
















Программное обеспечение «Платформа балансировки сетевого трафика flxGATE (Флексигейт)»


**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЕЙ С РОЛЯМИ «АДМИНИСТРАТОР», «ПОЛЬЗОВАТЕЛЬ»,
«АУДИТОР»**

Содержание


1	Общая архитектура решения	11
1.1	Схема решения	11
1.2	Функциональность	11
1.2.1	Control Plane	11
1.2.2	Data Plane	12
2	Control Plane	13
2.1	Компоненты	13
2.1.1	API Controller	13
2.1.2	Discovery Service	13
2.1.3	Storage	13
2.1.4	CLI	13
3	Data Plane	14
3.1	Компонентная схема	14
3.1.1	Traffic Proxy	14
3.1.1.1	Traffic Processor	14
3.1.1.2	Service Discovery	15
3.1.1.3	Logging	15
3.1.1.4	Security	15
3.1.1.5	Metrics	15
3.1.1.6	Request Routing	15
3.1.2	Network routing	15
3.1.2.1	BGP/OSPF/VRRP routing	15
3.1.2.2	Upstream Tracker	15
4	Установка	16
4.1	 Варианты поставки	16
4.2	 Операционные системы	16
4.3	Развертывание образа OVA	16
4.3.1	 Предварительные требования	16
4.3.2	 Импорт OVA образа	16
	1 Подключение к ESXi	16
	2 Запуск мастера импорта	16
	3 Импорт образа	16
	4 Создание виртуальной машины	16
	5 Настройка сетевого адаптера VM	16
	6 Запуск системы	16
	7 Настройка сети системы	17
4.4	Развертывание образа qcow2	17

4.4.1	 Предварительные требования	17
4.4.2	 Установка образа.....	17
1	Создание виртуальной машины	17
2	Настройка сети системы	17
5	Управление.....	18
5.1	 Command Line Interface (CLI)	18
5.2	 Запуск CLI.....	18
1	Подключение к серверу	18
2	Авторизация.....	18
3	Начало работы.....	18
5.3	 Управление настройками сети.....	18
5.3.1	Настройка устройств.....	19
5.3.1.1	Получение списка устройств	19
5.3.1.2	Просмотр информации об устройстве	19
5.3.1.3	Отключение устройства	19
5.3.1.4	Подключение устройства	19
5.3.2	Настройка соединений.....	19
5.3.2.1	Получение списка соединений.....	19
5.3.2.2	Создание соединения.....	20
5.3.2.3	Просмотр информации о подключении.....	20
5.3.2.4	Изменение соединения.....	21
5.3.2.5	Удаление соединения.....	21
5.3.2.6	Отключение соединения	21
5.3.2.7	Включение соединения.....	21
5.4	 Управление узлами.....	22
5.4.1	Получение списка узлов	22
5.4.2	Создание узла.....	22
5.4.2.1	Обработка ошибок при создании узла.....	22
5.4.3	Просмотр информации об узле	22
5.4.4	Управление конфигурацией узла	22
5.4.4.1	Включение режима конфигурации.....	23
5.4.4.2	Выключение режима конфигурации	23
5.4.4.3	Публикация конфигурации.....	23
5.4.5	Удаление узла	23
5.4.6	Примеры использования	23
5.4.6.1	Пример 1. Создание и настройка нового узла.....	23
5.4.6.2	Пример 2. Обработка ошибок	24

5.5	 Управление upstream.....	24
5.5.1	Получение списка upstream кластеров	25
5.5.2	Создание upstream кластера.....	25
5.5.2.1	Параметры создания upstream кластера	25
5.5.2.2	Примеры создания upstream кластера	25
5.5.2.3	Обработка ошибок при создании upstream кластера	26
5.5.3	Просмотр информации об upstream кластера	26
5.5.3.1	Просмотр полной конфигурации upstream кластера	26
5.5.4	Настройка атрибутов upstream кластера	26
5.5.4.1	Таймаут подключения (connect_timeout)	26
5.5.4.2	Тип обнаружения сервисов (type).....	27
5.5.4.3	Параметры подключения (connection_options)	27
5.5.5	Удаление upstream кластера	27
5.5.6	Доступные команды в меню cluster	28
5.5.6.1	Команды в меню списка upstream	28
5.5.6.2	Команды в меню конкретного cluster	28
5.6	 Управление Circuit Breakers	28
5.6.1	Доступ к меню Circuit Breakers.....	28
5.6.2	Просмотр конфигурации	28
5.6.2.1	Просмотр полной конфигурации circuit breakers	28
5.6.3	Управление Thresholds (порогами)	29
5.6.3.1	Просмотр конфигурации thresholds.....	29
5.6.4	Приоритеты Thresholds.....	29
5.6.4.1	Настройка Default Priority	29
5.6.4.2	Настройка High Priority	31
5.6.4.3	Значения по умолчанию.....	31
5.6.5	Доступные команды	31
5.6.5.1	Команды в меню circuit_breakers	31
5.6.5.2	Команды в меню thresholds	32
5.6.5.3	Команды в меню default_priority / high_priority	32
5.7	 Управление Health Checks	32
5.7.1	Доступ к меню Health Checks	32
5.7.2	Просмотр конфигурации	33
5.7.2.1	Просмотр полной конфигурации health checks	33
5.7.3	Создание Health Check.....	33
5.7.3.1	Типы Health Checks	33
5.7.3.2	Параметры команды --add	33
5.7.3.3	Значения по умолчанию.....	33

5.7.4	Управление Health Check.....	34
5.7.4.1	Просмотр конфигурации Health Check.....	34
5.7.4.2	Удаление Health Check.....	34
5.7.5	Общие параметры Health Check	34
5.7.5.1	timeout.....	34
5.7.5.2	interval.....	35
5.7.5.3	unhealthy_threshold.....	35
5.7.5.4	healthy_threshold.....	35
5.7.6	HTTP Health Check.....	35
5.7.6.1	host.....	35
5.7.6.2	path.....	35
5.7.6.3	method.....	35
5.7.6.4	expected_statuses	36
5.7.6.5	send	36
5.7.6.6	receive.....	36
5.7.6.7	request_headers_to_add	36
5.7.6.8	request_headers_to_remove.....	37
5.7.6.9	response_buffer_size	37
5.7.7	TCP Health Check.....	37
5.7.7.1	send	37
5.7.7.2	receive.....	37
5.7.8	gRPC Health Check.....	37
5.7.8.1	service_name	37
5.7.8.2	authority.....	38
5.7.9	Доступные команды	38
5.7.9.1	Команды в меню health_checks.....	38
5.7.9.2	Команды в меню health check (item_N).....	38
5.8	 Управление Pools.....	39
5.8.1	Доступ к меню Pools	39
5.8.2	Просмотр конфигурации	39
5.8.2.1	Просмотр полной конфигурации pools.....	39
5.8.3	Создание Pool Member	40
5.8.3.1	Параметры команды --create	40
5.8.3.2	Примеры создания pool member.....	40
5.8.3.3	Обработка ошибок при создании pool member	41
5.8.4	Валидация параметров	42

5.8.4.1	Адрес (address).....	42
5.8.4.2	Порт (port).....	42
5.8.4.3	Вес (weight)	42
5.8.4.4	Протокол (protocol)	42
5.8.5	Значения по умолчанию.....	42
5.8.6	Доступные команды	42
5.8.6.1	Команды в меню pools	42
5.8.7	Структура Pool	42
5.8.8	Примечания	43
5.9	 Управление downstream.....	43
5.9.1	Получение списка downstream listener'ов	43
5.9.2	Создание downstream listener	43
5.9.2.1	Параметры создания downstream listener	44
5.9.2.2	Примеры создания downstream listener	44
5.9.2.3	Обработка ошибок при создании downstream listener.....	44
5.9.3	Просмотр информации о downstream listener	44
5.9.3.1	Просмотр полной конфигурации downstream listener.....	45
5.9.4	Настройка атрибутов downstream listener	45
5.9.4.1	Адрес и порт (address).....	45
5.9.4.2	Другие атрибуты	45
5.9.5	Управление фильтрами	46
5.9.6	Управление инспекторами трафика downstream listener	46
5.9.7	Удаление listener.....	46
5.9.8	Доступные команды в меню listener	46
5.9.8.1	Команды в меню списка downstream	46
5.9.8.2	Команды в меню конкретного listener.....	47
5.10	 Хранилище сертификатов	47
5.10.1	Доступ к меню certs	47
5.10.2	Просмотр списка сертификатов.....	47
5.10.3	Просмотр сертификатов в файловой системе	47
5.10.3.1	Поддерживаемые форматы файлов	48
5.10.3.2	Загрузка сертификата на узел управления	48
5.10.4	Загрузка сертификата в хранилище.....	48
5.10.4.1	Процесс загрузки	48
5.10.4.2	Параметры загрузки	48
5.10.4.3	Имя сертификата	48
5.10.4.4	Примеры загрузки	48
5.10.4.5	Обработка ошибок при загрузке.....	49

5.10.5	Просмотр информации о сертификате	49
5.10.6	Управление сертификатом	50
5.10.7	Типы сертификатов	50
5.10.7.1	TLS Certificate	50
5.10.7.2	TLS Validation Context	50
5.10.8	Использование сертификатов	50
5.10.9	Хранение сертификатов	50
5.10.10	Безопасность	51
5.10.10.1	Рекомендации	51
5.10.11	Доступные команды	51
5.10.11.1	Команды в меню certs	51
5.10.12	Примеры использования	51
5.10.12.1	Полный цикл работы с сертификатом	51
5.11	 Управление RHI (Route Health Injection)	52
5.11.1	Обзор структуры меню RHI	52
5.11.2	Настройки API	52
5.11.2.1	Переход в меню API	52
5.11.2.2	Настройка URL API	52
5.11.2.3	Настройка интервала опроса	52
5.11.2.4	Настройка timeout	53
5.11.3	Настройки BGP	53
5.11.3.1	Переход в меню BGP	53
5.11.3.2	Настройка режима при старте	53
5.11.3.3	Управление BGP правилами	53
5.11.3.4	Обработка ошибок при работе с BGP правилами	56
5.11.4	Настройки порогов состояний upstream кластеров	56
5.11.4.1	Переход в меню upstream кластеров	56
5.11.4.2	Получение списка upstream кластеров	56
5.11.4.3	Добавление upstream кластера для отслеживания	56
5.11.4.4	Настройка порогов состояний upstream кластера	57
5.11.5	Доступные команды	58
5.11.5.1	Команды в главном меню RHI	58
5.11.5.2	Команды в меню BGP правил	58
5.11.5.3	Команды в меню upstream кластеров	58
5.11.6	Примеры использования	59
5.11.6.1	Пример 1: Полная настройка RHI	59
5.11.6.2	Пример 2: Просмотр конфигурации	59

5.11.7	Примечания.....	60
5.12	Управление пользователями.....	60
5.12.1	Обзор.....	60
5.12.2	Структура меню.....	60
5.12.3	Команды меню.....	60
5.12.3.1	Список пользователей.....	60
5.12.3.2	Работа с конкретным пользователем.....	60
5.12.3.3	Создание нового пользователя.....	61
5.12.3.4	Обновление пользователя.....	61
5.12.3.5	Навигация.....	62
5.12.4	Ролевая модель.....	62
5.12.4.1	Роли пользователей.....	62
5.12.4.2	Множественные роли.....	63
5.12.4.3	Валидация ролей.....	63
5.12.5	Примеры использования.....	63
5.12.5.1	Создание администратора.....	63
5.12.5.2	Создание оператора с правами пользователя.....	63
5.12.5.3	Создание аудитора.....	63
5.12.5.4	Изменение ролей пользователя.....	63
5.12.5.5	Деактивация пользователя.....	64
5.12.5.6	Просмотр информации о пользователе.....	64
5.12.6	Удаление пользователя.....	64
5.13	🔑 Application Programming Interface (API).....	64
5.13.1	Swagger документация.....	64
5.13.1.1	Доступ к Swagger UI.....	64
5.13.2	Базовый URL.....	64
5.13.3	Формат данных.....	65
5.13.4	Авторизация.....	65
5.13.4.1	Использование токена.....	65
5.13.5	Основные группы эндпоинтов.....	65
5.13.5.1	Аутентификация.....	65
5.13.5.2	Узлы обработки трафика.....	65
5.13.5.3	Upstream кластеры.....	65
5.13.5.4	Downstream listeners.....	65
5.13.5.5	Цепочки фильтров.....	66
5.13.5.6	Фильтры listeners.....	66
5.13.5.7	TLS сертификаты.....	66

5.13.5.8	Контексты валидации TLS.....	67
5.13.5.9	Пользователи	67
5.13.6	Коды ответов	67
5.13.7	Формат ошибок.....	67
5.13.8	Примеры использования	67
5.13.8.1	Получение списка узлов.....	67
5.13.8.2	Создание узла	68
5.13.8.3	Получение информации о кластере	68
5.13.9	Дополнительная информация	68
6	Маршрутизация	69
6.1	BGP (Border Gateway Protocol).....	69
6.2	RHI (Route Health Injection)	69
6.2.1	Выбор upstream кластеров.....	69
6.2.2	Настройка порогов состояний	69
6.2.3	Настройка BGP правил.....	69
7	Мониторинг.....	70
7.1	Метрики	70
7.1.1	Содержание раздела	70
7.1.2	Метрики сервера	70
7.1.3	Метрики слушателей (Listeners)	71
7.1.3.1	Основные метрики слушателя	71
7.1.3.2	Метрики по обработчикам (Per-handler).....	72
7.1.3.3	Метрики UDP	72
7.1.4	Метрики HTTP Connection Manager.....	72
7.1.4.1	Соединения	72
7.1.4.2	Запросы.....	73
7.1.4.3	Метрики трассировки.....	74
7.1.4.4	Метрики HTTP/1 codec.....	74
7.1.4.5	Метрики HTTP/2 codec.....	75
7.1.4.6	Метрики HTTP/3 codec.....	76
7.1.5	Метрики кластеров (Clusters).....	76
7.1.5.1	Общие метрики кластера	76
7.1.5.2	Метрики запросов кластера.....	77
7.1.5.3	Динамические HTTP метрики.....	78
7.1.5.4	Метрики членства кластера.....	79
7.1.5.5	Метрики health check.....	79
7.1.5.6	Метрики обнаружения выбросов (Outlier Detection)	80

7.1.5.7	Метрики circuit breakers	81
7.1.5.8	Метрики бюджета таймаутов	81
7.1.5.9	Метрики балансировщика нагрузки	81
7.1.5.10	Метрики подмножеств балансировщика нагрузки	82
7.1.5.11	Метрики ring hash балансировщика нагрузки	82
7.1.5.12	Метрики Maglev балансировщика нагрузки	82
7.1.5.13	Метрики размеров запросов и ответов	82
7.1.6	Метрики TLS/SSL	83
7.1.6.1	Метрики сертификатов	83
7.1.7	Метрики TCP	83
7.1.8	Метрики QUIC	84
7.1.9	Метрики менеджера слушателей	85
7.1.10	Метрики менеджера кластеров	85
7.1.10.1	Метрики по рабочим потокам	86
7.1.11	Использование метрик	86
7.1.11.1	Доступ к метрикам через Prometheus	86
7.1.11.2	Конфигурация Prometheus	86
7.1.11.3	Интеграция с Zabbix	86
7.1.11.4	Интеграция с Grafana	87
7.1.11.5	Примеры PromQL запросов	90
7.1.12	Примечания	90
7.1.13	Дополнительные ресурсы	90
8	Термины, сокращения и определения	92

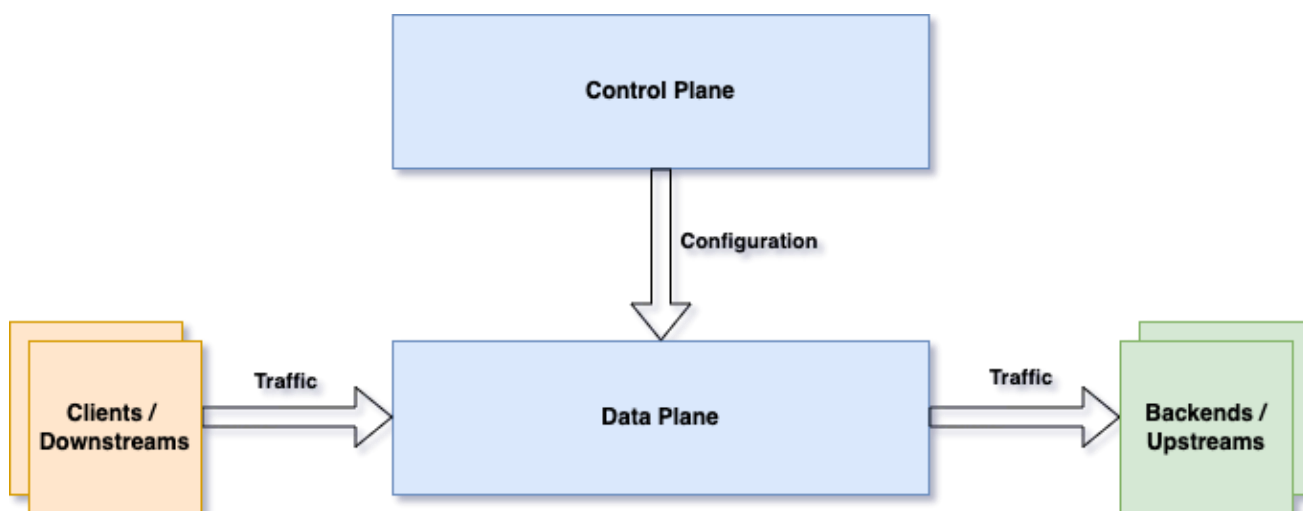
1 Общая архитектура решения

flxGATE — программное решение для балансировки нагрузки и управления сетевым трафиком, включающее балансировку на уровнях L4-L7, маршрутизацию и решения задач обеспечения отказоустойчивости.

