IPv4", "IPv6")				✔
$node.rules[].ICMP.Types	нет	String[]	[]			✔
$node.rules[].action	да	Enum("UNDEF", "ACCEPT", "DROP")		✔	✔	✔
$node.rules[].priority	нет	Object		✔	✔	✔
$node.rules[].priority.some	нет	Integer		✔	✔	✔
syncOp	да	Enum("Delete", "Upsert", "FullSync")		✔	✔	✔

Ограничения

sgSgRules.rules[].SG:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

sgSgRules.rules[].sgLocal:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

sgSgRules.rules[].ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

sgSgRules.rules[].ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.

- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

sgSgRules.rules[].ICMP.Types[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP

Ingress	Egress

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
"ieSgSgRules": {
"rules": [
{
"traffic": "Ingress",
"SG": "sg-example",
"sgLocal": "sg-local-example",
"logs": true,
"trace": true,
"ports": [{
"d": "64321",
"s": "443,80"
}],
"transport": "TCP",
"action": "ACCEPT",
"priority": {
"some": 0
}
}
]
},
"syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	"ieSgSgRules": {
	"rules": [
	{
	"traffic": "Egress",
	"SG": "sg-example",
	"sgLocal": "sg-local-example",
	"logs": true,
	"trace": true,
	"ports": [{
	"d": "64321",
	"s": "443,80"
	}],
	"transport": "TCP",
	"action": "ACCEPT",
	"priority": {
	"some": 0
	}
	}
	]
	},
	"syncOp": "Upsert"
	}'

UDP

Ingress	Egress

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieSgSgRules": {
  "rules": [
  {
  "traffic": "Ingress",
  "SG": "sg-example",
  "sgLocal": "sg-local-example",
  "logs": true,
  "trace": true,
  "ports": [{
  "d": "64321",
  "s": "443,80"
  }],
  "transport": "UDP",
  "action": "ACCEPT",
  "priority": {
  "some": 0
  }
  }
  ]
  },
  "syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	  "ieSgSgRules": {
	  "rules": [
	  {
	  "traffic": "Egress",
	  "SG": "sg-example",
	  "sgLocal": "sg-local-example",
	  "logs": true,
	  "trace": true,
	  "ports": [{
	  "d": "64321",
	  "s": "443,80"
	  }],
	  "transport": "UDP",
	  "action": "ACCEPT",
	  "priority": {
	  "some": 0
	  }
	  }
	  ]
	  },
	  "syncOp": "Upsert"
	}'

ICMP4

Ingress	Egress

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieSgSgIcmpRules": {
  "rules": [
  {
  "traffic": "Ingress",
  "SG": "sg-example",
  "sgLocal": "sg-local-example",
  "logs": true,
  "trace": true,
  "ICMP": {
  "IPv": "IPv4",
  "Types": [0, 8]
  },
  "action": "ACCEPT",
  "priority": {
  "some": -100
  }
  }
  ]
  },
  "syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	  "ieSgSgIcmpRules": {
	  "rules": [
	  {
	  "traffic": "Egress",
	  "SG": "sg-example",
	  "sgLocal": "sg-local-example",
	  "logs": true,
	  "trace": true,
	  "ICMP": {
	  "IPv": "IPv4",
	  "Types": [0, 8]
	  },
	  "action": "ACCEPT",
	  "priority": {
	  "some": -100
	  }
	  }
	  ]
	  },
	  "syncOp": "Upsert"
	}'

ICMP6

Ingress	Egress

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
"ieSgSgIcmpRules": {
"rules": [
{
"traffic": "Ingress",
"SG": "sg-example",
"sgLocal": "sg-local-example",
"logs": true,
"trace": true,
"ICMP": {
"IPv": "IPv6",
"Types": [0, 8]
},
"action": "ACCEPT",
"priority": {
"some": -100
}
}
]
},
"syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	"ieSgSgIcmpRules": {
	"rules": [
	{
	"traffic": "Egress",
	"SG": "sg-example",
	"sgLocal": "sg-local-example",
	"logs": true,
	"trace": true,
	"ICMP": {
	"IPv": "IPv6",
	"Types": [0, 8]
	},
	"action": "ACCEPT",
	"priority": {
	"some": -100
	}
	}
	]
	},
	"syncOp": "Upsert"
	}'

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

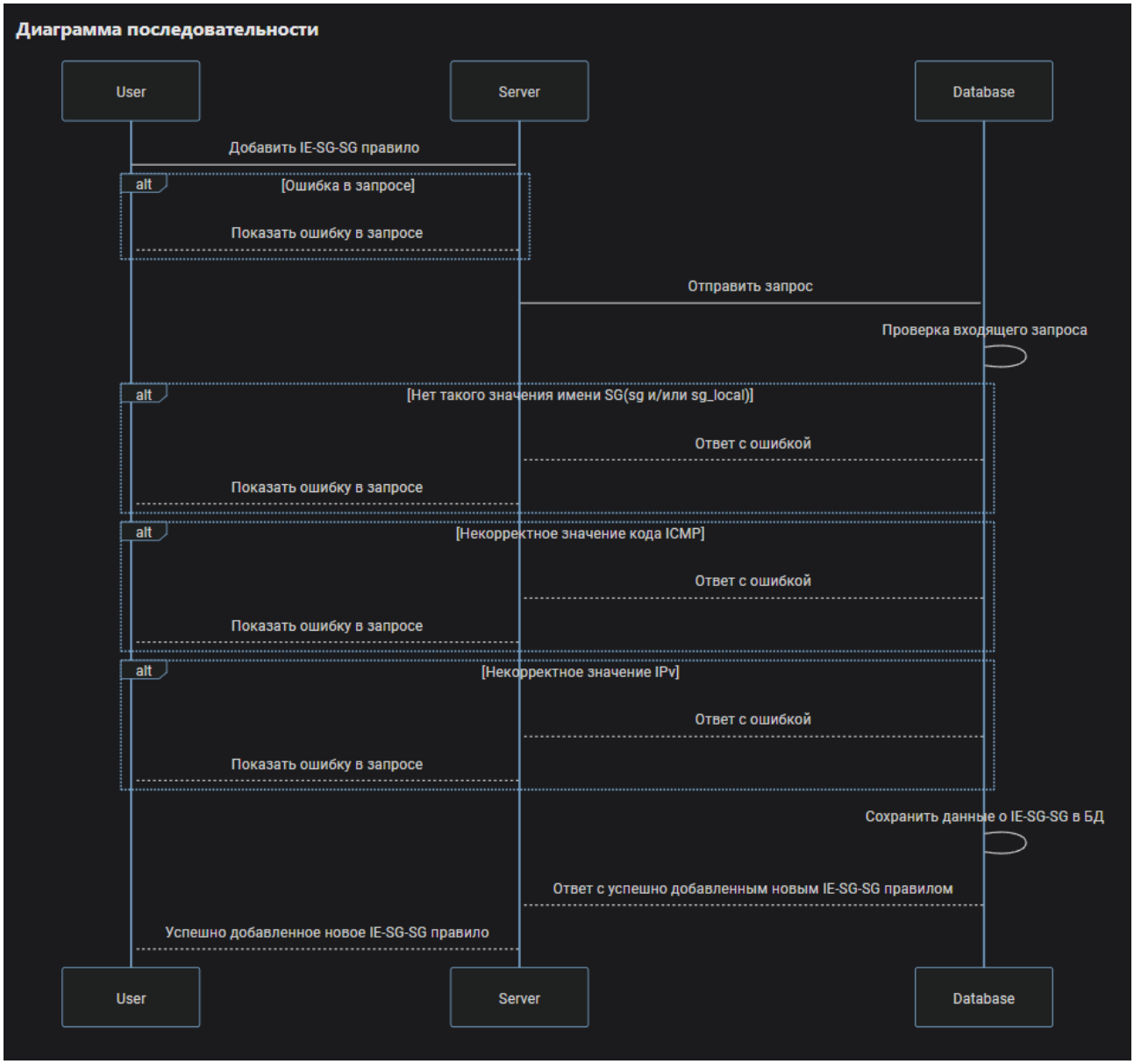
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