Архитектурно решение flxGATE состоит из двух слоев

- **Data Plane** - слой обработки трафика
- **Control Plane** - слой управления узлами обработки трафика

1.1 Схема решения



1.2 Функциональность

1.2.1 Control Plane

Обеспечивает централизованное управление конфигурацией узлов обработки трафика. Основные функции:

- **Управление конфигурацией:** предоставление REST API для создания, изменения и удаления конфигурационных объектов (узлы, слушатели, кластеры, маршруты, секреты)
- **Доставка конфигурации:** распространение изменений конфигурации до узлов Data Plane в режиме реального времени
- **Хранение состояния:** централизованное хранение конфигурационных данных в распределенном хранилище
- **Мониторинг изменений:** отслеживание изменений в хранилище и автоматическая синхронизация с узлами обработки трафика
- **Управление узлами:** контроль жизненного цикла узлов, блокировка конфигурации для безопасного обновления

1.2.2 Data Plane

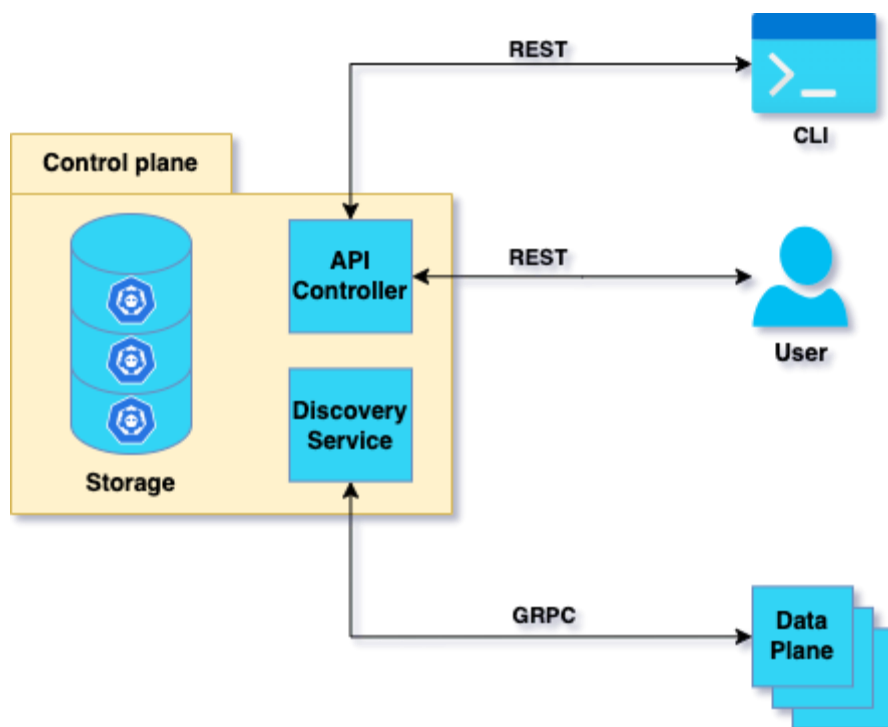
Отвечает за обработку сетевого трафика на основе конфигурации, полученной от Control Plane. Основные функции:

- **Обработка трафика:** прием, маршрутизация и балансировка входящего сетевого трафика
- **Динамическая конфигурация:** получение и применение обновлений конфигурации от Control Plane без перезапуска
- **Балансировка нагрузки:** распределение запросов между бэкенд-серверами с использованием различных алгоритмов (round-robin, least-request и др.)
- **Маршрутизация:** определение путей прохождения трафика на основе правил маршрутизации
- **Фильтрация трафика:** применение сетевых и HTTP фильтров для обработки, модификации и контроля трафика
- **Обеспечение безопасности:** поддержка TLS/SSL терминции, управление сертификатами и секретами
- **Мониторинг и наблюдаемость:** сбор метрик, логов доступа и статистики производительности
- **Управление маршрутизацией:** поддержка протоколов маршрутизации BGP, OSPF, VRRP, динамическое изменение маршрутов

2 Control Plane

Подсистема управления узлами обработки трафика

2.1 Компоненты



2.1.1 API Controller

Реализует интерфейс управления конфигурацией узлов Data Plane по протоколу REST.

Позволяет управлять основными сущностями конфигурации Data Plane: node, endpoint, clusters, listeners и т.д.

2.1.2 Discovery Service

Обеспечивает распространение изменений конфигурации до узлов Data Plane в режиме реального времени по протоколу GRPC.

2.1.3 Storage

Обеспечивает централизованное хранение конфигурационной информации.

Может быть развернуто в виде распределенного хранилища.

2.1.4 CLI

Обеспечивает возможность интерактивного управления узлами управления и обработки трафика.

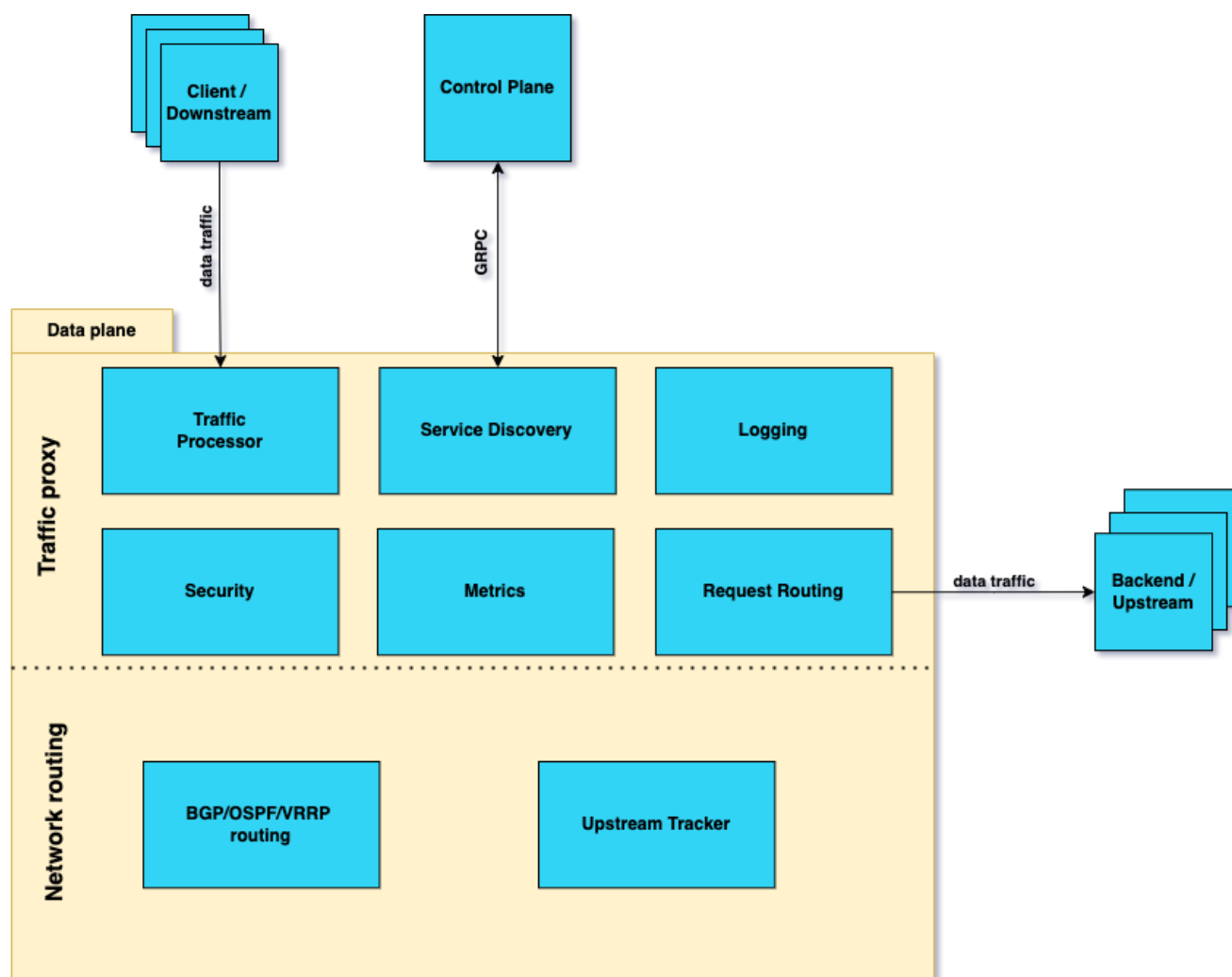
3 Data Plane

Подсистема обработки и управления трафиком

3.1 Компонентная схема

Состоит из двух основных подсистем:

- **Traffic proxy**: система обработки трафика
- **Network routing**: система управления сетевой маршрутизацией



3.1.1 Traffic Proxy

Реализует логику обработки трафика в соответствии с заданной конфигурацией.

Предоставляет функции обработки и маршрутизации трафика, сбора метрик, динамического конфигурирования, журналирования и безопасности.

3.1.1.1 Traffic Processor

Реализует функцию обработки и фильтрации поступающих запросов/пакетов.

3.1.1.2 Service Discovery

Реализует интерфейс взаимодействия с Control Plane для получения актуальной информации о конфигурации.

3.1.1.3 Logging

Реализует функционал журналирования: access-логи, трассировка.

3.1.1.4 Security

Реализует функции безопасности: работа с TLS/SSL, аутентификация/авторизация, хранение и использование чувствительной информации.

3.1.1.5 Metrics

Реализует сбор внутренних метрик с целью их дальнейшей передачи через систему мониторинга (Prometheus)

3.1.1.6 Request Routing

Реализует правила и действия для маршрутизации проходящего через Data Plane запроса.

3.1.2 Network routing

Реализует логику изменения маршрутизации на узле обработки трафика.

3.1.2.1 BGP/OSPF/VRRP routing

Предоставляет функции работы с протоколами маршрутизации BGP/OSPF/VRRP.

3.1.2.2 Upstream Tracker

Реализует логику проверки изменения маршрутизации в зависимости от состояния серверов, на которые балансируется трафик.

4 Установка

flxGATE поставляется как виртуальное устройство (Virtual Appliance).

4.1 Варианты поставки

Возможны следующие варианты установки:

1. **OVA** (Open Virtual Appliance VMWare) для VMware, VirtualBox;
2. **qcow2** (QEMU Copy-On-Write v2) для Linux-сред.

4.2 Операционные системы

Образы могут поставляться на базе следующих операционных систем:

- **Ubuntu 24.04**

4.3 Развертывание образа OVA

4.3.1 Предварительные требования

Для развертывания образа OVA необходима система виртуализации, например VMware ESXi не ниже 7 версии.

4.3.2 Импорт OVA образа

Через веб-интерфейс VMware ESXi, выполните следующие действия:

Подключение к ESXi

Войдите в веб-интерфейс VMware ESXi через браузер.

Запуск мастера импорта

В контекстном меню сервера выберите **Deploy OVF template**.

Импорт образа

Следуйте указаниям мастера, чтобы начать импорт образа OVA.

Создание виртуальной машины

В результате на сервере появится новая виртуальная машина.

Настройка сетевого адаптера VM

Убедитесь, что в виртуальной машине добавлен хотя бы один сетевой интерфейс.

Запуск системы

Запустите созданную виртуальную машину.

7 Настройка сети системы

Выполните настройку сети в запущенной системе.

4.4 Развертывание образа qcow2

4.4.1 Предварительные требования

Для развертывания образа qcow2 необходима система виртуализации, например qemu. Также должны быть установлены следующие утилиты:

- **libvirt**
- **qemu**
- **virsh**
- **virt-instsall**
- **virt-viewer**

4.4.2 Установка образа

Выполните в консоли Host OS

```
sudo mv flxgate-x86_64.qcow2 /var/lib/libvirt/images/
```

1 Создание виртуальной машины

Выполните в консоли Host OS

```
sudo virt-install --name flxgate1 --ram 2048 --disk /var/lib/libvirt/images/flxgate-x86_64.qcow2 --os-variant generic --import
```

2 Настройка сети системы

Выполните настройку сети в запущенной системе.

5 Управление

5.1 Command Line Interface (CLI)

Через интерфейс командной строки вы можете управлять настройками flxGATE.

5.2 Запуск CLI

Для запуска интерфейса командной строки (CLI), выполните следующие действия:

1 Подключение к серверу

Подключитесь к SSH-серверу flxGATE:

```
ssh flxgate@flxgate-host
```

2 Авторизация

После подключения введите пароль (пароль предоставляется отдельно при выдаче дистрибутива):

```
flxgate@flxgate-host's password: [введите пароль]
```

3 Начало работы

Если авторизация прошла успешно, откроется терминал flxGATE:

```
flxgate-host#
```

 **Справка:** Для получения помощи используйте команду `--help`

5.3 Управление настройками сети

Через интерфейс командной строки вы можете управлять настройками сети узла flxGATE.

Для управления используются:

- **Устройство** — это физическое или виртуальное сетевое оборудование, которое есть в системе
- **Соединение** — это набор настроек, которые можно применить к устройству. Это конфигурационный профиль.

После запуска и авторизации в CLI введите **network**:

```
flxgate-host#network
```

```
flxgate-host#network>
```

5.3.1 Настройка устройств

Находясь в меню настройки сети введите **devices**

```
flxgate-host#network>devices
```

```
flxgate-host#network.devices>
```

5.3.1.1 Получение списка устройств

Находясь в меню списка устройств нажмите клавишу табуляции

```
flxgate-host#network.devices>
ens160  vlan100 lo      --back --exit --help --clear
```

Будут отображены сетевые устройства и команды управления

5.3.1.2 Просмотр информации об устройстве

Введите название устройства и нажмите клавишу ввода. Затем введите команду **--show**

```
flxgate-host#network.devices>ens160
```

```
flxgate-host#network.devices.ens160>--show
{
  "device": "ens160",
  "type": "ethernet",
  "state": "100 (connected)",
  "connection": "ens160",
  "hwaddr": "00:50:56:B4:BC:0E",
  "speed": "10000 Mb/s"
}
```

5.3.1.3 Отключение устройства

В меню устройства введите команду **--disconnect**

```
flxgate-host#network.devices.ens160>--disconnect
```

```
flxgate-host#network.devices.ens160>
```

5.3.1.4 Подключение устройства

В меню устройства введите команду **--connect**

```
flxgate-host#network.devices.ens160>--connect
```

```
flxgate-host#network.devices.ens160>
```

5.3.2 Настройка соединений

Находясь в меню настройки сети введите **connections**

```
flxgate-host#network>connections
```

```
flxgate-host#network.connections>
```

5.3.2.1 Получение списка соединений

Находясь в меню списка соединений нажмите клавишу табуляции

```
flxgate-host#network.connections>
--help --clear vlan100 lo      ens160  --create --back --exit
```

Будут отображены сетевые подключения и команды управления

5.3.2.2 Создание соединения

Находясь в меню соединения, введите команду **--create** с параметрами.

Пример 1. Создание соединения с DHCP

```
flxgate-host#network.connections>--create ethernet eth0 eth0
ipv4.method=auto;ipv6.method=disabled
```

```
Creating connection eth0...
Connection eth0 created successfully
flxgate-host#network.connections>
```

Пример 2. Создание соединения со статическими параметрами ipv4

```
flxgate-host#network.connections>--create ethernet eth0 eth0
ipv4.method=manual;ipv4.addresses=192.168.1.100/24;ipv4.gateway=192.168.1.1;ipv4.dns=8.8.8.8;ipv6.method=disabled
```

```
Creating connection eth0...
Connection eth0 created successfully
flxgate-host#network.connections>
```

Пример 3. Создание соединения с VLAN ID 100

```
flxgate-host#network.connections>--create vlan vlan100 vlan100
ipv4.method=auto;ipv6.method=disabled;id=100;dev=ens160
```

```
Creating connection vlan100...
Connection vlan100 created successfully
flxgate-host#network.connections>
```

5.3.2.3 Просмотр информации о подключении

Введите название соединения и нажмите клавишу ввода. Затем введите команду **--show**

```
flxgate-host#network.connections>ens160
flxgate-host#network.connections.ens160>--show
```

```
{
  "id": "ens160",
  "type": "802-3-ethernet",
  "interface": "ens160",
  "autoconnect": "yes",
  "params": {
    "ipv4.addresses": "10.31.72.71/24",
    "ipv4.auto-route-ext-gw": "-1",
    "ipv4.dad-timeout": "-1",
    "ipv4.dhcp-client-id": "",
    "ipv4.dhcp-dscp": "",
    "ipv4.dhcp-fqdn": "",
    "ipv4.dhcp-hostname": "",
    "ipv4.dhcp-hostname-flags": "0x0",
    "ipv4.dhcp-iaid": "",
    "ipv4.dhcp-reject-servers": "",
    "ipv4.dhcp-send-hostname": "yes",
    "ipv4.dhcp-timeout": "0",
    "ipv4.dhcp-vendor-class-identifier": "",
    "ipv4.dns": "8.8.8.8",
    "ipv4.dns-options": "",
    "ipv4.dns-priority": "0",
    "ipv4.dns-search": "",
    "ipv4.gateway": "10.31.72.1",
    "ipv4.ignore-auto-dns": "no",
    "ipv4.ignore-auto-routes": "no",
```

```

    "ipv4.link-local": "0",
    "ipv4.may-fail": "yes",
    "ipv4.method": "manual",
    "ipv4.never-default": "no",
    "ipv4.replace-local-rule": "-1",
    "ipv4.required-timeout": "-1",
    "ipv4.route-metric": "-1",
    "ipv4.route-table": "0",
    "ipv4.routes": "",
    "ipv4.routing-rules": ""
  }
}
flxgate-host#network.connections.ens160>

```

5.3.2.4 Изменение соединения

Находясь в меню соединения, которое необходимо изменить, введите команду **--modify** с параметрами.

Пример 1. Включение автоматического получения IPV4 адреса

```
flxgate-host#network.connections.ens160>--modify ipv4.method=auto;ipv6.method=disabled
```

```
flxgate-host#network.connections>
```

Пример 2. Настройка статических параметров соединения

```
flxgate-host#network.connections.ens160>--modify
ipv4.method=manual;ipv4.addresses=192.168.1.100/24;ipv4.gateway=192.168.1.1;ipv4.dns=8.8.8
.8;ipv6.method=disabled
```

```
flxgate-host#network.connections>
```

5.3.2.5 Удаление соединения

Находясь в меню соединения, которое необходимо удалить, введите команду **--delete**

```
flxgate-host#network.connections.ens160>--delete
```

```
Connection deleted successfully
flxgate-host#network.connections>
```

5.3.2.6 Отключение соединения

Находясь в меню соединения, которое необходимо отключить, введите команду **--down**

```
flxgate-host#network.connections.ens160>--down
```

```
flxgate-host#network.connections>
```

5.3.2.7 Включение соединения

Находясь в меню соединения, которое необходимо включить, введите команду **--up**

```
flxgate-host#network.connections.ens160>--up
```

```
flxgate-host#network.connections>
```

 **Справка:** Для получения помощи используйте команду **--help**

5.4 Управление узлами

Интерфейс командной строки позволяет управлять узлам обработки трафика (Data Plane nodes).

После запуска и авторизации в CLI введите **nodes**:

```
flxgate-host#nodes↵
```

```
flxgate-host#nodes>
```

5.4.1 Получение списка узлов

Находясь в меню узлов, нажмите клавишу табуляции:

```
flxgate-host#nodes>
node1 node2 --back --exit --help --clear
```

Будут отображены доступные узлы и команды управления.

Команды управления имеют префикс '--'.


5.4.2 Создание узла

Для создания нового узла введите команду **--create** с указанием идентификатора узла:

```
flxgate-host#nodes>--create node1↵
```

```
Creating node node1...
Node node1 created successfully
flxgate-host#nodes>
```

После создания узла он появится в списке доступных узлов, и вы сможете перейти к его настройке.

 **Примечание:** Идентификатор узла должен быть уникальным. При попытке создать узел с уже существующим идентификатором будет возвращена ошибка.

5.4.2.1 Обработка ошибок при создании узла

При попытке создать узел с уже существующим идентификатором:

```
flxgate-host#nodes>--create node1↵
```

```
Creating node node1...
Error: Node already exists
flxgate-host#nodes>
```

5.4.3 Просмотр информации об узле

Введите название узла и нажмите клавишу ввода:

```
flxgate-host#nodes>node1↵
```

```
flxgate-host#nodes.node1>
```

5.4.4 Управление конфигурацией узла

Находясь в меню конкретного узла, доступны следующие команды для управления конфигурацией:

5.4.4.1 Включение режима конфигурации

Для начала редактирования конфигурации узла необходимо включить режим конфигурации. Введите команду **--conf_on**:

```
flxgate-host#nodes.node1>--conf_on↵
```

```
Configuration mode enabled successfully
flxgate-host#nodes.node1>
```

⚠ **Важно:** Режим конфигурации должен быть включен перед внесением изменений в конфигурацию узла.

5.4.4.2 Выключение режима конфигурации

После завершения редактирования конфигурации выключите режим конфигурации командой **--conf_off**:

```
flxgate-host#nodes.node1>--conf_off↵
```

```
Configuration mode disabled successfully
flxgate-host#nodes.node1>
```

5.4.4.3 Публикация конфигурации

После внесения изменений в конфигурацию и выключения режима конфигурации, примените изменения к data plane командой **--conf_publish**:

```
flxgate-host#nodes.node1>--conf_publish↵
```

```
Configuration promoted successfully
flxgate-host#nodes.node1>
```

💡 **Примечание:** Публикация конфигурации возможна только после выключения режима конфигурации.