В этом разделе мы покажем, как правила фильтрации трафика, созданные с помощью Terraform и API, внедряются в настройки nftables. Это позволяет легко интегрировать сложные правила безопасности прямо в вашу систему фильтрации трафика.

- $Trace — Включить/отключить трассировку.
- $SrcSgroup — Security Group, с которой устанавливаются правила взаимодействия.

- $DstGroup — Security Group, с которой устанавливаются правила взаимодействия.
- $Transport — Протокол L3/L4 уровня модели OSI.
- $NftRuleType — Характеристика описывающая, что принимается трафик типа ip.
- $IcmpTypeList — Список, определяющий допустимые типы ICMP запросов.
- $SrcPorts — Набор открытых портов отправителя.
- $DstPorts — Набор открытых портов получателя
- $NftCounter — Счетчик количества байтов и пакетов.
- $Log — Включить/отключить логирование.
- $NftRuleVerdict — Результат применения правила, определяющий действие, которое будет применено к пакету.

Области применения полей относительно используемого протокола

Области применения полей относительно используемого протокола					
шаблон параметра	**структура параметра**	**значения**	**transport***		
			TCP	**UDP**	**ICMP**
$Trace	`nftrace set`	• **1** - трассировка включена • **0** - трассировка выключена	✔	✔	✔
$SrcSgroup	`saddr`	`@${IPSet(sgName)}`	✔	✔	✔
$DstSgroup	`daddr`	`@${IPSet(sgName)}`	✔	✔	✔
$Transport	`tcp | udp | icmp`		✔	✔	✔
$NftRuleType	`ip`		✔	✔	✔
$IcmpTypeList	`type {}`	Набор целочисленных значений от 0 до 255			✔
$SrcPorts	`sport {}`	Набор целочисленных значений от 1 до 65535	✔	✔	
$DstPorts	`dport {}`	Набор целочисленных значений от 1 до 65535	✔	✔	
$NftCounter	`counter`	`packets 0 bytes 0`	✔	✔	✔
$Log	`log`	`level debug flags ip options`	✔	✔	✔
$NftRuleVerdict	`accept`	*$NftRuleVerdict определяет действие, которое будет применено к пакету в соответсвии с правилом. Это поле может принимать значение* `accept` *или* `drop` *в зависимости от указанного в правиле.* *Подробнее: Verdict statement*	✔	✔	✔

Пример использования

TCP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $SrcSgroup $Transport $SrcPorts $DstPorts $NftCounter
$Log $NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
    nftrace set 1 ip saddr @NetIPv4-sg-example tcp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr @NetIPv6-sg-example tcp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstSgroup $Transport $SrcPorts $DstPorts $NftCounter
$Log $NftRuleVerdict
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr @NetIPv4-sg-example tcp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr @NetIPv6-sg-example tcp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

UDP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $SrcSgroup $Transport $SrcPorts $DstPorts $NftCounter
$Log $NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
    nftrace set 1 ip saddr @NetIPv4-sg-example udp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr @NetIPv6-sg-example udp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstSgroup $Transport $SrcPorts $DstPorts $NftCounter
$Log $NftRuleVerdict
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr @NetIPv4-sg-example udp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr @NetIPv6-sg-example udp dport { 80, 443 } sport {
64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

ICMP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $DstSgroup $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
    nftrace set 1 ip saddr @NetIPv4-sg-example icmp   type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr @NetIPv6-sg-example icmpv6 type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstSgroup $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr @NetIPv4-sg-example icmp   type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr @NetIPv6-sg-example icmpv6 type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

Sgroup to CIDR (I/E)

Ресурс Security Group to CIDR представляет собой введенную нами абстракцию, которая обеспечивает гибкое управление и контроль за сетевым трафиком между различными группами безопасности и подсетями, используя TCP, UDP и ICMP протоколы. Этот ресурс дает возможность детально настроить, какой трафик разрешен к передаче между группами безопасности и определенными подсетями, тем самым гарантируя высокий уровень защиты и управления сетевой инфраструктурой.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

Далее везде в документе подразумевается что все места, содержащие переменную $traffic, могут принять одно из двух значений: Ingress Egress. Аналогчично для $transport подразумевается одно из четырех значений: tcp, udp, icmpIPv4, icmpIPv6

- rules — Структура, содержащая описание создаваемых правил.
- rules.$traffic[] — Поле описывающий направление трафика.
- rules.$traffic[].cidrSet[] — Список, содержащий подсети типа IP.
- rules.$traffic[].access.$transport — Протокол L3/L4 уровня модели OSI.
- rules.$traffic[].access.$transport.action — Действие для пакетов в сформированных правил в цепочке.
- rules.$traffic[].access.$transport.priority — Поле определяющее порядок применения правил в цепочке.
- rules.$traffic[].access.$transport.logs — Включить/отключить логирование.
- rules.$traffic[].access.$transport.trace — Включить/отключить трассировку.
- rules.$traffic[].access.$transport.ports[].description — Формальное текстовое описание.
- rules.$traffic[].access.$transport.ports[].ports_to[] — Набор открытых портов получателя
- rules.$traffic[].access.$transport.ports[].ports_from[] — Набор открытых портов отправителя.
- rules.$traffic[].access.$transport.types[].description — Формальное текстовое описание.
- rules.$traffic[].access.$transport.types[].type[] — Список, определяющий допустимые типы ICMP запросов.

Области применения полей относительно используемого протокола

название параметра	тип данных	значение по умолчанию	transport*		
			TCP	UDP	ICMP
rules	Object[]	[]	✔	✔	✔
rules.$traffic[]	Object[]		✔	✔	✔

название параметра	тип данных	значение по умолчанию	transport*		
			TCP	UDP	ICMP
rules.$traffic[].cidrSet[]	String[]		✔	✔	✔
rules.$traffic[].access.$transport	Object		✔	✔	✔
rules.$traffic[].access.$transport.action	Enum("ACCEPT", "DROP")		✔	✔	✔
rules.$traffic[].access.$transport.priority	String		✔	✔	✔
rules.$traffic[].access.$transport.log	Boolean	false	✔	✔	✔
rules.$traffic[].access.$transport.trace	Boolean	false	✔	✔	✔
rules.$traffic[].access.$transport.ports[]	Object[]		✔	✔	✔
rules.$traffic[].access.$transport.ports[].description	String	""	✔	✔	
rules.$traffic[].access.$transport.ports[].ports_to[]	Integer[]	null	✔	✔	
rules.$traffic[].access.$transport.ports[].ports_from[]	Integer[]	null	✔	✔	
rules.$traffic[].access.$transport.types[]	Object[]				✔
rules.$traffic[].access.$transport.types[].description	String	""			✔
rules.$traffic[].access.$transport.types[].type[]	Integer[]	null			✔

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

rules.$traffic[].cidrSet:

- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

rules.$traffic[].access.$transport.priority:

- Значения должны находиться в интервале от -32768 до 32767

rules.$traffic[].access.$transport.ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

rules.$traffic[].access.$transport.ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

rules.$traffic[].access.$transport.types[].type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP

Ingress	Egress
name: sg-local-example rules: ingress: - cidrSet: - "10.0.0.0/8" access: tcp: action: ACCEPT priority: 300 logs: true trace: true ports: - description: "example" ports_from: - 64231 ports_to: - 443 - 80	name: sg-local-example rules: egress: - cidrSet: - "10.0.0.0/8" access: tcp: action: ACCEPT priority: 300 logs: true trace: true ports: - description: "example" ports_from: - 64231 ports_to: - 443 - 80

UDP

Ingress	Egress

name: sg-local-example
rules:
  ingress:
  - cidrSet:
  - "10.0.0.0/8"
  access:
  udp:
  action: ACCEPT
  priority: 300
  logs: true
  trace: true
  ports:
  - description: "example"
  ports_from:
  - 64231
  ports_to:
  - 443
  - 80
	name: sg-local-example
	rules:
	  egress:
	  - cidrSet:
	  - "10.0.0.0/8"
	  access:
	  udp:
	  action: ACCEPT
	  priority: 300
	  logs: true
	  trace: true
	  ports:
	  - description: "example"
	  ports_from:
	  - 64231
	  ports_to:
	  - 443
	  - 80

ICMP4

Ingress	Egress

name: sg-local-example
rules:
  ingress:
  - cidrSet:
  - "10.0.0.0/8"
  access:
  icmpIPv4:
  action: ACCEPT
  priority: 200
  logs: true
  trace: true
  types:
  - description: "example"
  type:
  - 0
  - 8
	name: sg-local-example
	rules:
	  egress:
	  - cidrSet:
	  - "10.0.0.0/8"
	  access:
	  icmpIPv4:
	  action: ACCEPT
	  priority: 200
	  logs: true
	  trace: true
	  types:
	  - description: "example"
	  type:
	  - 0
	  - 8

ICMP6

Ingress	Egress

name: sg-local-example
rules:
  ingress:
  - cidrSet:
  - "::ffff:a00:0/104"
  access:
  icmpIPv6:
  action: ACCEPT
  priority: 200
  logs: true
  trace: true
  types:
  - description: "example"
  type:
  - 0
  - 8
	name: sg-local-example
	rules:
	  egress:
	  - cidrSet:
	  - "::ffff:a00:0/104"
	  access:
	  icmpIPv6:
	  action: ACCEPT
	  priority: 200
	  logs: true
	  trace: true
	  types:
	  - description: "example"
	  type:
	  - 0
	  - 8

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

- items — список ресурсов создаваемые terraform ресурсом.
- items.$ruleName — уникальное имя создаваемого ресурса.
- items.$ruleName.traffic — Поле описывающий направление трафика.
- items.$ruleName.transport — Протокол L3/L4 уровня модели OSI.
- items.$ruleName.cidr — Список, содержащий подсети типа IP.
- items.$ruleName.sg_name — Security Group относительно которой рассматриваются правила.
- items.$ruleName.ports — Блок описывающий набор пар портов (src-dst).
- items.$ruleName.ports[].d — Набор открытых портов получателя
- items.$ruleName.ports[].s — Набор открытых портов отправителя.

- items.$ruleName.logs — Включить/отключить логирование.
- items.$ruleName.trace — Включить/отключить трассировку.
- items.$ruleName.ip_v — Версия IP для ICMP (IPv4 или IPv6).
- items.$ruleName.type — Список, определяющий допустимые типы ICMP запросов.
- items.$ruleName.action — Действие для пакетов в сформированных правил в цепочке.
- items.$ruleName.priority — Поле определяющее порядок применения правил в цепочке.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
items	да	Object		✔	✔	✔
items.$ruleName	да	Object		✔	✔	✔
items.$ruleName.traffic	да	Enum("ingress", "egress")		✔	✔	✔
items.$ruleName.transport	да	Enum("TCP", "UDP")		✔	✔	
items.$ruleName.cidr	да	String		✔	✔	✔
items.$ruleName.sg_name	да	String		✔	✔	✔
items.$ruleName.ports	нет	Object[]	null	✔	✔	
items.$ruleName.ports[].d	нет	String	""	✔	✔	
items.$ruleName.ports[].s	нет	String	""	✔	✔	
items.$ruleName.logs	нет	Boolean	false	✔	✔	✔
items.$ruleName.trace	нет	Boolean	false	✔	✔	✔
items.$ruleName.ip_v	да	Enum("IPv4", "IPv6")				✔
items.$ruleName.type	да	String[]	null			✔
items.$ruleName.action	да	Enum("ACCEPT", "DROP")		✔	✔	✔
items.$ruleName.priority	нет	Integer		✔	✔	✔

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.$ruleName:

- Форма ruleName должна быть организована в соответствии с определенной последовательностью, которую нужно соблюдать "${transport}:cidr(${cidr})sg(${sg_name})${traffic}".

items.$ruleName.sg_name:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.$ruleName.cidr:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

items.$ruleName.ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

items.$ruleName.ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

items.$ruleName.type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

items.$ruleName.priority:

- Значения должны находиться в интервале от -32768 до 32767

Пример использования

TCP

Ingress	Egress

resource "sgroups_cidr_rules" "rules" {
  items = {
  "tcp:cidr(10.0.0.0/8)sg(sg-example)ingress" = {
  traffic = "ingress"
  logs = true
  trace = true
  sg_name = "sg-local-example"
  cidr = "10.0.0.0/8"
  transport = "tcp"
  ports = [{
  d = "80"
  s = ""
  }]
  action = "ACCEPT"
  priority = 300
  }
  }
}
	resource "sgroups_cidr_rules" "rules" {
	  items = {
	  "tcp:cidr(10.0.0.0/8)sg(sg-example)egress" = {
	  traffic = "egress"
	  logs = true
	  trace = true
	  sg_name = "sg-local-example"
	  cidr = "10.0.0.0/8"
	  transport = "tcp"
	  ports = [{
	  d = "80"
	  s = ""
	  }]
	  action = "ACCEPT"
	  priority = 300
	  }
	  }
	}