5.4.5 Удаление узла

Для удаления узла введите команду **--delete**:

```
flxgate-host#nodes.node1>--delete↵
```

```
Deleting node node1...
Node deleted successfully
flxgate-host#nodes>
```

⚠ **Внимание:** Удаление узла невозможно, если узел заблокирован (находится в режиме конфигурации) или имеет активные зависимости.

5.4.6 Примеры использования

5.4.6.1 Пример 1. Создание и настройка нового узла

```
# Переход в меню узлов
flxgate-host#nodes↵
```

```
# Создание нового узла
flxgate-host#nodes>--create node1↵
Creating node node1...
Node node1 created successfully
```

```
# Выбор узла
```

```
flxgate-host#nodes>node1↵

# Включение режима конфигурации
flxgate-host#nodes.node1>--conf_on↵
Configuration mode enabled successfully

# Настройка кластеров и слушателей
flxgate-host#nodes.node1>clusters↵
# ... настройка кластеров ...
flxgate-host#nodes.node1.clusters>--back↵

flxgate-host#nodes.node1>listeners↵
# ... настройка слушателей ...
flxgate-host#nodes.node1.listeners>--back↵

# Выключение режима конфигурации
flxgate-host#nodes.node1>--conf_off↵
Configuration mode disabled successfully

# Публикация конфигурации
flxgate-host#nodes.node1>--conf_publish↵
Configuration promoted successfully
```

5.4.6.2 Пример 2. Обработка ошибок

При попытке публикации конфигурации без предварительного выключения режима конфигурации:

```
flxgate-host#nodes.node1>--conf_publish↵

Node is not locked
flxgate-host#nodes.node1>
```

При попытке удаления узла, который находится в режиме конфигурации:

```
flxgate-host#nodes.node1>--delete↵

Deleting node node1...
Error: Cannot delete node node1 - conflict
flxgate-host#nodes.node1>
```

При попытке включения режима конфигурации, когда он уже включен:

```
flxgate-host#nodes.node1>--conf_on↵

Configuration mode is already enabled
flxgate-host#nodes.node1>
```

5.5 Управление upstream

Интерфейс командной строки позволяет управлять upstream кластерами, обрабатывающими входящие запросы от flxGATE Data Plane.

После запуска и авторизации в CLI перейдите в меню узла и введите **upstream**:

```
flxgate-host#nodes.node1>upstream↵

flxgate-host#nodes.node1.upstream>
```


5.5.1 Получение списка upstream кластеров

Находясь в меню upstream, нажмите клавишу табуляции:

```
flxgate-host#nodes.node1.upstream>
http_8080_web-api https_443_api --create --back --exit --help --clear
```

Будут отображены доступные upstream кластера и команды управления.

Команды управления имеют префикс '--'.

⚠ Важно:

Режим редактирования конфигурации должен быть включен перед созданием, удалением, внесением изменений в конфигурацию upstream кластера.

После внесения изменений необходимо опубликовать изменения в конфигурацию.

Команды управления конфигурацией (**--conf_on**, **--conf_off**, **--conf_publish**) доступны как в меню списка upstream кластеров, так и в меню конкретного upstream кластера.

5.5.2 Создание upstream кластера

Для создание нового upstream кластера необходимо перевести узел обработки данных в режим редактирования.

Для создания нового upstream кластера введите команду **--create** с указанием протокола, порта и имени сервиса:

```
flxgate-host#nodes.node1.upstream>--create http 8080 web-api↵
Creating cluster http_8080_web-api...
Cluster http_8080_web-api created successfully
flxgate-host#nodes.node1.upstream>
```

5.5.2.1 Параметры создания upstream кластера

Команда **--create** принимает следующие параметры:

- **protocol** — протокол: `https`, `http`, `tcp`, `udp`, `grpc`
- **port** — номер порта
- **service** — имя сервиса

Имя upstream кластера формируется автоматически в формате: `{protocol}_{port}_{service}`

5.5.2.2 Примеры создания upstream кластера

Пример 1. Создание HTTP upstream кластера:

```
flxgate-host#nodes.node1.upstream>--create http 8080 web-api↵
Creating cluster http_8080_web-api...
Cluster http_8080_web-api created successfully
flxgate-host#nodes.node1.upstream>
```

Пример 2. Создание HTTPS upstream кластера:

```
flxgate-host#nodes.node1.upstream>--create https 443 api↵
Creating cluster https_443_api...
Cluster https_443_api created successfully
```

```
flxgate-host#nodes.node1.upstream>
```

5.5.2.3 Обработка ошибок при создании upstream кластера

При попытке создать upstream кластера с уже существующим именем:

```
flxgate-host#nodes.node1.upstream>--create http 8080 web-api↵
```

```
Creating cluster http_8080_web-api...
Error: Cluster http_8080_web-api already exists
flxgate-host#nodes.node1.upstream>
```

При указании неверных параметров:

```
flxgate-host#nodes.node1.upstream>--create http↵
```

```
Usage: --create <protocol> <port> <service>
      protocol: https|http|tcp|udp|grpc
      port: port number
      service: service name
Example: --create http 8080 web-api
flxgate-host#nodes.node1.upstream>
```

5.5.3 Просмотр информации об upstream кластера

Введите название upstream кластера и нажмите клавишу ввода:

```
flxgate-host#nodes.node1.upstream>http_8080_web-api↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>
```

5.5.3.1 Просмотр полной конфигурации upstream кластера

Для просмотра полной конфигурации upstream кластера в формате JSON используйте команду -**show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>--show↵
```

```
{
  "name": "http_8080_web-api",
  "type": "STATIC",
  "lb_policy": "LEAST_REQUEST",
  "connect_timeout": "5s",
  "load_assignment": {
    "cluster_name": "http_8080_web-api",
    "endpoints": []
  }
}
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>
```

5.5.4 Настройка атрибутов upstream кластера

Находясь в меню конкретного upstream кластера, доступны следующие атрибуты для настройки:

5.5.4.1 Таймаут подключения (connect_timeout)

Настройка таймаута для новых сетевых подключений к хостам в кластере:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>connect_timeout↵
```

```
current 'connect_timeout' attribute value: 5s
```

Введите новое значение в формате длительности (например, `10s`, `500ms`) или нажмите Enter для сохранения текущего значения.

5.5.4.2 Тип обнаружения сервисов (type)

Настройка типа обнаружения сервисов для разрешения кластера:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>type↵
```

```
current 'type' attribute value: STATIC
STATIC STRICT_DNS LOGICAL_DNS ORIGINAL_DST
```

Доступные типы:

- **STATIC** — конфигурация явно указывает сетевой адрес каждого upstream хоста
- **STRICT_DNS** — непрерывное и асинхронное разрешение указанных DNS целей
- **LOGICAL_DNS** — логический DNS кластер использует только первый IP адрес, возвращаемый при необходимости инициировать новое соединение
- **ORIGINAL_DST** — запросы перенаправляются на upstream хосты без явной конфигурации хоста

Выберите нужный тип, введя его название.

5.5.4.3 Параметры подключения (connection_options)

Настройка параметров для upstream подключений:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>connection_options↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.connection_options>
```

Доступны следующие под-атрибуты:

- **keepalive_probes** — максимальное количество keepalive проб без ответа перед решением о том, что соединение мертво
- **keepalive_time** — количество секунд, в течение которых соединение должно быть простаивающим, прежде чем начнут отправляться keep-alive пробы
- **keepalive_interval** — количество секунд между keep-alive пробами

Для изменения атрибута введите его название и следуйте инструкциям:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.connection_options>keepalive_probes↵
```

```
current 'keepalive_probes' attribute value: 9
```

Введите новое значение или нажмите Enter для сохранения текущего.

5.5.5 Удаление upstream кластера

Для удаления upstream кластера необходимо перевести узел обработки данных в режим редактирования.

Для удаления upstream кластера введите команду **--delete**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>--delete↵
```

```
Deleting cluster http_8080_web-api...
Cluster deleted successfully
flxgate-host#nodes.node1.upstream>
```

После удаления необходимо опубликовать сделанные изменения.

⚠ **Внимание:** Удаление upstream кластера невозможно, если узел заблокирован (находится в режиме конфигурации) или upstream кластера имеет активные зависимости.

5.5.6 Доступные команды в меню cluster

5.5.6.1 Команды в меню списка upstream

- **--create** — создание нового upstream кластера
- **--back** — возврат в предыдущее меню
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.5.6.2 Команды в меню конкретного cluster

- **--show** — просмотр полной конфигурации cluster в формате JSON
- **--delete** — удаление upstream кластера
- **--back** — возврат в меню списка upstream
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.6 Управление Circuit Breakers

Circuit breakers защищают upstream кластеры от перегрузки, ограничивая количество соединений, запросов и повторных попыток.

5.6.1 Доступ к меню Circuit Breakers

Для управления circuit breakers перейдите в меню upstream и введите команду **circuit_breakers**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>circuit_breakers
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers>
```

⚠ **Важно:**

[Режим редактирования](#) конфигурации должен быть включен перед внесением изменений в circuit breakers.

После внесения изменений необходимо опубликовать изменения в конфигурацию.

Команды управления конфигурацией (**--conf_on**, **--conf_off**, **--conf_publish**) доступны в меню circuit breakers.

5.6.2 Просмотр конфигурации

5.6.2.1 Просмотр полной конфигурации circuit breakers

Для просмотра полной конфигурации circuit breakers в формате JSON используйте команду **--show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers>--show
```

```
{
  "thresholds": [
    {
      "priority": "DEFAULT",
      "max_connections": 1024,
      "max_pending_requests": 1024,
      "max_requests": 1024,
      "max_retries": 3
    }
  ]
}
```

5.6.3 Управление Thresholds (порогами)

Для настройки порогов circuit breakers введите команду **thresholds**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers>thresholds
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers.thresholds>
```

5.6.3.1 Просмотр конфигурации thresholds

Для просмотра конфигурации thresholds используйте команду **--show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers.thresholds>--show
```

```
[
  {
    "priority": "DEFAULT",
    "max_connections": 1024,
    "max_pending_requests": 1024,
    "max_requests": 1024,
    "max_retries": 3
  }
]
```

5.6.4 Приоритеты Thresholds

Система поддерживает два уровня приоритетов для **thresholds**:

- **DEFAULT** — пороги по умолчанию для всех запросов
- **HIGH** — пороги для запросов с высоким приоритетом

5.6.4.1 Настройка Default Priority

Для настройки порогов по умолчанию введите команду **default_priority**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers.thresholds>default_priority
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers.thresholds.default_priority>
```

5.6.4.1.1 Просмотр конфигурации Default Priority

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.circuit_breakers.thresholds.default_priority>--show
```

```
{
```

```
"priority": "DEFAULT",
"max_connections": 1024,
"max_pending_requests": 1024,
"max_requests": 1024,
"max_retries": 3
}
```

5.6.4.1.2 Параметры Default Priority

Доступны следующие параметры:

- **max_connections** — максимальное количество соединений к upstream кластеру
- **max_pending_requests** — максимальное количество ожидающих запросов к upstream кластеру
- **max_requests** — максимальное количество параллельных запросов к upstream кластеру
- **max_retries** — максимальное количество параллельных повторных попыток к upstream кластеру

НАСТРОЙКА MAX_CONNECTIONS

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.default_priority>max_connections↵
```

current 'max_connections' attribute value: 1024

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

НАСТРОЙКА MAX_PENDING_REQUESTS

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.default_priority>max_pending_requests↵
```

current 'max_pending_requests' attribute value: 1024

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

НАСТРОЙКА MAX_REQUESTS

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.default_priority>max_requests↵
```

current 'max_requests' attribute value: 1024

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

НАСТРОЙКА MAX_RETRIES

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.default_priority>max_retries↵
```

current 'max_retries' attribute value: 3

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

5.6.4.2 Настройка High Priority

Для настройки порогов высокого приоритета введите команду **high_priority**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds>high_priority↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.high_priority>
```

5.6.4.2.1 Просмотр конфигурации High Priority

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.circuit_breakers.thresholds.high_priority>--show↵
```

```
{
  "priority": "HIGH",
  "max_connections": 1024,
  "max_pending_requests": 1024,
  "max_requests": 1024,
  "max_retries": 3
}
```

5.6.4.2.2 Параметры High Priority

Параметры High Priority идентичны параметрам Default Priority:

- **max_connections** — максимальное количество соединений к upstream кластеру
- **max_pending_requests** — максимальное количество ожидающих запросов к upstream кластеру
- **max_requests** — максимальное количество параллельных запросов к upstream кластеру
- **max_retries** — максимальное количество параллельных повторных попыток к upstream кластеру

Настройка параметров выполняется аналогично Default Priority.

5.6.4.3 Значения по умолчанию

При создании нового threshold используются следующие значения по умолчанию:

- **max_connections**: 1024
- **max_pending_requests**: 1024
- **max_requests**: 1024
- **max_retries**: 3

5.6.5 Доступные команды

5.6.5.1 Команды в меню circuit_breakers

- **thresholds** — управление пороговыми значениями circuit breakers
- **--show** — просмотр полной конфигурации circuit breakers в формате JSON
- **--back** — возврат в меню upstream кластера
- **--conf_on** — включение режима конфигурации

- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.6.5.2 Команды в меню thresholds

- **default_priority** — настройка порогов по умолчанию
- **high_priority** — настройка порогов высокого приоритета
- **--show** — просмотр конфигурации thresholds в формате JSON
- **--back** — возврат в меню circuit_breakers
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.6.5.3 Команды в меню default_priority / high_priority

- **max_connections** — настройка максимального количества соединений
- **max_pending_requests** — настройка максимального количества ожидающих запросов
- **max_requests** — настройка максимального количества параллельных запросов
- **max_retries** — настройка максимального количества параллельных повторных попыток
- **--show** — просмотр конфигурации threshold в формате JSON
- **--back** — возврат в меню thresholds
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.7 Управление Health Checks

Health checks (проверки здоровья) позволяют автоматически определять состояние upstream хостов и исключать неработающие хосты из балансировки нагрузки.

5.7.1 Доступ к меню Health Checks

Для управления health checks перейдите в меню upstream кластера и введите команду **health_checks**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>health_checks^
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks>
```

 **Важно:**

Режим редактирования конфигурации должен быть включен перед созданием, удалением или внесением изменений в health checks.

После внесения изменений необходимо опубликовать изменения в конфигурацию.

Команды управления конфигурацией (**--conf_on**, **--conf_off**, **--conf_publish**) доступны в меню `health checks`.

5.7.2 Просмотр конфигурации

5.7.2.1 Просмотр полной конфигурации `health checks`

Для просмотра полной конфигурации всех `health checks` в формате JSON используйте команду **--show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks>--show
```

```
[
  {
    "timeout": "1s",
    "interval": "5s",
    "unhealthy_threshold": 3,
    "healthy_threshold": 2,
    "http_health_check": {
      "host": "localhost",
      "path": "/health",
      "method": "GET",
      "expected_statuses": [200]
    }
  }
]
```

5.7.3 Создание Health Check

Для создания нового `health check` используйте команду **--add** с указанием типа проверки:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks>--add http
```

```
Health check added successfully to cluster: http_8080_web-api at position 1
```

5.7.3.1 Типы Health Checks

Поддерживаются следующие типы `health checks`:

- **http** — HTTP health check
- **tcp** — TCP health check
- **grpc** — gRPC health check

5.7.3.2 Параметры команды **--add**

```
--add <health_check_type>
```

Где `health_check_type` может быть: `http`, `tcp` или `grpc`.

5.7.3.3 Значения по умолчанию

При создании нового `health check` используются следующие значения по умолчанию:

- **timeout**: 1s (1 секунда)
- **interval**: 5s (5 секунд)
- **unhealthy_threshold**: 3
- **healthy_threshold**: 2

Для HTTP health check дополнительно: - **host**: "localhost" - **path**: "/health" - **method**: "GET"
- **expected_statuses**: [200]

Для TCP health check дополнительно: - **send**: "TCP health check request" - **receive**: ["TCP health check response"]

Для gRPC health check дополнительно: - **service_name**: "grpc.health.v1.Health"

5.7.4 Управление Health Check

После создания health check в меню появляется элемент **item_N**, где N — порядковый номер health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks>
item_1 --add --show --back --conf_on --conf_off --conf_publish
```

Для настройки конкретного health check введите его номер:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks>item_1↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>
```

5.7.4.1 Просмотр конфигурации Health Check

Для просмотра конфигурации конкретного health check используйте команду **--show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>--show↵
```

```
{
  "timeout": "1s",
  "interval": "5s",
  "unhealthy_threshold": 3,
  "healthy_threshold": 2,
  "http_health_check": {
    "host": "localhost",
    "path": "/health",
    "method": "GET",
    "expected_statuses": [200]
  }
}
```

5.7.4.2 Удаление Health Check

Для удаления health check используйте команду **--delete**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>--delete↵
```

```
Health check deleted successfully from cluster: http_8080_web-api
```

5.7.5 Общие параметры Health Check

Все типы health checks имеют следующие общие параметры:

5.7.5.1 timeout

Время ожидания ответа на health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>timeout↵
```

```
current 'timeout' attribute value: 1s
```

Введите новое значение в формате длительности (например, 2s, 500ms) или нажмите Enter для сохранения текущего.

5.7.5.2 interval

Интервал между health checks:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>interval↵
```

```
current 'interval' attribute value: 5s
```

Введите новое значение в формате длительности (например, 10s, 1m) или нажмите Enter для сохранения текущего.

5.7.5.3 unhealthy_threshold

Количество неуспешных health checks, после которого хост помечается как нездоровый:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>unhealthy_threshold↵
```

```
current 'unhealthy_threshold' attribute value: 3
```

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

5.7.5.4 healthy_threshold

Количество успешных health checks, после которого хост помечается как здоровый:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>healthy_threshold↵
```

```
current 'healthy_threshold' attribute value: 2
```

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

5.7.6 HTTP Health Check

Для HTTP health check доступны следующие дополнительные параметры:

5.7.6.1 host

Значение заголовка Host в HTTP запросе health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>host↵
```

```
current 'host' attribute value: localhost
```

Введите новое значение или нажмите Enter для сохранения текущего.

5.7.6.2 path

HTTP путь, который будет запрошен во время health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>path↵
```

```
current 'path' attribute value: /health
```

Введите новый путь или нажмите Enter для сохранения текущего.

5.7.6.3 method

HTTP метод для health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>method↵
```

```
current 'method' attribute value: GET
```

```
GET HEAD POST PUT DELETE OPTIONS TRACE PATCH
```

Доступные методы: **GET, HEAD, POST, PUT, DELETE, OPTIONS, TRACE, PATCH.**

Выберите нужный метод, введя его название.

5.7.6.4 expected_statuses

HTTP статус-коды ответа, которые считаются успешными:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>expected_statuses↵
```

```
current 'expected_statuses' attribute value: [200]
```

Введите список статус-кодов через запятую (например 200,201,204) или нажмите Enter для сохранения текущего.

5.7.6.5 send

Тело запроса для отправки (используется для POST, PUT и других методов с телом):

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>send↵
```

```
current 'send' attribute value:
```

Введите тело запроса или нажмите Enter для сохранения текущего.

5.7.6.6 receive

Ожидаемое содержимое ответа от сервера:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>receive↵
```

```
current 'receive' attribute value:
```

Введите ожидаемое содержимое или нажмите Enter для сохранения текущего.

5.7.6.7 request_headers_to_add

HTTP заголовки для добавления к запросу:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>request_headers_to_add↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1].request_headers_to_add>
```

5.7.6.7.1 Добавление заголовка

Для добавления заголовка используйте команду **--add**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1].request_headers_to_add>--add X-MSISDN 9999999999↵
```

5.7.6.7.2 Управление заголовком

После добавления заголовков появляется в меню. Для настройки значения заголовка введите его название:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1].request_headers_to_add>X-MSISDN↵
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1].request_headers_to_add.X-MSISDN>
```

Настройка значения заголовка

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1].request_headers_to_add.X-MSISDN>value↵
```

current 'value' attribute value: 9999999999

Введите новое значение или нажмите Enter для сохранения текущего.

Удаление заголовка

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.health_checks[1].request_headers_to_add.X-MSISDN>--delete
```

5.7.6.8 request_headers_to_remove

HTTP заголовки для удаления из запроса:

```
flxgate-host#nodes.node1.upstream.http_8080_web-
api.health_checks[1]>request_headers_to_remove
```

current 'request_headers_to_remove' attribute value:

Введите список имен заголовков через запятую (например, X-Forwarded-For,User-Agent) или нажмите Enter для сохранения текущего.

5.7.6.9 response_buffer_size

Размер буфера ответа в байтах для сопоставления содержимого:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>response_buffer_size
```

current 'response_buffer_size' attribute value: 0

Введите новое значение (положительное целое число) или нажмите Enter для сохранения текущего.

5.7.7 TCP Health Check

Для TCP health check доступны следующие параметры:

5.7.7.1 send

Данные для отправки:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>send
```

current 'send' attribute value: TCP health check request

Введите новые данные или нажмите Enter для сохранения текущего.

5.7.7.2 receive

Ожидаемые данные от сервера:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>receive
```

current 'receive' attribute value: TCP health check response

Введите ожидаемые данные или нажмите Enter для сохранения текущего.

5.7.8 gRPC Health Check

Для gRPC health check доступны следующие параметры:

5.7.8.1 service_name

Имя gRPC сервиса для health check:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>service_name
```

current 'service_name' attribute value: grpc.health.v1.Health
Введите новое имя сервиса или нажмите Enter для сохранения текущего.

5.7.8.2 authority

Значение заголовка authority в gRPC health check запросе:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.health_checks[1]>authority
```

current 'authority' attribute value:
Введите новое значение или нажмите Enter для сохранения текущего.

5.7.9 Доступные команды

5.7.9.1 Команды в меню health_checks

- **item_N** — управление health check с номером N
- **--add** — создание нового health check
- **--show** — просмотр полной конфигурации всех health checks в формате JSON
- **--back** — возврат в меню upstream кластера
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.7.9.2 Команды в меню health check (item_N)

- **timeout** — настройка времени ожидания ответа
- **interval** — настройка интервала между проверками
- **unhealthy_threshold** — настройка порога для пометки хоста как нездорового
- **healthy_threshold** — настройка порога для пометки хоста как здорового
- **--show** — просмотр конфигурации health check в формате JSON
- **--delete** — удаление health check
- **--back** — возврат в меню health_checks
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.7.9.2.1 Дополнительные команды для HTTP health check

- **host** — настройка заголовка Host
- **path** — настройка HTTP пути
- **method** — выбор HTTP метода
- **expected_statuses** — настройка ожидаемых статус-кодов
- **send** — настройка тела запроса
- **receive** — настройка ожидаемого содержимого ответа

- **request_headers_to_add** — управление заголовками для добавления
- **request_headers_to_remove** — настройка заголовков для удаления
- **response_buffer_size** — настройка размера буфера ответа

5.7.9.2.2 Дополнительные команды для TCP health check

- **send** — настройка данных для отправки
- **receive** — настройка ожидаемых данных

5.7.9.2.3 Дополнительные команды для gRPC health check

- **service_name** — настройка имени gRPC сервиса
- **authority** — настройка заголовка authority

5.8 Управление Pools

Pools (пулы) — это upstream хосты, которые обрабатывают входящие запросы. Управление pools позволяет добавлять, просматривать и настраивать upstream хосты для кластера.

5.8.1 Доступ к меню Pools

Для управления pool members перейдите в меню upstream кластера и введите команду **pools**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api>pools
```

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>
```

 **Важно:**

Режим редактирования конфигурации должен быть включен перед созданием pools.

После внесения изменений необходимо опубликовать изменения в конфигурацию.

Команды управления конфигурацией (**--conf_on**, **--conf_off**, **--conf_publish**) доступны в меню pools.

5.8.2 Просмотр конфигурации

5.8.2.1 Просмотр полной конфигурации pools

Для просмотра полной конфигурации всех pools в формате JSON используйте команду **--show**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--show
```

```
[
  {
    "pools": [
      {
        "pool_member": {
          "address": {
            "address": "192.168.10.1",
            "port_value": 443,
            "protocol": "TCP"
          }
        }
      }
    ]
  }
]
```

```

        "health_check_config": {
            "port_value": 443
        },
        "load_balancing_weight": 1
    }
]
]

```

5.8.3 Создание Pool Member

Для создания нового pool member используйте команду **--create**:

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443
```

```

Creating pool member 192.168.10.1:443...
Pool member 192.168.10.1:443 created successfully in cluster http_8080_web-api

```

5.8.3.1 Параметры команды --create

Команда **--create** принимает следующие параметры:

```
--create <address> <port> [weight] [<protocol>]
```

5.8.3.1.1 Обязательные параметры

- **address** — адрес pool member (FQDN, IPv4 или IPv6 адрес)
- **port** — номер порта (от 1 до 65535)

5.8.3.1.2 Опциональные параметры

- **weight** — вес для балансировки нагрузки (от 1 до 4294967295, по умолчанию: 1)
- **protocol** — протокол (tcp или udp, по умолчанию: tcp)

5.8.3.2 Примеры создания pool member

5.8.3.2.1 Пример 1. Создание pool member с адресом и портом

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443
```

```

Creating pool member 192.168.10.1:443...
Pool member 192.168.10.1:443 created successfully in cluster http_8080_web-api

```

5.8.3.2.2 Пример 2. Создание pool member с весом

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443 100
```

```

Creating pool member 192.168.10.1:443...
Pool member 192.168.10.1:443 created successfully in cluster http_8080_web-api

```

5.8.3.2.3 Пример 3. Создание UDP pool member

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443 1 udp
```

```

Creating pool member 192.168.10.1:443...
Pool member 192.168.10.1:443 created successfully in cluster http_8080_web-api

```

5.8.3.2.4 Пример 4. Создание pool member с FQDN

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create api.example.com 443
```



```
Creating pool member api.example.com:443...
Pool member api.example.com:443 created successfully in cluster http_8080_web-api
```

5.8.3.2.5 Пример 5. Создание pool member с IPv6 адресом

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 2001:db8::1 443
```

```
Creating pool member 2001:db8::1:443...
Pool member 2001:db8::1:443 created successfully in cluster http_8080_web-api
```

5.8.3.3 Обработка ошибок при создании pool member

5.8.3.3.1 Недостаточно параметров

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1
```

```
Usage: --create <address> <port> [weight] [<protocol>]
  address: FQDN or IP4/IPV5 address
  port: number from 1 to 65535
  weight: number from 1 to 4294967295, default: 1
  protocol: tcp|udp
```

```
Example: --create 192.168.10.1 443 1
```

```
Example: --create 192.168.10.1 443 100 udp
```

5.8.3.3.2 Неверный адрес

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create invalid-address 443
```

```
Invalid address: address must be a valid IPv4, IPv6, or FQDN
Address must be a valid IPv4, IPv6, or FQDN
```

5.8.3.3.3 Неверный порт

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 70000
```

```
Invalid port: port must be between 1 and 65535, got 70000
Port must be a number between 1 and 65535
```

5.8.3.3.4 Неверный протокол

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443 1
http
```

```
Invalid protocol: protocol must be 'tcp' or 'udp', got 'http'
Protocol must be 'tcp' or 'udp'
```

5.8.3.3.5 Неверный вес

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443
5000000000
```

```
Invalid weight: weight must be between 1 and 4294967295, got 5000000000
Weight must be a number between 1 and 4294967295
```

5.8.3.3.6 Дублирующийся pool member

```
flxgate-host#nodes.node1.upstream.http_8080_web-api.pools>--create 192.168.10.1 443
```

```
Error: pool member with address 192.168.10.1 and port 443 already exists
```

5.8.4 Валидация параметров

5.8.4.1 Адрес (address)

Адрес может быть:

- **IPv4 адрес:** например, 192.168.10.1
- **IPv6 адрес:** например, 2001:db8::1
- **FQDN:** например, api.example.com

FQDN должен: - Содержать хотя бы одну точку - Состоять из букв, цифр, точек и дефисов - Не начинаться и не заканчиваться точкой или дефисом

5.8.4.2 Порт (port)

Порт должен быть числом от 1 до 65535.

5.8.4.3 Вес (weight)

Вес должен быть числом от 1 до 4294967295. По умолчанию используется значение 1.

5.8.4.4 Протокол (protocol)

Протокол может быть: - **tcp** (по умолчанию) - **udp**

5.8.5 Значения по умолчанию

При создании нового pool используются следующие значения по умолчанию:

- **weight:** 1
- **protocol:** TCP
- **health_check_config.port_value:** равен указанному порту

5.8.6 Доступные команды

5.8.6.1 Команды в меню pools

- **--create** — создание нового pool
- **--show** — просмотр полной конфигурации всех pools в формате JSON
- **--back** — возврат в меню upstream кластера
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.8.7 Структура Pool

Каждый pool содержит следующую информацию:

- **address** — адрес pool member (IP или FQDN)
- **port_value** — номер порта
- **protocol** — протокол (TCP или UDP)

- **load_balancing_weight** — вес для балансировки нагрузки
- **health_check_config** — конфигурация health check для pool member

5.8.8 Примечания

- Pool members с одинаковым адресом и портом не могут быть созданы дважды в одном кластере
- Вес pool member влияет на балансировку нагрузки при использовании алгоритмов, которые учитывают вес
- Health check для pool member использует тот же порт, что и сам pool member (если не указано иное)

5.9 Управление downstream

Интерфейс командной строки позволяет управлять настройками приёма входящих подключений, инициируемых от клиентов к flxGATE Data Plane.

После запуска и авторизации в CLI перейдите в меню узла и введите **downstream**:

```
flxgate-host#nodes.node1>downstream
```

```
flxgate-host#nodes.node1.downstream>
```

5.9.1 Получение списка downstream listener'ов

Находясь в меню downstream, нажмите клавишу табуляции:

```
flxgate-host#nodes.node1.downstream>
http_8080_web-api https_443_api --create --back --exit --help --clear
```

Будут отображены доступные downstream listener'ы и команды управления.

Команды управления имеют префикс '--'.

Важно:

Режим редактирования конфигурации должен быть включен перед созданием, удалением, внесением изменений в конфигурацию listener.

После внесения изменений необходимо опубликовать изменения в конфигурацию.

Команды управления конфигурацией (**--conf_on**, **--conf_off**, **--conf_publish**) доступны как в меню списка downstream, так и в меню конкретного listener.

5.9.2 Создание downstream listener

Для создание нового downstream listener необходимо перевести узел обработки данных в режим редактирования.

Для создания нового downstream listener введите команду **--create** с указанием протокола, порта и имени сервиса:

```
flxgate-host#nodes.node1.downstream>--create http 8080 web-api↵
```

```
Creating listener http_8080_web-api...
Listener http_8080_web-api created successfully
flxgate-host#nodes.node1.downstream>
```

5.9.2.1 Параметры создания downstream listener

Команда **--create** принимает следующие параметры:

- **protocol** — протокол: `https`, `http`, `tcp`, `udp`, `grpc`
- **port** — номер порта
- **service** — имя сервиса

Имя downstream listener формируется автоматически в формате: `{protocol}_{port}_{service}`

5.9.2.2 Примеры создания downstream listener

Пример 1. Создание TCP downstream listener:

```
flxgate-host#nodes.node1.downstream>--create tcp 80 proxy↵
```

```
Creating listener tcp_80_proxy...
Listener tcp_80_proxy created successfully
flxgate-host#nodes.node1.downstream>
```

Пример 2. Создание HTTPS downstream listener:

```
flxgate-host#nodes.node1.downstream>--create https 443 api↵
```

```
Creating listener https_443_api...
Listener https_443_api created successfully
flxgate-host#nodes.node1.downstream>
```

5.9.2.3 Обработка ошибок при создании downstream listener

При попытке создать downstream listener с уже существующим именем:

```
flxgate-host#nodes.node1.downstream>--create http 8080 web-api↵
```

```
Creating listener http_8080_web-api...
Error: Listener http_8080_web-api already exists
flxgate-host#nodes.node1.downstream>
```

При указании неверных параметров:

```
flxgate-host#nodes.node1.downstream>--create http↵
```

```
Usage: --create <protocol> <port> <service>
  protocol: https|http|tcp|udp|grpc
  port: port number
  service: service name
```

```
Example: --create http 8080 web-api
flxgate-host#nodes.node1.downstream>
```

5.9.3 Просмотр информации о downstream listener

Введите название downstream listener и нажмите клавишу ввода:

```
flxgate-host#nodes.node1.downstream>http_8080_web-api↵
```

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>
```

5.9.3.1 Просмотр полной конфигурации downstream listener

Для просмотра полной конфигурации downstream listener в формате JSON используйте команду **--show**:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>--show
{
  "name": "http_8080_web-api",
  "stat_prefix": "http_8080_web-api",
  "address": {
    "address": "0.0.0.0",
    "port_value": 8080
  },
  "filter_chains": [
    {
      "name": "default_http_web-api",
      "filters": []
    }
  ],
  "listener_filters": []
}
flxgate-host#nodes.node1.downstream.http_8080_web-api>
```

5.9.4 Настройка атрибутов downstream listener

Находясь в меню конкретного downstream listener, доступны следующие атрибуты для настройки:

5.9.4.1 Адрес и порт (address)

Настройка адреса и порта, на котором downstream listener будет принимать соединения:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>address
```

```
current 'address' attribute value: 0.0.0.0:8080
```

Введите новый адрес в формате IP:PORT или нажмите Enter для сохранения текущего значения.

5.9.4.2 Другие атрибуты

Доступны следующие атрибуты для настройки:

- **per_connection_buffer_limit_bytes** — soft лимит размера буферов чтения и записи для новых соединений. Значение по-умолчанию - 1 MiB.
- **listener_filters_timeout** — таймаут ожидания завершения операций всех фильтров listener объекта
- **continue_on_listener_filters_timeout** — создавать соединение при таймауте фильтров listener объект
- **freebind** — установка опции сокета IP_FREEBIND
- **tcp_fast_open_queue_length** — длина очереди для соединений TCP Fast Open
- **enable_reuse_port** — включение опции сокета SO_REUSEPORT
- **tcp_backlog_size** — максимальная длина очереди ожидающих соединений TCP listener объект

- **max_connections_to_accept_per_socket_event** — максимальное количество соединений для принятия из ядра на одно событие сокета
- **bind_to_port** — привязка listener объекта к порту
- **enable_mptcp** — включение MPTCP (multi-path TCP) на listener объекте
- **ignore_global_conn_limit** — игнорирование глобального лимита максимальных downstream соединений
- **bypass_overload_manager** — обход настроенных действий менеджера перегрузки

Для изменения атрибута введите его название и следуйте инструкциям:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>per_connection_buffer_limit_bytes↵
```

```
current 'per_connection_buffer_limit_bytes' attribute value: 1048576
```

Введите новое значение или нажмите Enter для сохранения текущего.

5.9.5 Управление фильтрами

Для управления фильтрами введите команду **filters**:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>filters↵
```

```
flxgate-host#nodes.node1.downstream.http_8080_web-api.filters>
```

Подробнее об управлении фильтрами см. в соответствующей документации.

5.9.6 Управление инспекторами трафика downstream listener

Для управления фильтрами listener введите команду **listener_filters**:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>listener_filters↵
```

```
flxgate-host#nodes.node1.downstream.http_8080_web-api.listener_filters>
```

Подробнее об управлении фильтрами см. в соответствующей документации.

5.9.7 Удаление listener

Для удаления listener необходимо перевести узел обработки данных в режим редактирования.

Для удаления listener введите команду **--delete**:

```
flxgate-host#nodes.node1.downstream.http_8080_web-api>--delete↵
```

```
Deleting listener http_8080_web-api...
```

```
Listener deleted successfully
```

```
flxgate-host#nodes.node1.downstream>
```

После удаления необходимо опубликовать сделанные изменения.

⚠ Внимание: Удаление listener невозможно, если узел заблокирован (находится в режиме конфигурации) или слушатель имеет активные зависимости.

5.9.8 Доступные команды в меню listener

5.9.8.1 Команды в меню списка downstream

- **--create** — создание нового listener
- **--back** — возврат в предыдущее меню

- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.9.8.2 Команды в меню конкретного listener

- **--show** — просмотр полной конфигурации listener в формате JSON
- **--delete** — удаление listener
- **--back** — возврат в меню списка downstream
- **--conf_on** — включение режима конфигурации
- **--conf_off** — выключение режима конфигурации
- **--conf_publish** — публикация конфигурации в data plane

5.10 Хранилище сертификатов

Хранилище сертификатов позволяет управлять TLS сертификатами и контекстами валидации, используемыми для защиты соединений в flxGATE.

5.10.1 Доступ к меню certs

Для управления сертификатами введите команду **certs** в корневом меню CLI:

```
flxgate-host#certsd
```

```
flxgate-host#certs>
```

5.10.2 Просмотр списка сертификатов

Находясь в меню certs, нажмите клавишу табуляции для просмотра доступных сертификатов:

```
flxgate-host#certs>
```

```
my-cert server-cert --dir --upload --back --exit --help --clear
```

Будут отображены все загруженные TLS сертификаты и команды управления.

5.10.3 Просмотр сертификатов в файловой системе

Для просмотра списка сертификатов, доступных в локальной файловой системе, используйте команду **--dir**:

```
flxgate-host#certs>--dird
```

FileName	Size	Created
-----	-----	-----
server.crt	2048	2024-01-15 10:30:45
server.key	1704	2024-01-15 10:30:45
ca.crt	1024	2024-01-15 10:30:40

Команда отображает: - **FileName** — имя файла сертификата - **Size** — размер файла в байтах
- **Created** — дата и время создания файла

5.10.3.1 Поддерживаемые форматы файлов

Система автоматически распознает файлы со следующими расширениями:

- .pem - .crt - .cert - .key - .cer

5.10.3.2 Загрузка сертификата на узел управления

Сертификаты могут быть загружены на узел управления в виде файлов по протоколу sftp / scp:

```
echo "put локальный_файл" | sftp flxgate@flxgate-host
scp локальный_файл flxgate@flxgate-host:
```

5.10.4 Загрузка сертификата в хранилище

Для загрузки нового TLS сертификата используйте команду **--upload**:

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: server.key
Private key password:
TLS certificates uploaded successfully with name: server.crt
```

5.10.4.1 Процесс загрузки

1. **Certificate chain filename** — введите имя файла сертификата (или полный путь, если файл находится вне стандартной директории)
2. **Private key filename** — введите имя файла приватного ключа
3. **Private key password** — введите пароль для приватного ключа (если ключ защищен паролем, иначе нажмите Enter)

5.10.4.2 Параметры загрузки

- **Certificate chain filename** — имя файла сертификата или полный путь к файлу
- **Private key filename** — имя файла приватного ключа или полный путь
- **Private key password** — пароль для расшифровки приватного ключа (опционально)

5.10.4.3 Имя сертификата

Имя сертификата в хранилище автоматически устанавливается равным имени файла сертификата. Например, если вы загружаете файл `server.crt`, сертификат будет сохранен с именем `server.crt`.