UDP

Ingress	Egress

```
resource "sgroups_cidr_rules" "rules" {
  items = {
    "udp:cidr(10.0.0.0/8)sg(sg-example)ingress"
= {
      traffic     = "ingress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "10.0.0.0/8"
      transport   = "udp"
      ports = [{
        d = "80"
        s = ""
      }]
      action      = "ACCEPT"
      priority    = 300
    }
  }
}
```

```
resource "sgroups_cidr_rules" "rules" {
  items = {
    "udp:cidr(10.0.0.0/8)sg(sg-example)egress"
= {
      traffic     = "egress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "10.0.0.0/8"
      transport   = "udp"
      ports = [{
        d = "80"
        s = ""
      }]
      action      = "ACCEPT"
      priority    = 300
    }
  }
}
```

ICMP4

Ingress

```
resource "sgroups_cidr_icmp_rules" "rules" {
  items = {
    "icmp4:cidr(10.0.0.0/8)sg(sg-
example)ingress" = {
      traffic     = "ingress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "10.0.0.0/8"
      ip_v        = "IPv4"
      type        = [0,8]
      action      = "ACCEPT"
      priority    = 200
    }
  }
}
```

Egress

```
resource "sgroups_cidr_icmp_rules" "rules" {
  items = {
    "icmp4:cidr(10.0.0.0/8)sg(sg-
example)egress" = {
      traffic     = "egress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "10.0.0.0/8"
      ip_v        = "IPv4"
      type        = [0,8]
      action      = "ACCEPT"
      priority    = 200
    }
  }
}
```

ICMP6

Ingress

Egress

```
resource "sgroups_cidr_icmp_rules" "rules" {
  items = {
    "icmp6:cidr(::ffff:a00:0/104)sg(sg-example)ingress" = {
      traffic     = "ingress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "::ffff:a00:0/104"
      ip_v        = "IPv6"
      type        = [0,8]
      action      = "ACCEPT"
      priority    = 200
    }
  }
}
```

```
resource "sgroups_cidr_icmp_rules" "rules" {
  items = {
    "icmp6:cidr(::ffff:a00:0/104)sg(sg-example)egress" = {
      traffic     = "egress"
      logs        = true
      trace       = true
      sg_name     = "sg-local-example"
      cidr        = "::ffff:a00:0/104"
      ip_v        = "IPv6"
      type        = [0,8]
      action      = "ACCEPT"
      priority    = 200
    }
  }
}
```

API

Далее везде в документе подразумевается что все места, содержащие переменную $node, могут принять одно из двух значений: cidrSgRules cidrSgIcmpRules.

Входные параметры

- $node.rules[] — Структура, содержащая описание создаваемых правил.
- $node.rules[].CIDR — Список, содержащий подсети типа IP.
- $node.rules[].SG — Security Group относительно которой рассматриваются правила.
- $node.rules[].logs — Включить/отключить логирование.
- $node.rules[].trace — Включить/отключить трассировку.
- $node.rules[].ports — Блок описывающий набор пар портов (src-dst).
- $node.rules[].ports[].d — Набор открытых портов получателя
- $node.rules[].ports[].s — Набор открытых портов отправителя.
- $node.rules[].traffic — Поле описывающий направление трафика.
- $node.rules[].transport — Протокол L3/L4 уровня модели OSI.
- $node.rules[].ICMP — Структура, содержащая описание создаваемых правил типа ICMP.
- $node.rules[].ICMP.IPv — Версия IP для ICMP (IPv4 или IPv6).
- $node.rules[].ICMP.Types — Список, определяющий допустимые типы ICMP запросов.
- $node.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- $node.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.

- $node.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.
- syncOp — Поле определяющее действие с данными из запроса.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*		
				TCP	UDP	ICMP
$node.rules[]	да	Object	null	✔	✔	✔
$node.rules[].CIDR	да	String		✔	✔	✔
$node.rules[].SG	да	String		✔	✔	✔
$node.rules[].logs	нет	Boolean	false	✔	✔	✔
$node.rules[].trace	нет	Boolean	false	✔	✔	✔
$node.rules[].ports	нет	Object[]	null	✔	✔	
$node.rules[].ports[].d	нет	String	null	✔	✔	
$node.rules[].ports[].s	нет	String	null	✔	✔	
$node.rules[].traffic	да	Enum("Ingress", "Egress")		✔	✔	✔
$node.rules[].transport	нет	Enum("TCP", "UDP")	TCP	✔	✔	
$node.rules[].ICMP	да	Object				✔
$node.rules[].ICMP.IPv	да	Enum("IPv4", "IPv6")				✔
$node.rules[].ICMP.Types	нет	String[]	[]			✔
$node.rules[].action	да	Enum("UNDEF", "ACCEPT", "DROP")		✔	✔	✔
$node.rules[].priority	нет	Object		✔	✔	✔
$node.rules[].priority.some	нет	Integer		✔	✔	✔
syncOp	да	Enum("Delete", "Upsert", "FullSync")		✔	✔	✔

Ограничения

$node.rules[].SG:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

$node.rules[].CIDR:

- Значение поля должно начинаться и заканчиваться символами без пробелов.
- В пределах одной Security Group и направления трафика (I/E), необходимо обеспечить, непересекаемость диапазонов адресов подсетей.
- Подсеть должна соответствовать формату записи, определенному в RFC 4632.

$node.rules[].ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.

- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

$node.rules[].ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

$node.rules[].type[]:

- Значение должно быть числом в диапазоне от 0 до 255.
- Повторения значений в списке не допускаются.

Пример использования

TCP

Ingress	Egress

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgRules": {
    "rules": [
      {
        "traffic": "Ingress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ports": [{
          "d": "64321",
          "s": "443,80"
        }],
        "transport": "TCP",
        "action": "ACCEPT",
        "priority": {
          "some": 300
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgRules": {
    "rules": [
      {
        "traffic": "Egress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ports": [{
          "d": "64321",
          "s": "443,80"
        }],
        "transport": "TCP",
        "action": "ACCEPT",
        "priority": {
          "some": 300
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

UDP

Ingress	Egress

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgRules": {
    "rules": [
      {
        "traffic": "Ingress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ports": [{
          "d": "64321",
          "s": "443,80"
        }],
        "transport": "UDP",
        "action": "ACCEPT",
        "priority": {
          "some": 300
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgRules": {
    "rules": [
      {
        "traffic": "Egress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ports": [{
          "d": "64321",
          "s": "443,80"
        }],
        "transport": "UDP",
        "action": "ACCEPT",
        "priority": {
          "some": 300
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

ICMP4

Ingress	Egress

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgIcmpRules": {
    "rules": [
      {
        "traffic": "Ingress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ICMP": {
          "IPv": "IPv4",
          "Types": [0, 8]
        },
        "action": "ACCEPT",
        "priority": {
          "some": 200
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

```
curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgIcmpRules": {
    "rules": [
      {
        "traffic": "Egress",
        "CIDR": "10.0.0.0/8",
        "SG": "sg-local-example",
        "logs": true,
        "trace": true,
        "ICMP": {
          "IPv": "IPv4",
          "Types": [0, 8]
        },
        "action": "ACCEPT",
        "priority": {
          "some": 200
        }
      }
    ]
  },
  "syncOp": "Upsert"
}'
```