5.10.4.4 Примеры загрузки

5.10.4.4.1 Пример 1. Загрузка сертификата без пароля

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: server.key
Private key password:
TLS certificates uploaded successfully with name: server.crt
```

5.10.4.4.2 Пример 2. Загрузка сертификата с паролем

```
flxgate-host#certs>--upload↵
```



```
Certificate chain filename: secure-server.crt
Private key filename: secure-server.key
Private key password: *****
TLS certificates uploaded successfully with name: secure-server.crt
```

5.10.4.4.3 Пример 3. Загрузка сертификата с абсолютным путем

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: /home/user/certs/my-cert.crt
Private key filename: /home/user/certs/my-cert.key
Private key password:
TLS certificates uploaded successfully with name: my-cert.crt
```

5.10.4.5 Обработка ошибок при загрузке

5.10.4.5.1 Файл сертификата не найден

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: nonexistent.crt
Failed to certificate chain file "nonexistent.crt" - secret file not found:
/etc/ssl/certs/nonexistent.crt
```

5.10.4.5.2 Пустой файл сертификата

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: empty.crt
Empty certificate chain file content
```

5.10.4.5.3 Файл приватного ключа не найден

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: nonexistent.key
Failed to open private key file "nonexistent.key" - secret file not found:
/etc/ssl/certs/nonexistent.key
```

5.10.4.5.4 Пустой файл приватного ключа

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: empty.key
Empty private key file content
```

5.10.4.5.5 Сертификат уже существует

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: server.key
Private key password:
Error getting TLS certificates: TLS certificate already exists
```

5.10.5 Просмотр информации о сертификате

Для просмотра информации о конкретном сертификате введите его имя:

```
flxgate-host#certs>server.crt↵
```

```
flxgate-host#certs.server.crt>
```

В меню сертификата доступны команды для просмотра и управления.

5.10.6 Управление сертификатом

После ввода имени сертификата вы переходите в меню управления этим сертификатом. Доступные команды зависят от реализации и могут включать:

- Просмотр детальной информации о сертификате
- Обновление сертификата
- Удаление сертификата

5.10.7 Типы сертификатов

Система поддерживает два типа секретов:

5.10.7.1 TLS Certificate


TLS сертификат содержит: - **Certificate chain** — цепочка сертификатов (сертификат сервера и промежуточные CA) - **Private key** — приватный ключ сервера - **Password** — пароль для приватного ключа (опционально)

Используется для защиты TLS соединений на стороне сервера (downstream listeners).

5.10.7.2 TLS Validation Context

Контекст валидации TLS содержит: - **Trusted CA** — доверенный корневой сертификат CA - **Allow expired certificate** — разрешить использование истекших сертификатов (опционально) - **CRL** — список отозванных сертификатов (опционально)

Используется для валидации сертификатов upstream серверов.

 **Примечание:** В текущей версии CLI поддерживается загрузка только TLS сертификатов через команду `--upload`. Контексты валидации можно создавать через API.

5.10.8 Использование сертификатов

После загрузки сертификаты доступны для использования в:

- **Downstream listeners** — для защиты входящих соединений
- **Upstream clusters** — для валидации сертификатов upstream серверов

При настройке listener или cluster укажите имя сертификата в соответствующих параметрах конфигурации.

5.10.9 Хранение сертификатов

Сертификаты хранятся в Control Plane Storage.

5.10.10 Безопасность

5.10.10.1 Рекомендации

1. **Защита приватных ключей:** Убедитесь, что файлы приватных ключей имеют ограниченные права доступа (например, 600)
2. **Безопасное хранение:** Не храните пароли от приватных ключей в открытом виде
3. **Регулярное обновление:** Обновляйте сертификаты до истечения срока их действия
4. **Ротация сертификатов:** Используйте процесс ротации сертификатов для минимизации простоев

5.10.11 Доступные команды

5.10.11.1 Команды в меню certs

- **{certificate_name}** — переход в меню управления сертификатом
- **--dir** — просмотр списка сертификатов в файловой системе
- **--upload** — загрузка нового TLS сертификата
- **--back** — возврат в корневое меню
- **--exit** — выход из CLI
- **--help** — справка по командам
- **--clear** — очистка экрана

5.10.12 Примеры использования

5.10.12.1 Полный цикл работы с сертификатом

1. Загрузка сертификата и ключа на файловую систему

```
echo "put server.crt" | sftp flxgate@flxgate-host
echo "put server.key" | sftp flxgate@flxgate-host
```

2. Просмотр доступных сертификатов в файловой системе:

```
flxgate-host#certs>--dir↵
```

FileName	Size	Created
server.crt	2048	2024-01-15 10:30:45
server.key	1704	2024-01-15 10:30:45

3. Загрузка сертификата в хранилище:

```
flxgate-host#certs>--upload↵
```

```
Certificate chain filename: server.crt
Private key filename: server.key
Private key password:
TLS certificates uploaded successfully with name: server.crt
```

4. Проверка загруженного в хранилище сертификата:

```
flxgate-host#certs>
server.crt --dir --upload --back --exit --help --clear
```

Сертификат `server.crt` теперь доступен в меню и может быть использован при настройке downstream или upstream.

5.11 Управление RHI (Route Health Injection)

Интерфейс командной строки позволяет управлять настройками RHI (Route Health Injection) для динамического управления маршрутами на основе состояния upstream кластеров.

После запуска и авторизации в CLI введите **rhi**:

```
flxgate-host#rhi
```

```
flxgate-host#rhi>
```

5.11.1 Обзор структуры меню RHI

Меню RHI организовано в иерархическую структуру:

```
rhi
├── api          # Настройки API для опроса здоровья кластеров
├── bgp          # Настройки BGP правил
└── upstreams   # Настройки порогов состояний для кластеров
```

5.11.2 Настройки API

Меню **api** содержит параметры для опроса состояния upstream кластеров через API.

5.11.2.1 Переход в меню API

```
flxgate-host#rhi>api
```

```
flxgate-host#rhi.api>
```

5.11.2.2 Настройка URL API

Для настройки URL API сервера введите команду **url**:

```
flxgate-host#rhi.api>url
```

```
current 'url' attribute value: http://flxgate-host:8080/
```

Введите новый URL или нажмите Enter для сохранения текущего значения.

Пример:

```
flxgate-host#rhi.api>url
```

```
current 'url' attribute value: http://flxgate-host:8080/
```

```
> http://new-host:8080/
```

```
API attributes updated successfully
```

```
current 'url' attribute value: http://new-host:8080/
```

5.11.2.3 Настройка интервала опроса

Для настройки интервала опроса здоровья upstream-кластеров введите команду **interval_sec**:

```
flxgate-host#rhi.api>interval_sec
```

```
current 'interval_sec' attribute value: 5
```

Введите новое значение в секундах или нажмите Enter для сохранения текущего.

Пример:

```
flxgate-host#rhi.api>interval_sec↵
```

```
current 'interval_sec' attribute value: 5
```

```
> 10
```

```
API attributes updated successfully
```

```
current 'interval_sec' attribute value: 10
```

5.11.2.4 Настройка timeout

Для настройки timeout ожидания ответа от API введите команду **timeout_sec**:

```
flxgate-host#rhi.api>timeout_sec↵
```

```
current 'timeout_sec' attribute value: 2
```

Введите новое значение в секундах или нажмите Enter для сохранения текущего.

5.11.3 Настройки BGP

Меню **bgp** содержит настройки BGP правил для управления маршрутами.

5.11.3.1 Переход в меню BGP

```
flxgate-host#rhi>bgp↵
```

```
flxgate-host#rhi.bgp>
```

5.11.3.2 Настройка режима при старте

Для настройки режима анонсирования маршрутов при старте введите команду **promote_on_start**:

```
flxgate-host#rhi.bgp>promote_on_start↵
```

```
current 'promote_on_start' attribute value: permit
```

Доступные значения: - **permit** — разрешить анонсирование маршрутов при старте - **deny** — запретить анонсирование маршрутов при старте - **none** — не выполнять никаких действий при старте

Пример:

```
flxgate-host#rhi.bgp>promote_on_start↵
```

```
current 'promote_on_start' attribute value: permit
```

```
> deny
```

```
BGP attributes updated successfully
```

```
current 'promote_on_start' attribute value: deny
```

5.11.3.3 Управление BGP правилами

Для управления BGP правилами перейдите в меню **rules**:

```
flxgate-host#rhi.bgp>rules↵
```

```
flxgate-host#rhi.bgp.rules>
```

5.11.3.3.1 Получение списка BGP правил

Находясь в меню правил, нажмите клавишу табуляции:

```
flxgate-host#rhi.bgp.rules>
bgp1 bgp2 --create --back --exit --help --clear
Будут отображены доступные BGP правила и команды управления.
```

5.11.3.3.2 Создание BGP правила

Для создания нового BGP правила введите команду **--create** с указанием параметров:

```
flxgate-host#rhi.bgp.rules>--create bgp3 172.16.1.3/32 15 SERVICES_OUT least_one
Creating BGP rule bgp3...
BGP rule bgp3 created successfully
flxgate-host#rhi.bgp.rules>
```

Параметры команды --create:

- **name** — имя правила (например: bgp3)
- **vip** — VIP-адрес в формате CIDR (например: 172.16.1.3/32)
- **seq** — sequence номер префикс листа (например: 15)
- **route_map** — имя route map (например: SERVICES_OUT)
- **unhealthy_mode** — режим перехода в аварийное состояние: `all`, `least_one`, `none`

Режимы unhealthy_mode: - **all** — переход в аварийное состояние при недоступности всех upstream кластеров - **least_one** — переход в аварийное состояние при недоступности хотя бы одного upstream кластера - **none** — никогда не переходить в аварийное состояние

Примеры создания BGP правил:

Пример 1. Создание правила с режимом "все кластеры":

```
flxgate-host#rhi.bgp.rules>--create bgp4 172.16.1.4/32 20 SERVICES_OUT all
Creating BGP rule bgp4...
BGP rule bgp4 created successfully
```

Пример 2. Создание правила с режимом "никогда":

```
flxgate-host#rhi.bgp.rules>--create bgp5 172.16.1.5/32 25 SERVICES_OUT none
Creating BGP rule bgp5...
BGP rule bgp5 created successfully
```

5.11.3.3.3 Просмотр и настройка BGP правила

Для просмотра и настройки конкретного BGP правила введите его имя:

```
flxgate-host#rhi.bgp.rules>bgp1
flxgate-host#rhi.bgp.rules.bgp1>
```

[Просмотр полной конфигурации правила](#)

Для просмотра полной конфигурации BGP правила в формате JSON используйте команду **--show**:

```
flxgate-host#rhi.bgp.rules.bgp1>--show
```

```
{
  "name": "bgp1",
  "vip": "172.16.1.1/32",
  "seq": 5,
  "route_map": "SERVICES_OUT",
  "unhealthy_mode": "least_one",
  "upstreams": [
    "http_backend_cluster_rr",
    "http_backend_cluster_lr"
  ]
}
```

Настройка атрибутов BGP правила

Доступны следующие атрибуты для настройки:

- **name** — имя правила
- **vip** — VIP-адрес в формате CIDR
- **seq** — sequence номер префикс листа
- **route_map** — имя route map
- **unhealthy_mode** — режим перехода в аварийное состояние

Пример настройки VIP:

```
flxgate-host#rhi.bgp.rules.bgp1>vip␣
current 'vip' attribute value: 172.16.1.1/32
> 172.16.1.10/32
BGP rule attributes updated successfully
current 'vip' attribute value: 172.16.1.10/32
```

Пример настройки sequence:

```
flxgate-host#rhi.bgp.rules.bgp1>seq␣
current 'seq' attribute value: 5
> 7
BGP rule attributes updated successfully
current 'seq' attribute value: 7
```

Управление кластерами в BGP правиле

Для управления списком отслеживаемых кластеров перейдите в меню **upstreams**:

```
flxgate-host#rhi.bgp.rules.bgp1>upstreams␣
flxgate-host#rhi.bgp.rules.bgp1.upstreams>
```

Добавление кластера:

```
flxgate-host#rhi.bgp.rules.bgp1.upstreams>--add http_backend_cluster_rr␣
Cluster http_backend_cluster_rr added to rule bgp1
```

Удаление кластера:

```
flxgate-host#rhi.bgp.rules.bgp1.upstreams>--remove http_backend_cluster_rr␣
Cluster http_backend_cluster_rr removed from rule bgp1
```

Просмотр списка кластеров:

Нажмите клавишу табуляции для просмотра списка кластеров:

```
flxgate-host#rhi.bgp.rules.bgp1.upstreams>
http_backend_cluster_rr http_backend_cluster_lr --add --remove --back --exit --help
--clear
```

5.11.3.3.4 Удаление BGP правила

Для удаления BGP правила введите команду **--delete**:

```
flxgate-host#rhi.bgp.rules.bgp1>--delete^
```

```
Deleting BGP rule bgp1...
BGP rule bgp1 deleted successfully
flxgate-host#rhi.bgp.rules>
```

⚠ **Внимание:** Удаление BGP правила невозможно, если оно используется в активной конфигурации.

5.11.3.4 Обработка ошибок при работе с BGP правилами

Ошибка при создании правила с существующим именем:

```
flxgate-host#rhi.bgp.rules>--create bgp1 172.16.1.1/32 5 SERVICES_OUT least_one^
```

```
Creating BGP rule bgp1...
Error: BGP rule bgp1 already exists
```

Ошибка при указании неверных параметров:

```
flxgate-host#rhi.bgp.rules>--create bgp3^
```

```
Usage: --create <name> <vip> <seq> <route_map> <unhealthy_mode>
name: rule name
vip: VIP address in CIDR format (e.g., 172.16.1.1/32)
seq: sequence number
route_map: route map name
unhealthy_mode: all|least_one|none
Example: --create bgp3 172.16.1.3/32 15 SERVICES_OUT least_one
```

5.11.4 Настройки порогов состояний upstream кластеров

Меню **upstreams** содержит настройки порогов состояний для отслеживаемых upstream кластеров.

5.11.4.1 Переход в меню upstream кластеров

```
flxgate-host#rhi>upstreams^
```

```
flxgate-host#rhi.upstreams>
```

5.11.4.2 Получение списка upstream кластеров

Находясь в меню upstream кластеров, нажмите клавишу табуляции:

```
flxgate-host#rhi.upstreams>
http_backend_cluster_rr http_backend_cluster_lr iperf_backend_cluster --add --back --
exit --help --clear
```

Будут отображены доступные кластеры и команды управления.

5.11.4.3 Добавление upstream кластера для отслеживания

Для добавления нового upstream кластера для отслеживания введите команду **--add**:


```
flxgate-host#rhi.upstreams>--add new_backend_cluster↵
```

```
Cluster new_backend_cluster added for monitoring
```

После добавления кластер появится в списке доступных upstream кластеров.

5.11.4.4 Настройка порогов состояний upstream кластера

Для настройки порогов состояний конкретного upstream кластера введите его имя:

```
flxgate-host#rhi.upstreams>http_backend_cluster_rr↵
```

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr>
```

5.11.4.4.1 Переход в меню порогов состояний

Для настройки порогов состояний перейдите в меню **healthy_states**:

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr>healthy_states↵
```

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states>
```

5.11.4.4.2 Настройка порога здорового состояния

Для настройки порога здорового состояния перейдите в меню **healthy**:

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states>healthy↵
```

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states.healthy>
```

Настройка threshold_percent (процент здоровых хостов):

```
flxgate-
host#rhi.upstreams.http_backend_cluster_rr.healthy_states.healthy>threshold_percent↵
```

```
current 'threshold_percent' attribute value: 100
```

```
> 80
```

```
Cluster health states updated successfully
```

```
current 'threshold_percent' attribute value: 80
```

Настройка penalty_period_sec (переходный период):

```
flxgate-
host#rhi.upstreams.http_backend_cluster_rr.healthy_states.healthy>penalty_period_sec↵
```

```
current 'penalty_period_sec' attribute value: 0
```

```
> 30
```

```
Cluster health states updated successfully
```

```
current 'penalty_period_sec' attribute value: 30
```

5.11.4.4.3 Настройка порога аварийного состояния

Для настройки порога аварийного состояния перейдите в меню **unhealthy**:

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states>unhealthy↵
```

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states.unhealthy>
```

Настройка threshold_percent (процент нездоровых хостов):

```
flxgate-
host#rhi.upstreams.http_backend_cluster_rr.healthy_states.unhealthy>threshold_percent↵
```

```
current 'threshold_percent' attribute value: 100
```

```
> 50
```

```
Cluster health states updated successfully
```

```
current 'threshold_percent' attribute value: 50
```

Настройка `penalty_period_sec` (переходный период):

```
flxgate-
```

```
host#rhi.upstreams.http_backend_cluster_rr.healthy_states.unhealthy>penalty_period_sec
```

```
current 'penalty_period_sec' attribute value: 0
```

```
> 60
```

```
Cluster health states updated successfully
```

```
current 'penalty_period_sec' attribute value: 60
```

5.11.4.4 Просмотр конфигурации порогов состояний

Для просмотра полной конфигурации порогов состояний кластера используйте команду **--show**:

```
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states>--show
```

```
{
  "healthy": {
    "threshold_percent": 100,
    "penalty_period_sec": 0
  },
  "unhealthy": {
    "threshold_percent": 100,
    "penalty_period_sec": 0
  }
}
```

5.11.5 Доступные команды

5.11.5.1 Команды в главном меню RHI

- **api** — переход в меню настройки API
- **bgp** — переход в меню настройки BGP правил
- **upstreams** — переход в меню настройки порогов состояний upstream кластеров
- **--back** — возврат в предыдущее меню
- **--exit** — выход из CLI
- **--help** — справка
- **--clear** — очистка экрана

5.11.5.2 Команды в меню BGP правил

- **--create** — создание нового BGP правила
- **--back** — возврат в меню BGP
- **--show** — просмотр полной конфигурации (в меню конкретного правила)

5.11.5.3 Команды в меню upstream кластеров

- **--add** — добавление кластера для отслеживания
- **--back** — возврат в главное меню RHI
- **--show** — просмотр конфигурации порогов состояний (в меню `healthy_states`)

5.11.6 Примеры использования

5.11.6.1 Пример 1: Полная настройка RHI

```
# 1. Настройка API
flxgate-host#rhi>api↵
flxgate-host#rhi.api>url↵
current 'url' attribute value: http://flxgate-host:8080/
> http://flxgate-host:8080/↵
flxgate-host#rhi.api>interval_sec↵
current 'interval_sec' attribute value: 5
> 10↵
flxgate-host#rhi.api>--back↵

# 2. Создание BGP правила
flxgate-host#rhi>bgp↵
flxgate-host#rhi.bgp>rules↵
flxgate-host#rhi.bgp.rules>--create bgp1 172.16.1.1/32 5 SERVICES_OUT least_one↵
Creating BGP rule bgp1...
BGP rule bgp1 created successfully

# 3. Добавление кластеров в правило
flxgate-host#rhi.bgp.rules>bgp1↵
flxgate-host#rhi.bgp.rules.bgp1>upstreams↵
flxgate-host#rhi.bgp.rules.bgp1.upstreams>--add http_backend_cluster_rr↵
Cluster http_backend_cluster_rr added to rule bgp1
flxgate-host#rhi.bgp.rules.bgp1.upstreams>--add http_backend_cluster_lr↵
Cluster http_backend_cluster_lr added to rule bgp1
flxgate-host#rhi.bgp.rules.bgp1.upstreams>--back↵

# 4. Настройка порогов для кластера
flxgate-host#rhi>upstreams↵
flxgate-host#rhi.upstreams>http_backend_cluster_rr↵
flxgate-host#rhi.upstreams.http_backend_cluster_rr>healthy_states↵
flxgate-host#rhi.upstreams.http_backend_cluster_rr.healthy_states>healthy↵
flxgate-
host#rhi.upstreams.http_backend_cluster_rr.healthy_states.healthy>threshold_percent↵
current 'threshold_percent' attribute value: 100
> 80↵
```

5.11.6.2 Пример 2: Просмотр конфигурации


```
# Просмотр конфигурации BGP правила
flxgate-host#rhi>bgp rules bgp1 --show↵
```


```
{
  "name": "bgp1",
  "vip": "172.16.1.1/32",
  "seq": 5,
  "route_map": "SERVICES_OUT",
  "unhealthy_mode": "least_one",
  "upstreams": [
    "http_backend_cluster_rr",
    "http_backend_cluster_lr"
  ]
}
```


```
# Просмотр порогов состояний кластера
flxgate-host#rhi>upstreams http_backend_cluster_rr healthy_states --show↵
```

```
{
  "healthy": {
    "threshold_percent": 80,
    "penalty_period_sec": 0
  },
  "unhealthy": {
    "threshold_percent": 100,
    "penalty_period_sec": 0
  }
}
```

5.11.7 Примечания

 **Совет:** Используйте клавишу табуляции для автодополнения и просмотра доступных команд и объектов.

 **Важно:** Изменения в конфигурации RNI применяются немедленно. Убедитесь, что все параметры настроены корректно перед сохранением.