ICMP6

Ingress	Egress

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
  "ieCidrSgIcmpRules": {
  "rules": [
  {
  "traffic": "Ingress",
  "CIDR": "::ffff:a00:0/104",
  "SG": "sg-local-example",
  "logs": true,
  "trace": true,
  "ICMP": {
  "IPv": "IPv6",
  "Types": [0, 8]
  },
  "action": "ACCEPT",
  "priority": {
  "some": 200
  }
  }
  ]
  },
  "syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	  "ieCidrSgIcmpRules": {
	  "rules": [
	  {
	  "traffic": "Egress",
	  "CIDR": "::ffff:a00:0/104",
	  "SG": "sg-local-example",
	  "logs": true,
	  "trace": true,
	  "ICMP": {
	  "IPv": "IPv6",
	  "Types": [0, 8]
	  },
	  "action": "ACCEPT",
	  "priority": {
	  "some": 200
	  }
	  }
	  ]
	  },
	  "syncOp": "Upsert"
	}'

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

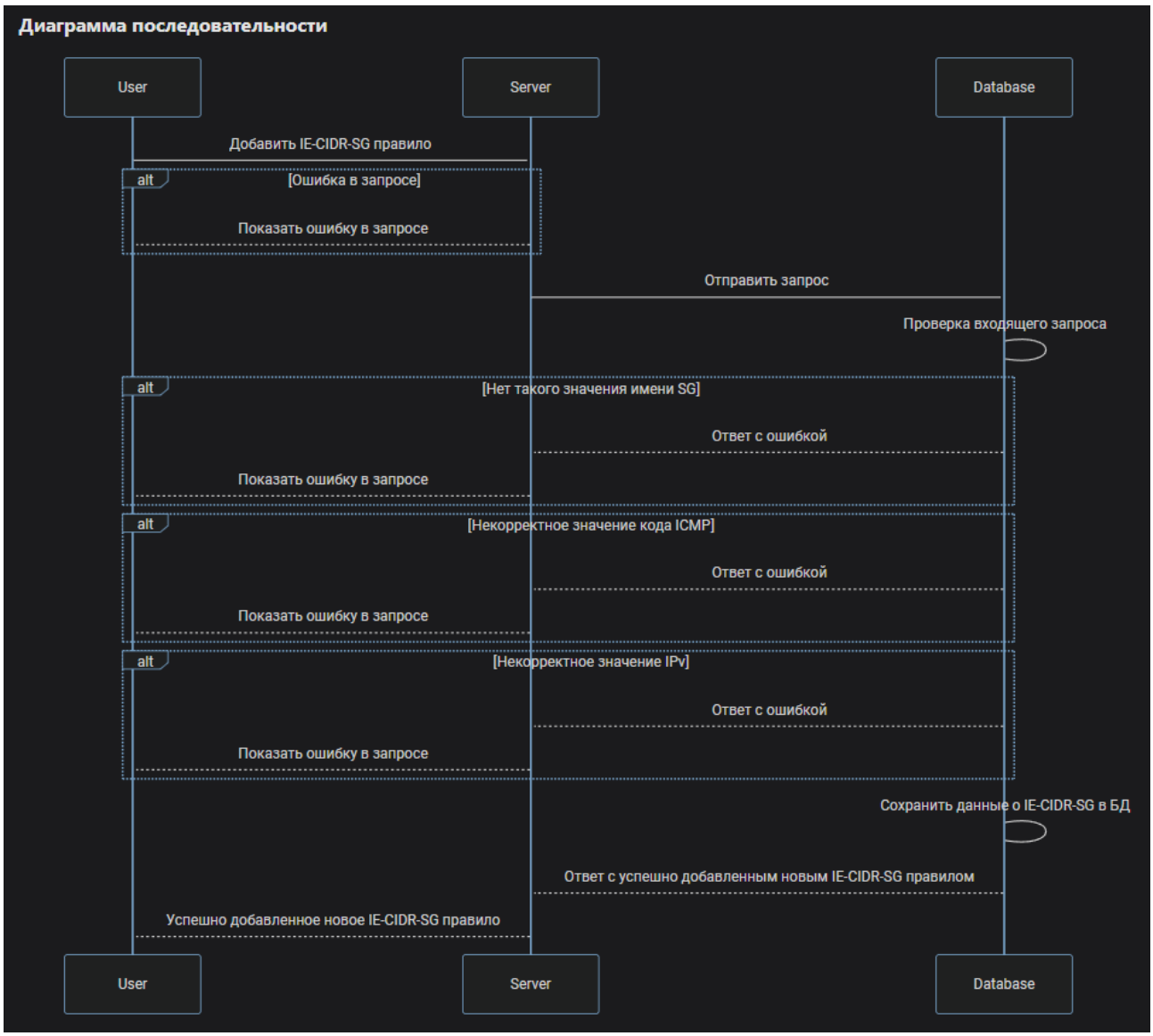
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

этом разделе мы покажем, как правила фильтрации трафика, созданные с помощью Terraform и API, внедряются в настройки nftables. Это позволяет легко интегрировать сложные правила безопасности прямо в вашу систему фильтрации трафика.

- $Trace — Включить/отключить трассировку.
- $SrcCIDR — CIDR, с которой устанавливаются правила взаимодействия.
- $DstCIDR — CIDR, с которой устанавливаются правила взаимодействия.
- $Transport — Протокол L3/L4 уровня модели OSI.
- $NftRuleType — Характеристика описывающая, что принимается трафик типа ip.
- $IcmpTypeList — Список, определяющий допустимые типы ICMP запросов.
- $SrcPorts — Набор открытых портов отправителя.
- $DstPorts — Набор открытых портов получателя

- $NftCounter — Счетчик количества байтов и пакетов.
- $Log — Включить/отключить логирование.
- $NftRuleVerdict — Результат применения правила, определяющий действие, которое будет применено к пакету.

Области применения полей относительно используемого протокола

шаблон параметра	структура параметра	значения	transport*		
			TCP	UDP	ICMP
$Trace	`nftrace set`	• **1** — трассировка включена • **0** — трассировка выключена	✔	✔	✔
$SrcCIDR	`saddr`	`${CIDR}`	✔	✔	✔
$DstCIDR	`daddr`	`${CIDR}`	✔	✔	✔
$Transport	Enum("tcp", "udp", "icmp")		✔	✔	✔
$NftRuleType	`ip`		✔	✔	✔
$IcmpTypeList	`type {}`	Набор целочисленных значений от 0 до 255			✔
$SrcPorts	`sport {}`	Набор целочисленных значений от 0 до 65535	✔	✔	
$DstPorts	`dport {}`	Набор целочисленных значений от 0 до 65535	✔	✔	
$NftCounter	`counter`	`packets 0 bytes 0`	✔	✔	✔
$Log	`log`	`level debug flags ip options`	✔	✔	✔
$NftRuleVerdict	`accept`	*$NftRuleVerdict определяет действие, которое будет применено к пакету в соответсвии с правилом. Это поле может принимать значение `accept` или `drop` в зависимости от указанного в правиле.* *Подробнее: Verdict statement*	✔	✔	✔