 **Примечание:** Кластеры должны быть созданы в системе перед добавлением их в BGP правила или настройкой порогов состояний.

5.12 Управление пользователями

5.12.1 Обзор

CLI предоставляет интерактивную секцию для управления пользователями системы.

5.12.2 Структура меню

```
flxgate-host#users↵
```

При входе в секцию `users` открывается подменю для управления пользователями:

```
flxgate-host#users>
```

5.12.3 Команды меню

5.12.3.1 Список пользователей

При входе в секцию автоматически отображается список всех пользователей системы:

```
flxgate-host#users>
Username    Roles                Active    Created
admin       [admin]              true      2024-01-15 10:30:00
operator    [user]               true      2024-01-16 14:20:00
auditor1    [auditor]            true      2024-01-17 09:15:00
viewer      [auditor, user]      true      2024-01-18 11:45:00
```

5.12.3.2 Работа с конкретным пользователем

Для каждого пользователя создается команда с именем пользователя:

```
flxgate-host#users>admin↵
```

При выборе пользователя открывается подменю для управления этим пользователем:

```
flxgate-host#users.admin>
```

5.12.3.2.1 Команды в подменю пользователя

- **password** - изменить пароль пользователя
- Запрашивает новый пароль (с подтверждением)
- Пароль не отображается при вводе
- **roles** - управление ролями пользователя
- Отображает текущие роли
- Позволяет добавить/удалить роли
- Валидирует роли (только: admin, user, auditor)
- **active** - активировать/деактивировать пользователя
- Переключает статус **active** пользователя
- Деактивированные пользователи не могут войти в систему
- **--show** - показать полную информацию о пользователе
- Отображает: username, roles, active status, created date, last modified
- **--delete** - удалить пользователя
- Требуется подтверждения
- Удаляет пользователя из системы

5.12.3.3 Создание нового пользователя

```
flxgate-host#users> --create
```

Интерактивный процесс создания: 1. Запрос имени пользователя (username) 2. Запрос пароля (с подтверждением, скрытый ввод) 3. Выбор ролей (можно выбрать несколько): - **admin** - Администратор - **user** - Пользователь - **auditor** - Аудитор

Пример интерактивного создания:

```
flxgate-host#users> --create
Username: operator
Password: *****
Confirm password: *****
Select roles (comma-separated: admin,user,auditor): user
User 'operator' created successfully
```

5.12.3.4 Обновление пользователя

```
flxgate-host#users>admin
flxgate-host#users.admin> password
New password: *****
Confirm password: *****
Password updated successfully
```

```
flxgate-host#users.admin>roles
Current roles: [admin]
Available roles: admin, user, auditor
Enter roles (comma-separated): admin,user
Roles updated successfully
```

```
flxgate-host#users.admin>active
Current status: active
```

```
Toggle to inactive? (y/n): n
Status unchanged
```

5.12.3.5 Навигация

- `--back` - вернуться в предыдущее меню
- `--help` - показать справку по командам

5.12.4 Ролевая модель

5.12.4.1 Роли пользователей

Система поддерживает три роли с различными уровнями доступа:

5.12.4.1.1 1. admin (Администратор)

- **Полный доступ** ко всем операциям в системе
- Может управлять пользователями (создание, изменение, удаление)
- Может изменять конфигурацию узлов обработки трафика, upstream кластеров, downstream
- Может управлять сертификатами
- Может изменять сетевые настройки
- Доступ ко всем операциям чтения и записи

5.12.4.1.2 2. user (Пользователь)

- **Базовый доступ** для работы с конфигурацией
- Может просматривать конфигурацию (узлы обработки трафика, upstream кластеров, downstream)
- Может изменять конфигурацию (создание, изменение, удаление)
- Может управлять сертификатами
- **Не может** управлять пользователями
- **Не может** изменять системные настройки

5.12.4.1.3 3. auditor (Аудитор)

- **Только чтение** для мониторинга и аудита
- Может просматривать конфигурацию (узлы обработки трафика, upstream кластеров, downstream)
- Может просматривать список пользователей (без паролей)
- Может просматривать сертификаты и секреты
- **Не может** изменять конфигурацию
- **Не может** управлять пользователями
- **Не может** создавать или удалять ресурсы

5.12.4.2 Множественные роли

Пользователь может иметь несколько ролей одновременно. Права доступа объединяются (OR логика):

- ["admin"] - только администратор (полный доступ)
- ["user"] - только пользователь (базовый доступ)
- ["auditor"] - только аудитор (только чтение)
- ["admin", "user"] - администратор и пользователь (полный доступ, избыточно)
- ["user", "auditor"] - пользователь и аудитор (базовый доступ + чтение)
- ["admin", "auditor"] - администратор и аудитор (полный доступ)

Примечание: Роль `admin` всегда дает полный доступ, независимо от других ролей.

5.12.4.3 Валидация ролей

- При создании пользователя требуется указать хотя бы одну роль
- Допускаются только роли: `admin`, `user`, `auditor`
- При обновлении ролей необходимо указать все роли, которые должны быть у пользователя
- Роли валидируются на стороне сервера

5.12.5 Примеры использования

5.12.5.1 Создание администратора

```
flxgate-host#users> --create
Username: admin
Password: *****
Confirm password: *****
Select roles: admin
User 'admin' created successfully
```

5.12.5.2 Создание оператора с правами пользователя

```
flxgate-host#users> --create
Username: operator
Password: *****
Confirm password: *****
Select roles: user
User 'operator' created successfully
```

5.12.5.3 Создание аудитора

```
flxgate-host# users> --create
Username: auditor1
Password: *****
Confirm password: *****
Select roles: auditor
User 'auditor1' created successfully
```

5.12.5.4 Изменение ролей пользователя

```
flxgate-host# users>operator<
flxgate-host# users.operator>roles
Current roles: [user]
Available roles: admin, user, auditor
```

```
Enter roles (comma-separated): user,auditor
Roles updated successfully
```

5.12.5.5 Деактивация пользователя

```
flxgate-host# users>operator↵
flxgate-host# users.operator> active
Current status: active
Toggle to inactive? (y/n): y
User 'operator' deactivated successfully
```

5.12.5.6 Просмотр информации о пользователе

```
flxgate-host# users>admin↵
flxgate-host# users.admin>--show↵
Username: admin
Roles: [admin]
Active: true
Created: 2024-01-15 10:30:00
Last Modified: 2024-01-20 15:45:00
```

5.12.6 Удаление пользователя

```
flxgate-host# users>operator↵
flxgate-host# users.operator>--delete↵
Are you sure you want to delete user 'operator'? (y/n): y
User 'operator' deleted successfully
```

5.13 🛡 Application Programming Interface (API)

flxGATE предоставляет REST API для управления конфигурацией системы. API использует JSON для обмена данными и поддерживает JWT авторизацию.

5.13.1 Swagger документация

API полностью документирован через Swagger/OpenAPI спецификацию. Swagger UI доступен для интерактивного просмотра и тестирования API.

5.13.1.1 Доступ к Swagger UI

После запуска API сервера Swagger документация доступна по адресу:

- **Swagger UI:** <http://flxgate-host:10801/swagger/index.html>
- **JSON спецификация:** <http://flxgate-host:10801/swagger/doc.json>
- **YAML спецификация:** <http://flxgate-host:10801/swagger/swagger.yaml>

5.13.2 Базовый URL

По умолчанию API сервер запускается на адресе <http://flxgate-host:10801>.

5.13.3 Формат данных

API использует JSON для обмена данными. Все запросы и ответы должны иметь заголовок `Content-Type: application/json`.

5.13.4 Авторизация

API поддерживает JWT (JSON Web Token) авторизацию.

5.13.4.1 Использование токена

Для доступа к защищенным эндпоинтам добавьте JWT токен в заголовок запроса:

```
Authorization: Bearer <your-jwt-token>
```

5.13.5 Основные группы эндпоинтов

5.13.5.1 Аутентификация

- `POST /auth/login` — получение JWT токена
- `GET /auth/validate` — проверка валидности токена

5.13.5.2 Узлы обработки трафика

- `GET /nodes` — список всех узлов
- `GET /nodes/{node_id}` — информация об узле
- `POST /nodes/{node_id}` — создание узла
- `DELETE /nodes/{node_id}` — удаление узла
- `PUT /nodes/{node_id}/config/on` — включение режима конфигурации
- `PUT /nodes/{node_id}/config/off` — выключение режима конфигурации
- `POST /nodes/{node_id}/config/publish` — публикация конфигурации
- `GET /nodes/{node_id}/config/versions` — список версий конфигурации

5.13.5.3 Upstream кластеры

- `GET /nodes/{node_id}/clusters` — список кластеров узла
- `GET /nodes/{node_id}/clusters/{cluster_name}` — информация о кластере
- `POST /nodes/{node_id}/clusters` — создание кластера
- `PUT /nodes/{node_id}/clusters/{cluster_name}` — обновление кластера
- `DELETE /nodes/{node_id}/clusters/{cluster_name}` — удаление кластера

5.13.5.4 Downstream listeners

- `GET /nodes/{node_id}/listeners` — список listeners узла
- `GET /nodes/{node_id}/listeners/{lnr_name}` — информация о listener
- `POST /nodes/{node_id}/listeners` — создание listener
- `PUT /nodes/{node_id}/listeners/{lnr_name}` — обновление listener
- `DELETE /nodes/{node_id}/listeners/{lnr_name}` — удаление listener

5.13.5.5 Цепочки фильтров

- GET `/nodes/{node_id}/listeners/{lnr_name}/filter_chains` — список цепочек фильтров
- GET `/nodes/{node_id}/listeners/{lnr_name}/filter_chains/{filter_chain_name}` — информация о цепочке фильтров
- POST `/nodes/{node_id}/listeners/{lnr_name}/filter_chains` — создание цепочки фильтров
- PUT `/nodes/{node_id}/listeners/{lnr_name}/filter_chains/{filter_chain_name}` — обновление цепочки фильтров
- DELETE `/nodes/{node_id}/listeners/{lnr_name}/filter_chains/{filter_chain_name}` — удаление цепочки фильтров

5.13.5.6 Фильтры listeners

- GET `/nodes/{node_id}/listeners/{lnr_name}/listener_filters` — список фильтров listener
- GET `/nodes/{node_id}/listeners/{lnr_name}/listener_filters/{filter_name}` — информация о фильтре
- POST `/nodes/{node_id}/listeners/{lnr_name}/listener_filters` — создание фильтра
- PUT `/nodes/{node_id}/listeners/{lnr_name}/listener_filters/{filter_name}` — обновление фильтра
- DELETE `/nodes/{node_id}/listeners/{lnr_name}/listener_filters/{filter_name}` — удаление фильтра
- PUT `/nodes/{node_id}/listeners/{lnr_name}/listener_filters/{filter_name}/position` — изменение позиции фильтра

5.13.5.7 TLS сертификаты

- GET `/secrets` — список всех секретов
- GET `/secrets/tls_certificates` — список TLS сертификатов
- GET `/secrets/tls_certificates/{secret_name}` — информация о TLS сертификате
- POST `/secrets/tls_certificates` — создание TLS сертификата
- PUT `/secrets/tls_certificates/{secret_name}` — обновление TLS сертификата
- DELETE `/secrets/{secret_name}` — удаление секрета

5.13.5.8 Контексты валидации TLS

- GET /secrets/validation_contexts — список контекстов валидации
- GET /secrets/validation_contexts/{secret_name} — информация о контексте валидации
- POST /secrets/validation_contexts — создание контекста валидации
- PUT /secrets/validation_contexts/{secret_name} — обновление контекста валидации

5.13.5.9 Пользователи

- GET /users — список пользователей
- GET /users/{username} — информация о пользователе
- POST /users — создание пользователя
- PUT /users/{username} — обновление пользователя
- DELETE /users/{username} — удаление пользователя

5.13.6 Коды ответов

API использует стандартные HTTP коды ответов:

- 200 OK — успешный запрос
- 201 Created — ресурс успешно создан
- 204 No Content — успешное удаление (без тела ответа)
- 400 Bad Request — неверный запрос
- 401 Unauthorized — требуется авторизация
- 403 Forbidden — недостаточно прав доступа
- 404 Not Found — ресурс не найден
- 409 Conflict — конфликт (например, ресурс уже существует)
- 500 Internal Server Error — внутренняя ошибка сервера

5.13.7 Формат ошибок

При возникновении ошибки API возвращает ответ в следующем формате:

```
{
  "errors": [
    {
      "message": "Описание ошибки"
    }
  ]
}
```

5.13.8 Примеры использования

5.13.8.1 Получение списка узлов

```
curl -X GET http://flxgate-host:10801/nodes \
```

```
-H "Authorization: Bearer YOUR_JWT_TOKEN" \  
-H "Content-Type: application/json"
```

5.13.8.2 Создание узла

```
curl -X POST http://flxgate-host:10801/nodes/node1 \  
-H "Authorization: Bearer YOUR_JWT_TOKEN" \  
-H "Content-Type: application/json"
```

5.13.8.3 Получение информации о кластере

```
curl -X GET http://flxgate-host:10801/nodes/node1/clusters/http_8080_web-api \  
-H "Authorization: Bearer YOUR_JWT_TOKEN" \  
-H "Content-Type: application/json"
```

5.13.9 Дополнительная информация

Для полной документации всех эндпоинтов, параметров запросов и ответов используйте Swagger UI по адресу <http://flxgate-host:10801/swagger/index.html>.

6 Маршрутизация

6.1 BGP (Border Gateway Protocol)

- Режим Active-Active
- Режим Active-Standby

6.2 RHI (Route Health Injection)

RHI (Route Health Injection) позволяет на основании данных о состоянии upstream кластеров динамически управлять маршрутами, анонсируемыми протоколами BGP или OSPF.

Управление маршрутами осуществляется с помощью автоматического отзыва/восстановления префиксов при изменении состояния upstream кластеров.

Для настройки RHI необходимо выполнить следующие шаги:

6.2.1 Выбор upstream кластеров

Выберите upstream кластера, состояния которых будут отслеживаться для управления маршрутами.

6.2.2 Настройка порогов состояний

Для каждого upstream кластера укажите пороги состояний **healthy_states**:

Порог нормального (**healthy**) состояния:

- **threshold_percent** — процент здоровых хостов для нормального состояния
- **penalty_period_sec** — переходный период в секундах

Порог аварийного (**unhealthy**) состояния:

- **threshold_percent** — процент нездоровых хостов для аварийного состояния
- **penalty_period_sec** — переходный период в секундах

6.2.3 Настройка BGP правил

Настройте BGP правила, указав для каждого из них

- Соответствующий список upstream кластеров, выбранных на шаге 1
- VIP-адрес
- Sequence номер префикс листа
- Режим перехода в аварийное состояние (никогда, по недоступности хотя бы одного upstream кластера, по недоступности всех upstream кластеров)

7 Мониторинг

7.1 Метрики

Описание всех доступных Prometheus метрик, которые экспортируются FlxGate DataPlane через эндпоинт `/stats/prometheus`.

7.1.1 Содержание раздела

1. Метрики сервера
2. Метрики слушателей (Listeners)
3. Метрики HTTP Connection Manager
4. Метрики кластеров (Clusters)
5. Метрики TLS/SSL
6. Метрики TCP
7. Метрики QUIC
8. Метрики менеджера слушателей
9. Метрики менеджера кластеров

7.1.2 Метрики сервера

Метрики сервера находятся в пространстве имен `server.*` и описывают общее состояние экземпляра FlxGate DataPlane.

Метрика	Тип	Описание
<code>server.uptime</code>	Gauge	Текущее время работы сервера в секундах
<code>server.concurrency</code>	Gauge	Количество рабочих потоков
<code>server.memory_allocated</code>	Gauge	Текущий объем выделенной памяти в байтах. Сумма для нового и старого процессов FlxGate DataPlane при hot restart
<code>server.memory_heap_size</code>	Gauge	Текущий зарезервированный размер кучи в байтах. Размер кучи нового процесса FlxGate DataPlane при hot restart
<code>server.memory_physical_size</code>	Gauge	Текущая оценка общего количества байт физической памяти. Размер физической памяти нового процесса FlxGate DataPlane при hot restart
<code>server.live</code>	Gauge	1, если сервер не находится в режиме draining, иначе 0
<code>server.state</code>	Gauge	Текущее состояние сервера
<code>server.parent_connections</code>	Gauge	Общее количество соединений старого процесса FlxGate DataPlane при hot restart
<code>server.total_connections</code>	Gauge	Общее количество соединений для нового и старого процессов FlxGate DataPlane
<code>server.version</code>	Gauge	Целочисленный номер версии на основе SCM ревизии или <code>stats_server_version_override</code> , если установлен
<code>server.days_until_first_cert_expiring</code>	Gauge	Количество дней до истечения следующего управляемого сертификата
<code>server.seconds_until_first_ocsp_response_expiring</code>	Gauge	Количество секунд до истечения следующего управляемого OSCP ответа
<code>server.hot_restart_epoch</code>	Gauge	Текущая эпоха hot restart - целое число, переданное через флаг командной строки <code>--restart-epoch</code> , обычно указывающее поколение

Метрика	Тип	Описание
server.hot_restart_generation	Gauge	Текущее поколение hot restart - как hot_restart_epoch, но вычисляется автоматически путем инкремента от родителя
server.initialization_time_ms	Histogram	Общее время инициализации FlxGate DataPlane в миллисекундах. Время от запуска сервера до готовности рабочих потоков принимать новые соединения
server.debug_assertion_failures	Counter	Количество ошибок debug assertion, обнаруженных в release сборке, если скомпилировано с <code>--define log_debug_assert_in_release=enabled</code> , иначе 0
server.flxgate_bug_failures	Counter	Количество ошибок, обнаруженных в release сборке. Сообщите об ошибке, если это значение увеличивается, так как это может быть серьезно
server.static_unknown_fields	Counter	Количество сообщений в статической конфигурации с неизвестными полями
server.dynamic_unknown_fields	Counter	Количество сообщений в динамической конфигурации с неизвестными полями
server.wip_protos	Counter	Количество сообщений и полей, помеченных как work-in-progress, которые используются
server.compilation_settings.fips_mode	Gauge	Целое число, представляющее, является ли сборка FlxGate DataPlane FIPS совместимой или нет

7.1.3 Метрики слушателей (Listeners)

Метрики слушателей находятся в пространстве имен `listener.<address>.*` (или `listener.<stat_prefix>.*`, если `stat_prefix` не пустой).

7.1.3.1 Основные метрики слушателя

Метрика	Тип	Описание
listener.<address>.downstream_cx_total	Counter	Общее количество соединений
listener.<address>.downstream_cx_destroy	Counter	Общее количество уничтоженных соединений
listener.<address>.downstream_cx_active	Gauge	Общее количество активных соединений
listener.<address>.downstream_cx_length_ms	Histogram	Длительность соединения в миллисекундах
listener.<address>.downstream_cx_transport_socket_connect_timeout	Counter	Общее количество соединений, которые превысили таймаут во время переговоров транспортного сокета
listener.<address>.downstream_cx_overflow	Counter	Общее количество отклоненных соединений из-за ограничения лимита соединений слушателя
listener.<address>.downstream_cx_overload_reject	Counter	Общее количество отклоненных соединений из-за настроенных действий перегрузки
listener.<address>.downstream_global_cx_overflow	Counter	Общее количество отклоненных соединений из-за глобального лимита соединений
listener.<address>.connections_accepted_per_socket_event	Histogram	Количество соединений, принятых за событие сокета слушателя
listener.<address>.downstream_pre_cx_timeout	Counter	Сокеты, которые превысили таймаут во время обработки фильтров слушателя
listener.<address>.downstream_pre_cx_active	Gauge	Сокеты, которые в настоящее время проходят обработку фильтров слушателя
listener.<address>.extension_config_missing	Counter	Общее количество соединений, закрытых из-за отсутствия конфигурации расширения фильтра слушателя
listener.<address>.network_extension_config_missing	Counter	Общее количество соединений, закрытых из-за отсутствия конфигурации расширения сетевого фильтра
listener.<address>.no_filter_chain_match	Counter	Общее количество соединений, которые не соответствовали ни одной цепочке фильтров

Метрика	Тип	Описание
listener.<address>.downstream_listener_filter_remote_close	Counter	Общее количество соединений, закрытых удаленной стороной при реек данных для фильтров слушателя
listener.<address>.downstream_listener_filter_error	Counter	Общее количество ошибок чтения при реек данных для фильтров слушателя

7.1.3.2 Метрики по обработчикам (Per-handler)

Каждый слушатель имеет дополнительные метрики в пространстве имен listener.<address>.<handler>.*, где <handler> может быть main_thread, worker_0, worker_1 и т.д.

Метрика	Тип	Описание
listener.<address>.<handler>.downstream_cx_total	Counter	Общее количество соединений на этом обработчике
listener.<address>.<handler>.downstream_cx_active	Gauge	Общее количество активных соединений на этом обработчике

7.1.3.3 Метрики UDP

Метрика	Тип	Описание
listener.<address>.udp.downstream_rx_datagram_dropped	Counter	Количество датаграмм, отброшенных из-за переполнения ядра или усечения

7.1.4 Метрики HTTP Connection Manager

Метрики HTTP Connection Manager находятся в пространстве имен http.<stat_prefix>.*.

7.1.4.1 Соединения

Метрика	Тип	Описание
http.<stat_prefix>.downstream_cx_total	Counter	Общее количество соединений
http.<stat_prefix>.downstream_cx_ssl_total	Counter	Общее количество TLS соединений
http.<stat_prefix>.downstream_cx_http1_total	Counter	Общее количество HTTP/1.1 соединений
http.<stat_prefix>.downstream_cx_upgrades_total	Counter	Общее количество успешно обновленных соединений. Также учитываются как общее количество http1/http2 соединений
http.<stat_prefix>.downstream_cx_http2_total	Counter	Общее количество HTTP/2 соединений
http.<stat_prefix>.downstream_cx_http3_total	Counter	Общее количество HTTP/3 соединений
http.<stat_prefix>.downstream_cx_destroy	Counter	Общее количество уничтоженных соединений
http.<stat_prefix>.downstream_cx_destroy_remote	Counter	Общее количество соединений, уничтоженных из-за удаленного закрытия
http.<stat_prefix>.downstream_cx_destroy_local	Counter	Общее количество соединений, уничтоженных из-за локального закрытия
http.<stat_prefix>.downstream_cx_destroy_active_rq	Counter	Общее количество соединений, уничтоженных с 1+ активным запросом
http.<stat_prefix>.downstream_cx_destroy_local_active_rq	Counter	Общее количество соединений, уничтоженных локально с 1+ активным запросом
http.<stat_prefix>.downstream_cx_destroy_remote_active_rq	Counter	Общее количество соединений, уничтоженных удаленно с 1+ активным запросом
http.<stat_prefix>.downstream_cx_active	Gauge	Общее количество активных соединений
http.<stat_prefix>.downstream_cx_ssl_active	Gauge	Общее количество активных TLS соединений
http.<stat_prefix>.downstream_cx_http1_active	Gauge	Общее количество активных HTTP/1.1 соединений

Метрика	Тип	Описание
http.<stat_prefix>.downstream_cx_upgrades_active	Gauge	Общее количество активных обновленных соединений. Также учитываются как активные http1/http2 соединения
http.<stat_prefix>.downstream_cx_http1_soft_drain	Gauge	Общее количество активных HTTP/1.x соединений, ожидающих еще один downstream запрос для безопасного закрытия соединения
http.<stat_prefix>.downstream_cx_http2_active	Gauge	Общее количество активных HTTP/2 соединений
http.<stat_prefix>.downstream_cx_http3_active	Gauge	Общее количество активных HTTP/3 соединений
http.<stat_prefix>.downstream_cx_protocol_error	Counter	Общее количество ошибок протокола
http.<stat_prefix>.downstream_cx_length_ms	Histogram	Длительность соединения в миллисекундах
http.<stat_prefix>.downstream_cx_rx_bytes_total	Counter	Общее количество полученных байт
http.<stat_prefix>.downstream_cx_rx_bytes_buffered	Gauge	Общее количество полученных байт, в настоящее время буферизованных
http.<stat_prefix>.downstream_cx_tx_bytes_total	Counter	Общее количество отправленных байт
http.<stat_prefix>.downstream_cx_tx_bytes_buffered	Gauge	Общее количество отправленных байт, в настоящее время буферизованных
http.<stat_prefix>.downstream_cx_drain_close	Counter	Общее количество соединений, закрытых из-за draining
http.<stat_prefix>.downstream_cx_idle_timeout	Counter	Общее количество соединений, закрытых из-за таймаута простоя
http.<stat_prefix>.downstream_cx_max_duration_reached	Counter	Общее количество соединений, закрытых из-за максимальной длительности соединения
http.<stat_prefix>.downstream_cx_max_requests_reached	Counter	Общее количество соединений, закрытых из-за максимального количества запросов на соединение
http.<stat_prefix>.downstream_cx_overload_disable_keepalive	Counter	Общее количество соединений, для которых HTTP 1.x keepalive был отключен из-за перегрузки FlxGate DataPlane
http.<stat_prefix>.downstream_flow_control_paused_reading_total	Counter	Общее количество раз, когда чтение было отключено из-за управления потоком
http.<stat_prefix>.downstream_flow_control_resumed_reading_total	Counter	Общее количество раз, когда чтение было включено на соединении из-за управления потоком

7.1.4.2 Запросы

Метрика	Тип	Описание
http.<stat_prefix>.downstream_rq_total	Counter	Общее количество запросов
http.<stat_prefix>.downstream_rq_http1_total	Counter	Общее количество HTTP/1.1 запросов
http.<stat_prefix>.downstream_rq_http2_total	Counter	Общее количество HTTP/2 запросов
http.<stat_prefix>.downstream_rq_http3_total	Counter	Общее количество HTTP/3 запросов
http.<stat_prefix>.downstream_rq_active	Gauge	Общее количество активных запросов
http.<stat_prefix>.downstream_rq_rejected_via_ip_deduction	Counter	Общее количество запросов, отклоненных из-за неудачного определения исходного IP
http.<stat_prefix>.downstream_rq_response_before_rq_complete	Counter	Общее количество ответов, отправленных до завершения запроса
http.<stat_prefix>.downstream_rq_rx_reset	Counter	Общее количество полученных сбросов запросов
http.<stat_prefix>.downstream_rq_tx_reset	Counter	Общее количество отправленных сбросов запросов
http.<stat_prefix>.downstream_rq_non_relative_path	Counter	Общее количество запросов с неотносительным HTTP путем
http.<stat_prefix>.downstream_rq_too_large	Counter	Общее количество запросов, приведших к 413 из-за буферизации слишком большого тела
http.<stat_prefix>.downstream_rq_completed	Counter	Общее количество запросов, которые привели к ответу (например, не включает прерванные запросы)

Метрика	Тип	Описание
<code>http.<stat_prefix>.downstream_rq_failed_path_normalization</code>	Counter	Общее количество запросов, перенаправленных из-за разных исходного и нормализованного URL путей или когда нормализация пути не удалась
<code>http.<stat_prefix>.downstream_rq_1xx</code>	Counter	Общее количество 1xx ответов
<code>http.<stat_prefix>.downstream_rq_2xx</code>	Counter	Общее количество 2xx ответов
<code>http.<stat_prefix>.downstream_rq_3xx</code>	Counter	Общее количество 3xx ответов
<code>http.<stat_prefix>.downstream_rq_4xx</code>	Counter	Общее количество 4xx ответов
<code>http.<stat_prefix>.downstream_rq_5xx</code>	Counter	Общее количество 5xx ответов
<code>http.<stat_prefix>.downstream_rq_ws_on_non_ws_route</code>	Counter	Общее количество запросов на обновление, отклоненных маршрутами без обновления. Теперь применяется как к WebSocket, так и к не-WWebSocket обновлениям
<code>http.<stat_prefix>.downstream_rq_time</code>	Histogram	Общее время для запроса и ответа (миллисекунды)
<code>http.<stat_prefix>.downstream_rq_idle_timeout</code>	Counter	Общее количество запросов, закрытых из-за таймаута простоя
<code>http.<stat_prefix>.downstream_rq_max_duration_reached</code>	Counter	Общее количество запросов, закрытых из-за достижения максимальной длительности
<code>http.<stat_prefix>.downstream_rq_timeout</code>	Counter	Общее количество запросов, закрытых из-за таймаута на пути запроса
<code>http.<stat_prefix>.downstream_rq_overload_close</code>	Counter	Общее количество запросов, закрытых из-за перегрузки FlxGate DataPlane
<code>http.<stat_prefix>.downstream_rq_redirected_with_normalized_path</code>	Counter	Общее количество запросов, перенаправленных из-за разных исходного и нормализованного URL путей
<code>http.<stat_prefix>.downstream_rq_too_many_premature_resets</code>	Counter	Общее количество соединений, закрытых из-за слишком большого количества преждевременных сбросов запросов на соединении
<code>http.<stat_prefix>.rs_too_large</code>	Counter	Общее количество ошибок ответа из-за буферизации слишком большого тела

7.1.4.3 Метрики трассировки

Метрики трассировки находятся в пространстве имен `http.<stat_prefix>.tracing.*`:

Метрика	Тип	Описание
<code>http.<stat_prefix>.tracing.random_sampling</code>	Counter	Общее количество решений о трассировке по случайной выборке
<code>http.<stat_prefix>.tracing.service_forced</code>	Counter	Общее количество решений о трассировке по флагу runtime сервера <code>tracing.global_enabled</code>
<code>http.<stat_prefix>.tracing.client_enabled</code>	Counter	Общее количество решений о трассировке по заголовку запроса <code>x-force-trace</code>
<code>http.<stat_prefix>.tracing.not_traceable</code>	Counter	Общее количество решений о нетрассируемости по ID запроса
<code>http.<stat_prefix>.tracing.health_check</code>	Counter	Общее количество решений о нетрассируемости по health check

7.1.4.4 Метрики HTTP/1 codec

Метрики HTTP/1 codec находятся в пространстве имен `http1.*` (downstream) и `cluster.<name>.http1.*` (upstream):

Метрика	Тип	Описание
<code>http1.dropped_headers_with_underscores</code>	Counter	Общее количество отброшенных заголовков с именами, содержащими подчеркивания
<code>http1.metadata_not_supported_error</code>	Counter	Общее количество метаданных, отброшенных во время кодирования HTTP/1
<code>http1.response_flood</code>	Counter	Общее количество соединений, закрытых из-за переполнения ответов

Метрика	Тип	Описание
http1.requests_rejected_with_underscores_in_headers	Counter	Общее количество отклоненных запросов из-за имен заголовков, содержащих подчеркивания

7.1.4.5 Метрики HTTP/2 codec

Метрики HTTP/2 codec находятся в пространстве имен `http2.*` (downstream) и `cluster.<name>.http2.*` (upstream):

Метрика	Тип	Описание
http2.dropped_headers_with_underscores	Counter	Общее количество отброшенных заголовков с именами, содержащими подчеркивания
http2.goaway_sent	Counter	Общее количество кадров GOAWAY, которые были отправлены в кодек
http2.header_overflow	Counter	Общее количество сбросов соединений из-за заголовков, превышающих настроенное значение
http2.headers_cb_no_stream	Counter	Общее количество ошибок, когда callback заголовков вызывается без связанного потока
http2.inbound_empty_frames_flood	Counter	Общее количество соединений, завершенных из-за превышения лимита последовательных входящих кадров с пустой полезной нагрузкой и без флага end stream
http2.inbound_priority_frames_flood	Counter	Общее количество соединений, завершенных из-за превышения лимита входящих кадров типа PRIORITY
http2.inbound_window_update_frames_flood	Counter	Общее количество соединений, завершенных из-за превышения лимита входящих кадров типа WINDOW_UPDATE
http2.keepalive_timeout	Counter	Общее количество соединений, закрытых из-за таймаута keepalive
http2.metadata_empty_frames	Counter	Общее количество кадров метаданных, которые были получены и содержали пустые карты
http2.outbound_control_frames_active	Gauge	Общее количество активных исходящих управляющих кадров
http2.outbound_control_flood	Counter	Общее количество соединений, завершенных из-за превышения лимита исходящих кадров типов PING, SETTINGS и RST_STREAM
http2.outbound_frames_active	Gauge	Общее количество активных исходящих кадров
http2.outbound_flood	Counter	Общее количество соединений, завершенных из-за превышения лимита исходящих кадров всех типов
http2.requests_rejected_with_underscores_in_headers	Counter	Общее количество отклоненных запросов из-за имен заголовков, содержащих подчеркивания
http2.rx_messaging_error	Counter	Общее количество недопустимых полученных кадров, нарушивших раздел 8 спецификации HTTP/2
http2.rx_reset	Counter	Общее количество кадров сброса потока, полученных FlxGate DataPlane
http2.stream_refused_errors	Counter	Общее количество недопустимых кадров, полученных FlxGate DataPlane с кодом ошибки REFUSED_STREAM
http2.trailers	Counter	Общее количество трейлеров, замеченных на запросах, приходящих от downstream
http2.tx_flush_timeout	Counter	Общее количество таймаутов простоя потока, ожидающих открытия окна потока для сброса остатка потока
http2.tx_reset	Counter	Общее количество кадров сброса потока, переданных FlxGate DataPlane
http2.streams_active	Gauge	Активные потоки, наблюдаемые кодеком

Метрика	Тип	Описание
http2.pending_send_bytes	Gauge	В настоящее время буферизованные данные тела в байтах, ожидающие записи, когда окно потока/соединения открыто
http2.deferred_stream_close	Gauge	Количество HTTP/2 потоков, где поток был закрыт, но обработка закрытия потока была отложена из-за резервного копирования сети

7.1.4.6 Метрики HTTP/3 codec

Метрики HTTP/3 codec находятся в пространстве имен `http3.*` (downstream) и `cluster.<name>.http3.*` (upstream):

Метрика	Тип	Описание
http3.dropped_headers_with_underscores	Counter	Общее количество отброшенных заголовков с именами, содержащими подчеркивания
http3.requests_rejected_with_underscores_in_headers	Counter	Общее количество отклоненных запросов из-за имен заголовков, содержащих подчеркивания
http3.rx_reset	Counter	Общее количество кадров сброса потока, полученных FlxGate DataPlane
http3.tx_reset	Counter	Общее количество кадров сброса потока, переданных FlxGate DataPlane
http3.metadata_not_supported_error	Counter	Общее количество метаданных, отброшенных во время кодирования HTTP/3
http3.quic_version_h3_29	Counter	Общее количество QUIC соединений, использующих транспортную версию h3-29
http3.quic_version_rfc_v1	Counter	Общее количество QUIC соединений, использующих транспортную версию rfc-v1

7.1.5 Метрики кластеров (Clusters)

Метрики кластеров находятся в пространстве имен `cluster.<name>.*`.

7.1.5.1 Общие метрики кластера

Метрика	Тип	Описание
cluster.<name>.upstream_cx_total	Counter	Общее количество соединений
cluster.<name>.upstream_cx_active	Gauge	Общее количество активных соединений
cluster.<name>.upstream_cx_http1_total	Counter	Общее количество HTTP/1.1 соединений
cluster.<name>.upstream_cx_http2_total	Counter	Общее количество HTTP/2 соединений
cluster.<name>.upstream_cx_http3_total	Counter	Общее количество HTTP/3 соединений
cluster.<name>.upstream_cx_connect_fail	Counter	Общее количество ошибок соединения
cluster.<name>.upstream_cx_connect_timeout	Counter	Общее количество таймаутов соединения
cluster.<name>.upstream_cx_connect_with_0_rtt	Counter	Общее количество соединений, способных отправлять 0-rtt запросы (early data)
cluster.<name>.upstream_cx_idle_timeout	Counter	Общее количество таймаутов простоя соединения
cluster.<name>.upstream_cx_max_duration_reached	Counter	Общее количество соединений, закрытых из-за достижения максимальной длительности
cluster.<name>.upstream_cx_connect_attempts_exceeded	Counter	Общее количество последовательных ошибок соединения, превышающих настроенное количество попыток соединения
cluster.<name>.upstream_cx_overflow	Counter	Общее количество раз, когда circuit breaker соединений кластера переполнился
cluster.<name>.upstream_cx_connect_ms	Histogram	Время установления соединения в миллисекундах
cluster.<name>.upstream_cx_length_ms	Histogram	Длительность соединения в миллисекундах
cluster.<name>.upstream_cx_destroy	Counter	Общее количество уничтоженных соединений

Метрика	Тип	Описание
cluster.<name>.upstream_cx_destroy_local	Counter	Общее количество соединений, уничтоженных локально
cluster.<name>.upstream_cx_destroy_remote	Counter	Общее количество соединений, уничтоженных удаленно
cluster.<name>.upstream_cx_destroy_with_active_rq	Counter	Общее количество соединений, уничтоженных с 1+ активным запросом
cluster.<name>.upstream_cx_destroy_local_with_active_rq	Counter	Общее количество соединений, уничтоженных локально с 1+ активным запросом
cluster.<name>.upstream_cx_destroy_remote_with_active_rq	Counter	Общее количество соединений, уничтоженных удаленно с 1+ активным запросом
cluster.<name>.upstream_cx_close_notify	Counter	Общее количество соединений, закрытых через HTTP/1.1 заголовок connection close или HTTP/2 или HTTP/3 GOAWAY
cluster.<name>.upstream_cx_rx_bytes_total	Counter	Общее количество полученных байт соединения
cluster.<name>.upstream_cx_rx_bytes_buffered	Gauge	Полученные байты соединения, в настоящее время буферизованные
cluster.<name>.upstream_cx_tx_bytes_total	Counter	Общее количество отправленных байт соединения
cluster.<name>.upstream_cx_tx_bytes_buffered	Gauge	Отправленные байты соединения, в настоящее время буферизованные
cluster.<name>.upstream_cx_pool_overflow	Counter	Общее количество раз, когда circuit breaker пула соединений кластера переполнился
cluster.<name>.upstream_cx_protocol_error	Counter	Общее количество ошибок протокола соединения
cluster.<name>.upstream_cx_max_requests	Counter	Общее количество соединений, закрытых из-за максимального количества запросов
cluster.<name>.upstream_cx_none_healthy	Counter	Общее количество раз, когда соединение не было установлено из-за отсутствия здоровых хостов