Пример использования

TCP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $SrcCIDR $Transport $SrcPorts $DstPorts $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
```

```
    nftrace set 1 ip saddr { 10.0.0.0/8 } tcp dport { 80, 443 } sport { 64231 }
counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstCIDR $Transport $SrcPorts $DstPorts $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr { 10.0.0.0/8 } tcp dport { 80, 443 } sport { 64231 }
counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

UDP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $SrcCIDR $Transport $SrcPorts $DstPorts $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
    nftrace set 1 ip saddr { 10.0.0.0/8 } udp dport { 80, 443 } sport { 64231 }
counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstCIDR $Transport $SrcPorts $DstPorts $NftCounter $Log
$NftRuleVerdict
    # **********
```

```
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr { 10.0.0.0/8 } udp dport { 80, 443 } sport { 64231 }
counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

ICMP

ingress

Шаблон

```
chain INGRESS-INPUT-sgName {
    # **********
    $Trace $NftRuleType $DstCIDR $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    $Trace $NftRuleType $DstCIDR $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain INGRESS-INPUT-sg-local-example {
    # **********
    nftrace set 1 ip saddr { 10.0.0.0/8 }       icmp  type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip saddr { ::ffff:a00:0/104 } icmp6 type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

egress

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    $Trace $NftRuleType $DstCIDR $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    $Trace $NftRuleType $DstCIDR $Transport $IcmpTypeList $NftCounter $Log
$NftRuleVerdict
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr { 10.0.0.0/8 }       icmp  type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr { ::ffff:a00:0/104 } icmp6 type { 0, 8 } counter
packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

Sgroup to FQDN (E)

Ресурс Security Group to FQDN представляет собой введенную нами абстракцию, которая обеспечивает гибкое управление и контроль за сетевым трафиком между различными группами безопасности и IP адресами стоящие за FQDN записями, используя TCP, UDP протоколы. Этот ресурс дает возможность детально настроить, какой трафик разрешен к передаче между группами безопасности и IP адресами стоящие за определенными FQDN записями, тем самым гарантируя высокий уровень защиты и управления сетевой инфраструктурой.

Terraform module

Terraform module представляет собой высокоуровневую абстракцию над terraform resources, которая упрощает работу с ресурсами Terraform, скрывая сложность их непосредственного использования. Он предлагает простой и понятный интерфейс для взаимодействия, позволяя пользователям легко интегрироваться и управлять компонентами инфраструктуры без необходимости глубоко погружаться в детали каждого ресурса.

Далее везде в документе подразумевается что все места, содержащие переменную $traffic, могут принять значение: Egress. Аналогчично для $transport подразумевается одно из двух значений: tcp, udp.

- rules — Структура, содержащая описание создаваемых правил.
- rules.$traffic[] — Поле описывающий направление трафика.
- rules.$traffic[].fqdnSet[] — Список, содержащий FQDN записи.
- rules.$traffic[].access.$transport — Протокол L3/L4 уровня модели OSI.
- rules.$traffic[].access.$transport.action — Действие для пакетов в сформированных правил в цепочке.
- rules.$traffic[].access.$transport.priority — Поле определяющее порядок применения правил в цепочке.
- rules.$traffic[].access.$transport.log — Включить/отключить логирование.
- rules.$traffic[].access.$transport.trace — Включить/отключить трассировку.
- rules.$traffic[].access.$transport.ports[].description — Формальное текстовое описание.
- rules.$traffic[].access.$transport.ports[].ports_to[] — Набор открытых портов получателя
- rules.$traffic[].access.$transport.ports[].ports_from[] — Набор открытых портов отправителя.

Области применения полей относительно используемого протокола

название параметра	тип данных	значение по умолчанию	transport*	
			TCP	UDP
rules	Object[]	[]	✔	✔
rules.$traffic[]	Object[]		✔	✔
rules.$traffic[].fqdnSet[]	String[]		✔	✔
rules.$traffic[].access.$transport	Object		✔	✔
rules.$traffic[].access.$transport.action	Enum("ACCEPT", "DROP")		✔	✔
rules.$traffic[].access.$transport.priority	String		✔	✔

название параметра	тип данных	значение по умолчанию	transport*	
			TCP	UDP
rules.$traffic[].access.$transport.log	Boolean	false	✔	✔
rules.$traffic[].access.$transport.trace	Boolean	false	✔	✔
rules.$traffic[].access.$transport.ports[]	Object[]		✔	✔
rules.$traffic[].access.$transport.ports[].description	String	""	✔	✔
rules.$traffic[].access.$transport.ports[].ports_to[]	Integer[]	null	✔	✔
rules.$traffic[].access.$transport.ports[].ports_from[]	Integer[]	null	✔	✔

Ограничения

name:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

rules.$traffic[].fqdnSet:

- Длина значения элемента не должна превышать 256 символов.
- Значение элемента должно начинаться и заканчиваться символами без пробелов.
- Повторения значений в списке не допускаются.
- Необходимо указать минимум одно значение.
- FQDN должны соответствовать формату записи, определенному в RFC 1034, 1035, 1123.

rules.$traffic[].access.$transport.priority:

- Значения должны находиться в интервале от -32768 до 32767

rules.$traffic[].access.$transport.ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

rules.$traffic[].access.$transport.ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security

Group.

Пример использования

TCP	UDP

name: sg-local-example
rules:
  egress:
  - fqdnSet:
  - example.com
  access:
  tcp:
  action: ACCEPT
  priority: 100
  logs: true
  trace: true
  ports:
  - description: "example"
  ports_from:
  - 64231
  ports_to:
  - 443
  - 80
	name: sg-local-example
	rules:
	  egress:
	  - fqdnSet:
	  - example.com
	  access:
	  udp:
	  action: ACCEPT
	  priority: 100
	  logs: true
	  trace: true
	  ports:
	  - description: "example"
	  ports_from:
	  - 64231
	  ports_to:
	  - 443
	  - 80

Terraform resource

Terraform resource является ключевым элементом в Terraform, предназначенным для управления различными аспектами инфраструктуры через код. Он позволяет задавать, настраивать и управлять инфраструктурными компонентами без привязки к их конкретным типам, обеспечивая автоматизацию развертывания и поддержки инфраструктуры согласно подходу Infrastructure as Code (IaC).

- items — список ресурсов создаваемые terraform ресурсом.
- items.$ruleName — уникальное имя создаваемого ресурса.
- items.$ruleName.transport — Протокол L3/L4 уровня модели OSI.
- items.$ruleName.protocols[] — Список протоколов L7 уровня модели OSI.
- items.$ruleName.logs — Включить/отключить логирование.
- items.$ruleName.trace — Включить/отключить трассировку.
- items.$ruleName.sg_from — Security Group относительно которой рассматриваются правила.
- items.$ruleName.fqdn — Полное доменное имя (FQDN), для которого применяется данное правило.
- items.$ruleName.ports[] — Блок описывающий набор пар портов (src-dst).
- items.$ruleName.ports[].s — Набор открытых портов отправителя.
- items.$ruleName.ports[].d — Набор открытых портов получателя
- items.$ruleName.action — Действие для пакетов в сформированных правил в цепочке.

- items.$ruleName.priority — Поле определяющее порядок применения правил в цепочке.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*	
				TCP	UDP
items	да	Object		✔	✔
items.$ruleName	да	Object		✔	✔
items.$ruleName.transport	да	Enum("TCP", "UDP")		✔	✔
items.$ruleName.logs	нет	Boolean	false	✔	✔
items.$ruleName.trace	нет	Boolean	false	✔	✔
items.$ruleName.sg_from	да	String		✔	✔
items.$ruleName.fqdn	да	String		✔	✔
items.$ruleName.ports[]	нет	Object[]	null	✔	✔
items.$ruleName.ports[].s	нет	String	""	✔	✔
items.$ruleName.ports[].d	нет	String	""	✔	✔
items.$ruleName.action	да	Enum("ACCEPT", "DROP")		✔	✔
items.$ruleName.priority	нет	Integer		✔	✔

Ограничения

items:

- Каждое правило должно обладать уникальным ключом для предотвращения конфликтов.

items.$ruleName:

- Форма ruleName должна быть организована в соответствии с определенной последовательностью, которую нужно соблюдать "${transport}:sg(${sg_from})fqdn(${fqdn})".

items.$ruleName.protocols[]:

- Не допускаются дубликация протоколов в списке.
- Значения должны соответствовать поддерживаемым протоколам.

items.$ruleName.sg_from:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

items.$ruleName.fqdn:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- FQDN должен соответствовать формату записи, определенному в RFC 1034, 1035, 1123.

items.$ruleName.ports[].s:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

items.$ruleName.ports[].d:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

items.$ruleName.priority:

- Значения должны находиться в интервале от -32768 до 32767

Пример использования

TCP	UDP

resource "sgroups_fqdn_rules" "rules" {
  items = {
  "tcp:sg(sg-local-example)fqdn(example.com)" = {
  transport = "tcp"
  sg_from = "sg-local-example"
  fqdn = "example.com"
  ports = [{
  d = "443,80"
  s = "64231"
  }]
  logs = true
  trace = true
  action = "ACCEPT"
  priority = 100
  }
  }
}
	resource "sgroups_fqdn_rules" "rules" {
	  items = {
	  "udp:sg(sg-local-example)fqdn(example.com)" = {
	  transport = "udp"
	  sg_from = "sg-local-example"
	  fqdn = "example.com"
	  ports = [{
	  d = "443,80"
	  s = "64231"
	  }]
	  logs = true
	  trace = true
	  action = "ACCEPT"
	  priority = 100
	  }
	  }
	}

API

Входные параметры

- fqdnRules.rules — Структура, содержащая описание создаваемых правил.
- fqdnRules.rules[].FQDN — Полное доменное имя (FQDN), для которого применяется данное правило.
- fqdnRules.rules[].logs — Включить/отключить логирование.
- fqdnRules.rules[].ports — Включить/отключить трассировку.
- fqdnRules.rules[].ports[].d — Набор открытых портов получателя
- fqdnRules.rules[].ports[].s — Набор открытых портов отправителя.
- fqdnRules.rules[].sgFrom — Security Group относительно которой рассматриваются правила.
- fqdnRules.rules[].transport — Протокол L3/L4 уровня модели OSI.
- fqdnRules.rules[].action — Действие для пакетов в сформированных правил в цепочке.
- fqdnRules.rules[].priority — Структура, содержащая описание порядка применения правил в цепочке.
- fqdnRules.rules[].priority.some — Поле определяющее порядок применения правил в цепочке.
- syncOp — Поле определяющее действие с данными из запроса.

Области применения полей относительно используемого протокола

название	обязательность	тип данных	значение по умолчанию	transport*	
				TCP	UDP
fqdnRules.rules	да	Object		✔	✔
fqdnRules.rules[].FQDN	да	String		✔	✔
fqdnRules.rules[].logs	нет	Boolean	false	✔	✔
fqdnRules.rules[].ports	нет	Object[]	null	✔	✔
fqdnRules.rules[].ports[].d	нет	String	null	✔	✔
fqdnRules.rules[].ports[].s	нет	String	null	✔	✔
fqdnRules.rules[].sgFrom	да	String		✔	✔
fqdnRules.rules[].transport	да	Enum("TCP", "UDP")		✔	✔
fqdnRules.rules[].action	да	Enum("UNDEF", "ACCEPT", "DROP")		✔	✔
fqdnRules.rules[].priority	нет	Object		✔	✔
fqdnRules.rules[].priority.some	нет	Integer		✔	✔
syncOp	да	Enum("Delete", "Upsert", "FullSync")		✔	✔

Ограничения

fqdnRules.rules[].FQDN:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- FQDN должен соответствовать формату записи, определенному в RFC 1034, 1035, 1123.

fqdnRules.rules[].sgFrom:

- Длина значения поля не должна превышать 256 символов.
- Значение поля должно начинаться и заканчиваться символами без пробелов.
- Значение должно быть уникальным в рамках типа ресурса.

fqdnRules.rules[].protocols[]:

- Не допускаются дубликация протоколов в списке.
- Значения должны соответствовать поддерживаемым протоколам.

fqdnRules.rules[].ports[].ports_to[]:

- Значения портов должно находиться в интервале от 1 до 65535.

- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.
- Не допускаются пересечения портов в правилах в рамках одной пары Security Group.

fqdnRules.rules[].ports[].ports_from[]:

- Значения портов должно находиться в интервале от 1 до 65535.
- Если значение не будет указано то будет использоваться весь диапазон портов.
- Значения портов прописываются по одному или интервально используя '-'.

Пример использования

TCP	UDP

curl '127.0.0.1:9007/v2/sync' \
--header 'Content-Type: application/json' \
--data '{
"fqdnRules": {
"rules": [{
"FQDN": "example.com",
"logs": true,
"ports": [{
"d": "443,80",
"s": "64231"
}],
"sgFrom": "sg-local-example",
"transport": "TCP",
"action": "ACCEPT",
"priority": {
"some": 100
}
}]
},
"syncOp": "Upsert"
}'
	curl '127.0.0.1:9007/v2/sync' \
	--header 'Content-Type: application/json' \
	--data '{
	"fqdnRules": {
	"rules": [{
	"FQDN": "example.com",
	"logs": true,
	"ports": [{
	"d": "443,80",
	"s": "64231"
	}],
	"sgFrom": "sg-local-example",
	"transport": "UDP",
	"action": "ACCEPT",
	"priority": {
	"some": 100
	}
	}]
	},
	"syncOp": "Upsert"
	}'

Выходные параметры

название	тип данных	описание
-	Object	в случае успеха возвращается пустое тело

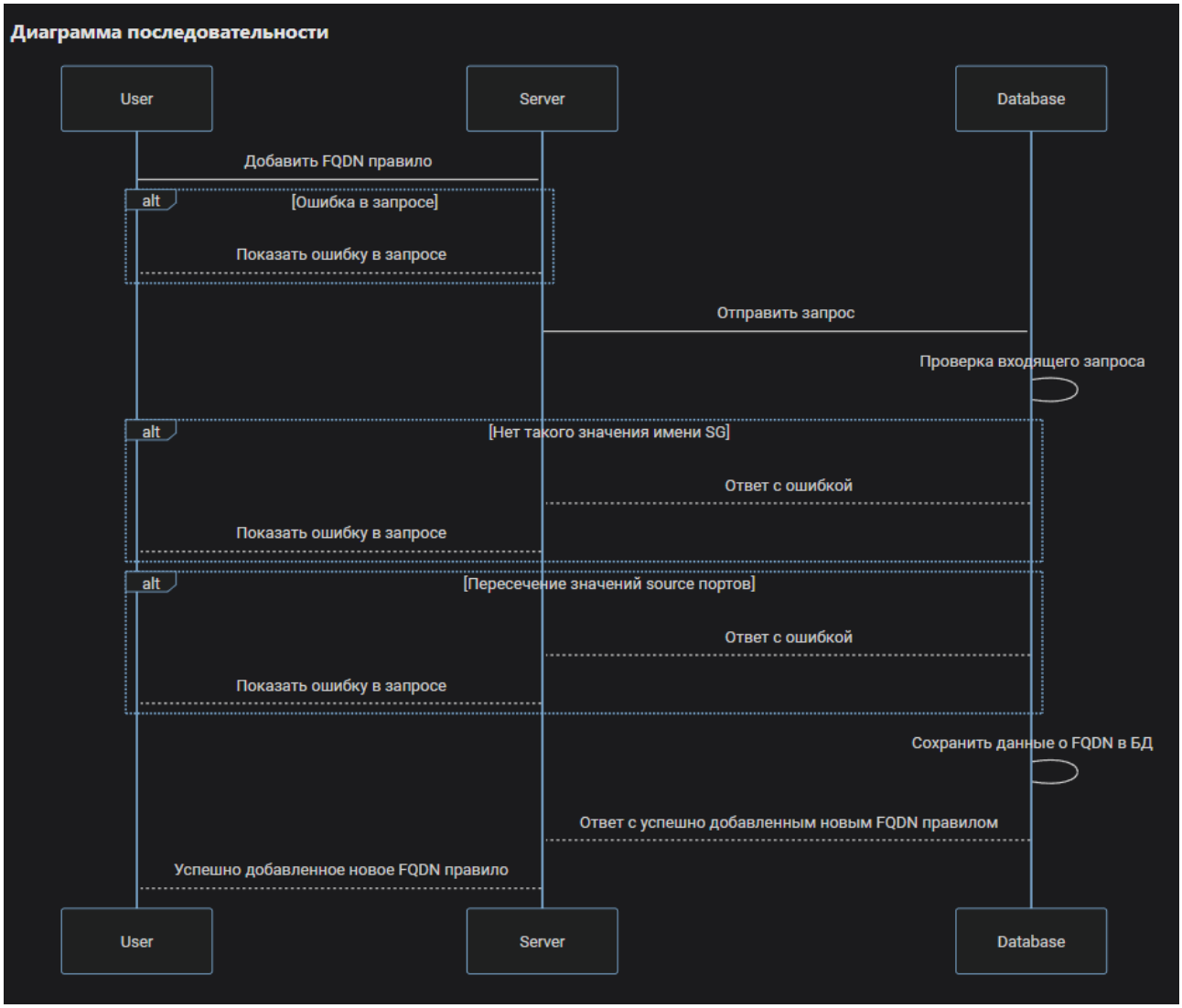
Возможные ошибки API

Пользователь указал некорректные значения агрументов

- HTTP code: 400
- gRPC code: INVALID_ARGUMENT
- gRPC number: 3

Не найден метод

- HTTP code: 404
- gRPC code: NOT_FOUND
- gRPC number: 5



nftables

В этом разделе мы покажем, как правила фильтрации трафика, созданные с помощью Terraform и API, внедряются в настройки nftables. Это позволяет легко интегрировать сложные правила безопасности прямо в вашу систему фильтрации трафика.

Области применения полей относительно используемого протокола

шаблон параметра	структура параметра	значение	описание
${Trace}	`nftrace set`	• **1** — трассировка включена • **0** — трассировка выключена	Трассировка указанного правила (опциональна, можно включить/выключить)
${DstFQDN}	`saddr @${IPSet(sgName)}`	Наименование IPSet в котором описаны сети в FQDN	Значение типа string, не должно содержать в себе пробелов
${Transport}	`tcp | udp`	протокол передачи данных в цепочке правил.	Одно из двух значений `tcp | udp`
${RuleType}	`ip`		Описывает, что принимает трафик типа ip
${SrcPorts}	`sport {}`	Набор целочисленных значений от 0 до 655353	Значения `sport` (source port). Может быть как одно значение, как и множество значений портов. В случае если одно значение у порта то передается значение либо как целочисленное значение либо как название порта. Если передается массив значений портов то они должны быть внутри `{}` перечислены через запятую.
${DstPorts}	`dport {}`	Набор целочисленных значений от 0 до 655353	Значения `dport` (destination port). Может быть как одно значение, как и множество значений портов. В случае если одно значение у порта то передается значение либо как целочисленное значение либо как название порта. Если передается массив значений портов то они должны быть внутри `{}` перечислены через запятую.
${Counter}	`counter packets 0 bytes 0`	Не параметризированный	Счетчик, учитывает количество пройденных пакетов с количеством байтов переданной информации в рамках указанной цепочки правил
${Log}	`log level debug`	Не параметризированный	Логирование указанного

шаблон параметра	структура параметра	значение	описание
	`flags ip options`		правила (опциональна, можно включить/выключить)
${Verdict}	`accept`	Не параметризированный *$Verdict определяет действие, которое будет применено к пакету в соответсвии с правилом. Это поле может принимать значение `accept` или `drop` в зависимости от указанного в правиле.* *Подробнее: Verdict statement*	Вердикт политики по пакетам данных

Шаблон

```
chain EGRESS-POSTROUTING-sgName {
    # **********
    ${Trace} ${RuleType} ${DstFQDN} ${Transport} ${SrcPorts} ${DstPorts}
${Counter} ${Log} ${Verdict}
    # **********
}
```

Пример использования

```
chain EGRESS-POSTROUTING-sg-local-example {
    # **********
    nftrace set 1 ip daddr @NetIPv4-fqdn-example.com tcp dport { 80, 443 } sport
{ 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr @NetIPv6-fqdn-example.com tcp dport { 80, 443 } sport
{ 64231 } counter packets 0 bytes 0 log level debug flags ip options accept

    nftrace set 1 ip daddr @NetIPv4-fqdn-example.com udp dport { 80, 443 } sport
{ 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    nftrace set 1 ip daddr @NetIPv6-fqdn-example.com udp dport { 80, 443 } sport
{ 64231 } counter packets 0 bytes 0 log level debug flags ip options accept
    # **********
}
```

Обращение в Службу технической поддержки

Если при работе с ПО у вас возникли проблемы или вопросы – свяжитесь со службой технической поддержки по электронной почте info@prorobotech.ru или с помощью Телеграм: https://t.me/sgroups_support.