7.1.5.2 Метрики запросов кластера

Метрика	Тип	Описание
cluster.<name>.upstream_rq_total	Counter	Общее количество запросов
cluster.<name>.upstream_rq_active	Gauge	Общее количество активных запросов
cluster.<name>.upstream_rq_pending_total	Counter	Общее количество запросов, ожидающих соединение из пула соединений
cluster.<name>.upstream_rq_pending_overflow	Counter	Общее количество запросов, которые переполнили пул соединений или circuit breaker запросов (в основном для HTTP/2 и выше) и были отклонены
cluster.<name>.upstream_rq_pending_failure_eject	Counter	Общее количество запросов, которые были отклонены из-за ошибки соединения пула соединений или удаленного завершения соединения
cluster.<name>.upstream_rq_pending_active	Gauge	Общее количество активных запросов, ожидающих соединение из пула соединений
cluster.<name>.upstream_rq_cancelled	Counter	Общее количество запросов, отмененных до получения соединения из пула соединений
cluster.<name>.upstream_rq_maintenance_mode	Counter	Общее количество запросов, которые привели к немедленному 503 из-за режима обслуживания
cluster.<name>.upstream_rq_timeout	Counter	Общее количество запросов, которые превысили таймаут ожидания ответа
cluster.<name>.upstream_rq_max_duration_reached	Counter	Общее количество запросов, закрытых из-за достижения максимальной длительности
cluster.<name>.upstream_rq_per_try_timeout	Counter	Общее количество запросов, которые достигли таймаута на попытку (кроме случаев, когда включен request hedging)
cluster.<name>.upstream_rq_rx_reset	Counter	Общее количество запросов, которые были сброшены удаленно

Метрика	Тип	Описание
cluster.<name>.upstream_rq_tx_reset	Counter	Общее количество запросов, которые были сброшены локально
cluster.<name>.upstream_rq_retry	Counter	Общее количество повторных попыток запросов
cluster.<name>.upstream_rq_retry_backoff_exponential	Counter	Общее количество повторных попыток с использованием стратегии экспоненциальной задержки
cluster.<name>.upstream_rq_retry_backoff_ratelimited	Counter	Общее количество повторных попыток с использованием стратегии задержки с ограничением скорости
cluster.<name>.upstream_rq_retry_limit_exceeded	Counter	Общее количество запросов, не повторенных из-за превышения настроенного количества максимальных повторных попыток
cluster.<name>.upstream_rq_retry_success	Counter	Общее количество успешных повторных попыток запросов
cluster.<name>.upstream_rq_retry_overflow	Counter	Общее количество запросов, не повторенных из-за circuit breaking или превышения бюджета повторных попыток
cluster.<name>.upstream_flow_control_paused_reading_total	Counter	Общее количество раз, когда управление потоком приостановило чтение от upstream
cluster.<name>.upstream_flow_control_resumed_reading_total	Counter	Общее количество раз, когда управление потоком возобновило чтение от upstream
cluster.<name>.upstream_flow_control_backed_up_total	Counter	Общее количество раз, когда upstream соединение создало резервное копирование и приостановило чтение от downstream
cluster.<name>.upstream_flow_control_drained_total	Counter	Общее количество раз, когда upstream соединение освободилось и возобновило чтение от downstream
cluster.<name>.upstream_internal_redirect_failed_total	Counter	Общее количество раз, когда неудачные внутренние перенаправления привели к передаче перенаправлений downstream
cluster.<name>.upstream_internal_redirect_succeeded_total	Counter	Общее количество раз, когда внутренние перенаправления привели ко второму upstream запросу

7.1.5.3 Динамические HTTP метрики

Если используется HTTP, доступны динамические метрики HTTP кодов ответа. Они находятся в пространстве имен `cluster.<name>.*`:

Метрика	Тип	Описание
cluster.<name>.upstream_rq_completed	Counter	Общее количество завершенных upstream запросов
cluster.<name>.upstream_rq_<*xx>	Counter	Агрегированные HTTP коды ответа (например, 2xx, 3xx и т.д.)
cluster.<name>.upstream_rq_<*>	Counter	Конкретные HTTP коды ответа (например, 201, 302 и т.д.)
cluster.<name>.upstream_rq_time	Histogram	Время запроса в миллисекундах
cluster.<name>.canary.upstream_rq_completed	Counter	Общее количество завершенных upstream canary запросов
cluster.<name>.canary.upstream_rq_<*xx>	Counter	Агрегированные HTTP коды ответа для upstream canary
cluster.<name>.canary.upstream_rq_<*>	Counter	Конкретные HTTP коды ответа для upstream canary
cluster.<name>.canary.upstream_rq_time	Histogram	Время запроса upstream canary в миллисекундах
cluster.<name>.internal.upstream_rq_completed	Counter	Общее количество завершенных внутренних origin запросов
cluster.<name>.internal.upstream_rq_<*xx>	Counter	Агрегированные HTTP коды ответа для внутреннего origin

Метрика	Тип	Описание
cluster.<name>.internal.upstream_rq_<*>	Counter	Конкретные HTTP коды ответа для внутреннего origin
cluster.<name>.internal.upstream_rq_time	Histogram	Время запроса внутреннего origin в миллисекундах
cluster.<name>.external.upstream_rq_completed	Counter	Общее количество завершённых внешних origin запросов
cluster.<name>.external.upstream_rq_<*>	Counter	Агрегированные HTTP коды ответа для внешнего origin
cluster.<name>.external.upstream_rq_<*>	Counter	Конкретные HTTP коды ответа для внешнего origin
cluster.<name>.external.upstream_rq_time	Histogram	Время запроса внешнего origin в миллисекундах

7.1.5.4 Метрики членства кластера

Метрика	Тип	Описание
cluster.<name>.membership_change	Counter	Общее количество изменений членства кластера
cluster.<name>.membership_healthy	Gauge	Текущее общее количество здоровых членов кластера (включая health checking и outlier detection)
cluster.<name>.membership_degraded	Gauge	Текущее общее количество деградированных членов кластера
cluster.<name>.membership_excluded	Gauge	Текущее общее количество исключённых членов кластера
cluster.<name>.membership_total	Gauge	Текущее общее количество членов кластера
cluster.<name>.retry_or_shadow_abandoned	Counter	Общее количество раз, когда буферизация shadowing или повторных попыток была отменена из-за лимитов буфера
cluster.<name>.config_reload	Counter	Общее количество получений API, которые привели к перезагрузке конфигурации из-за другой конфигурации
cluster.<name>.update_attempt	Counter	Общее количество попыток обновления членства кластера через service discovery
cluster.<name>.update_success	Counter	Общее количество успешных обновлений членства кластера через service discovery
cluster.<name>.update_failure	Counter	Общее количество неудачных обновлений членства кластера через service discovery
cluster.<name>.update_duration	Histogram	Количество времени, потраченного на обновление конфигураций
cluster.<name>.update_empty	Counter	Общее количество обновлений членства кластера, заканчивающихся пустым назначением нагрузки кластера и продолжающихся с предыдущей конфигурацией
cluster.<name>.update_no_rebuild	Counter	Общее количество успешных обновлений членства кластера, которые не привели к перестройке структур балансировки нагрузки кластера
cluster.<name>.version	Gauge	Хеш содержимого из последнего успешного получения API
cluster.<name>.warming_state	Gauge	Текущее состояние прогрева кластера
cluster.<name>.max_host_weight	Gauge	Максимальный вес любого хоста в кластере
cluster.<name>.bind_errors	Counter	Общее количество ошибок привязки сокета к настроенному исходному адресу
cluster.<name>.assignment_timeout_received	Counter	Общее количество полученных назначений с информацией об аренде конечной точки
cluster.<name>.assignment_stale	Counter	Количество раз, когда полученные назначения устарели до прибытия новых назначений

7.1.5.5 Метрики health check

Если настроен health check, кластер имеет дополнительные метрики в пространстве имен `cluster.<name>.health_check.*`:

Метрика	Тип	Описание
cluster.<name>.health_check.attempt	Counter	Количество health checks
cluster.<name>.health_check.success	Counter	Количество успешных health checks
cluster.<name>.health_check.failure	Counter	Количество немедленно неудачных health checks (например, HTTP 503), а также сетевых ошибок
cluster.<name>.health_check.passive_failure	Counter	Количество ошибок health check из-за пассивных событий (например, x-immediate-health-check-fail)
cluster.<name>.health_check.network_failure	Counter	Количество ошибок health check из-за сетевой ошибки
cluster.<name>.health_check.verify_cluster	Counter	Количество health checks, которые пытались проверить имя кластера
cluster.<name>.health_check.healthy	Gauge	Количество здоровых членов

7.1.5.6 Метрики обнаружения выбросов (Outlier Detection)

Если настроено обнаружение выбросов для кластера, метрики находятся в пространстве имен `cluster.<name>.outlier_detection.*`:

Метрика	Тип	Описание
cluster.<name>.outlier_detection.ejections_enforced_total	Counter	Количество принудительных исключений из-за любого типа выброса
cluster.<name>.outlier_detection.ejections_active	Gauge	Количество в настоящее время исключенных хостов
cluster.<name>.outlier_detection.ejections_overflow	Counter	Количество исключений, прерванных из-за максимального процента исключений
cluster.<name>.outlier_detection.ejections_enforced_consecutive_5xx	Counter	Количество принудительных исключений последовательных 5xx
cluster.<name>.outlier_detection.ejections_detected_consecutive_5xx	Counter	Количество обнаруженных исключений последовательных 5xx (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_success_rate	Counter	Количество принудительных исключений выбросов по success rate
cluster.<name>.outlier_detection.ejections_detected_success_rate	Counter	Количество обнаруженных исключений выбросов по success rate (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_consecutive_gateway_failure	Counter	Количество принудительных исключений последовательных ошибок шлюза
cluster.<name>.outlier_detection.ejections_detected_consecutive_gateway_failure	Counter	Количество обнаруженных исключений последовательных ошибок шлюза (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_consecutive_local_origin_failure	Counter	Количество принудительных исключений последовательных ошибок локального origin
cluster.<name>.outlier_detection.ejections_detected_consecutive_local_origin_failure	Counter	Количество обнаруженных исключений последовательных ошибок локального origin (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_local_origin_success_rate	Counter	Количество принудительных исключений выбросов по success rate для локально созданных ошибок
cluster.<name>.outlier_detection.ejections_detected_local_origin_success_rate	Counter	Количество обнаруженных исключений выбросов по success rate для локально созданных ошибок (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_failure_percentage	Counter	Количество принудительных исключений выбросов по проценту ошибок
cluster.<name>.outlier_detection.ejections_detected_failure_percentage	Counter	Количество обнаруженных исключений выбросов по проценту ошибок (даже если не принудительных)
cluster.<name>.outlier_detection.ejections_enforced_failure_percentage_local_origin	Counter	Количество принудительных исключений выбросов по проценту ошибок для локально созданных ошибок

Метрика	Тип	Описание
cluster.<name>.outlier_detection.ejections_detected_failure_percentage_local_origin	Counter	Количество обнаруженных исключений выбросов по проценту ошибок для локально созданных ошибок (даже если не принудительных)

7.1.5.7 Метрики circuit breakers

Метрики circuit breakers находятся в пространстве имен `cluster.<name>.circuit_breakers.<priority>.*`:

Метрика	Тип	Описание
cluster.<name>.circuit_breakers.<priority>.cx_open	Gauge	Открыт ли circuit breaker соединений под его лимитом конкурентности (0) или находится на пределе и больше не допускает (1)
cluster.<name>.circuit_breakers.<priority>.cx_pool_open	Gauge	Открыт ли circuit breaker пула соединений под его лимитом конкурентности (0) или находится на пределе и больше не допускает (1)
cluster.<name>.circuit_breakers.<priority>.rq_pending_open	Gauge	Открыт ли circuit breaker ожидающих запросов под его лимитом конкурентности (0) или находится на пределе и больше не допускает (1)
cluster.<name>.circuit_breakers.<priority>.rq_open	Gauge	Открыт ли circuit breaker запросов под его лимитом конкурентности (0) или находится на пределе и больше не допускает (1)
cluster.<name>.circuit_breakers.<priority>.rq_retry_open	Gauge	Открыт ли circuit breaker повторных попыток под его лимитом конкурентности (0) или находится на пределе и больше не допускает (1)
cluster.<name>.circuit_breakers.<priority>.remaining_cx	Gauge	Количество оставшихся соединений до достижения circuit breaker его лимита конкурентности
cluster.<name>.circuit_breakers.<priority>.remaining_pending	Gauge	Количество оставшихся ожидающих запросов до достижения circuit breaker его лимита конкурентности
cluster.<name>.circuit_breakers.<priority>.remaining_rq	Gauge	Количество оставшихся запросов до достижения circuit breaker его лимита конкурентности
cluster.<name>.circuit_breakers.<priority>.remaining_retries	Gauge	Количество оставшихся повторных попыток до достижения circuit breaker его лимита конкурентности

Примечание: Метрики с префиксом `remaining_` не генерируются по умолчанию. Чтобы отслеживать количество оставшихся ресурсов до открытия circuit breaker, установите параметр `track_remaining` в `true` в конфигурации circuit breaker.

7.1.5.8 Метрики бюджета таймаутов

Если включено отслеживание статистики бюджета таймаутов, метрики добавляются в `cluster.<name>.*`:

Метрика	Тип	Описание
cluster.<name>.upstream_rq_timeout_budget_percent_used	Histogram	Какой процент глобального таймаута был использован при ожидании ответа
cluster.<name>.upstream_rq_timeout_budget_percent_used_per_try_percent_used	Histogram	Какой процент таймаута на попытку был использован при ожидании ответа

7.1.5.9 Метрики балансировщика нагрузки

Метрика	Тип	Описание
cluster.<name>.lb_recalculate_zone_structures	Counter	Количество раз, когда структуры маршрутизации с учетом локалитета пересчитываются для быстрых решений по выбору upstream локалитета
cluster.<name>.lb_healthy_panic	Counter	Общее количество запросов, сбалансированных с балансировщиком нагрузки в режиме паники

Метрика	Тип	Описание
cluster.<name>.lb_zone_cluster_too_small	Counter	Нет маршрутизации с учетом зоны из-за малого размера upstream кластера
cluster.<name>.lb_zone_routing_all_directly	Counter	Отправка всех запросов напрямую в ту же зону
cluster.<name>.lb_zone_routing_sampled	Counter	Отправка некоторых запросов в ту же зону
cluster.<name>.lb_zone_routing_cross_zone	Counter	Режим маршрутизации с учетом зоны, но приходится отправлять между зонами
cluster.<name>.lb_local_cluster_not_ok	Counter	Локальный набор хостов не установлен или это режим паники для локального кластера
cluster.<name>.lb_zone_no_capacity_left	Counter	Общее количество раз, когда закончилось случайным выбором зоны из-за ошибки округления
cluster.<name>.original_dst_host_invalid	Counter	Общее количество недопустимых хостов, переданных балансировщику нагрузки original destination

7.1.5.10 Метрики подмножеств балансировщика нагрузки

Метрика	Тип	Описание
cluster.<name>.lb_subsets_active	Gauge	Количество в настоящее время доступных подмножеств
cluster.<name>.lb_subsets_created	Counter	Количество созданных подмножеств
cluster.<name>.lb_subsets_removed	Counter	Количество удаленных подмножеств из-за отсутствия хостов
cluster.<name>.lb_subsets_selected	Counter	Количество раз, когда любое подмножество было выбрано для балансировки нагрузки
cluster.<name>.lb_subsets_fallback	Counter	Количество раз, когда была вызвана политика fallback
cluster.<name>.lb_subsets_fallback_panic	Counter	Количество раз, когда сработал режим паники подмножества
cluster.<name>.lb_subsets_single_host_per_subset_duplicate	Gauge	Количество дублирующихся (неиспользуемых) хостов при использовании single_host_per_subset

7.1.5.11 Метрики ring hash балансировщика нагрузки

Метрика	Тип	Описание
cluster.<name>.ring_hash_lb.size	Gauge	Общее количество хешей хостов на кольце
cluster.<name>.ring_hash_lb.min_hashes_per_host	Gauge	Минимальное количество хешей для одного хоста
cluster.<name>.ring_hash_lb.max_hashes_per_host	Gauge	Максимальное количество хешей для одного хоста

7.1.5.12 Метрики Maglev балансировщика нагрузки

Метрика	Тип	Описание
cluster.<name>.maglev_lb.min_entries_per_host	Gauge	Минимальное количество записей для одного хоста
cluster.<name>.maglev_lb.max_entries_per_host	Gauge	Максимальное количество записей для одного хоста

7.1.5.13 Метрики размеров запросов и ответов

Если отслеживаются статистики размеров запросов и ответов, метрики добавляются в cluster.<name>*:

Метрика	Тип	Описание
cluster.<name>.upstream_rq_headers_size	Histogram	Размер заголовков запроса в байтах на upstream
cluster.<name>.upstream_rq_headers_count	Histogram	Количество заголовков запроса на upstream
cluster.<name>.upstream_rq_body_size	Histogram	Размер тела запроса в байтах на upstream
cluster.<name>.upstream_rs_headers_size	Histogram	Размер заголовков ответа в байтах на upstream
cluster.<name>.upstream_rs_headers_count	Histogram	Количество заголовков ответа на upstream
cluster.<name>.upstream_rs_body_size	Histogram	Размер тела ответа в байтах на upstream

7.1.6 Метрики TLS/SSL

Метрики TLS находятся в пространстве имен `listener.<address>.ssl.*` (для downstream) и `cluster.<name>.ssl.*` (для upstream).

Метрика	Тип	Описание
<code>*.ssl.connection_error</code>	Counter	Общее количество ошибок TLS соединения, не включая неудачные проверки сертификатов
<code>*.ssl.handshake</code>	Counter	Общее количество успешных TLS handshakes соединения
<code>*.ssl.session_reused</code>	Counter	Общее количество успешных повторных использований TLS сессии
<code>*.ssl.no_certificate</code>	Counter	Общее количество успешных TLS соединений без клиентского сертификата
<code>*.ssl.fail_verify_no_cert</code>	Counter	Общее количество TLS соединений, которые не удались из-за отсутствия клиентского сертификата
<code>*.ssl.fail_verify_error</code>	Counter	Общее количество TLS соединений, которые не удались из-за проверки CA
<code>*.ssl.fail_verify_san</code>	Counter	Общее количество TLS соединений, которые не удались из-за проверки SAN
<code>*.ssl.fail_verify_cert_hash</code>	Counter	Общее количество TLS соединений, которые не удались из-за проверки pinning сертификата
<code>*.ssl.ocsp_staple_failed</code>	Counter	Общее количество TLS соединений, которые не удались из-за несоблюдения политики OCSP
<code>*.ssl.ocsp_staple_omitted</code>	Counter	Общее количество TLS соединений, которые успешно завершились без stapling OCSP ответа
<code>*.ssl.ocsp_staple_responses</code>	Counter	Общее количество TLS соединений, где был доступен действительный OCSP ответ (независимо от того, запросил ли клиент stapling)
<code>*.ssl.ocsp_staple_requests</code>	Counter	Общее количество TLS соединений, где клиент запросил OCSP staple
<code>*.ssl.ciphers.<cipher></code>	Counter	Общее количество успешных TLS соединений, которые использовали шифр
<code>*.ssl.curves.<curve></code>	Counter	Общее количество успешных TLS соединений, которые использовали ECDHE кривую
<code>*.ssl.sigalgs.<sigalg></code>	Counter	Общее количество успешных TLS соединений, которые использовали алгоритм подписи
<code>*.ssl.versions.<version></code>	Counter	Общее количество успешных TLS соединений, которые использовали версию протокола
<code>*.ssl.was_key_usage_invalid</code>	Counter	Общее количество успешных TLS соединений, которые использовали недопустимое расширение keyUsage

7.1.6.1 Метрики сертификатов

Метрики сертификатов находятся в пространстве имен `*.ssl.certificate.<cert_name>.*`:

Метрика	Тип	Описание
<code>*.ssl.certificate.<cert_name>.expiration_unix_time_seconds</code>	Gauge	Количество секунд с эпохи UNIX даты истечения сертификата

7.1.7 Метрики TCP

Метрики TCP доступны при использовании транспортного сокета TCP stats и находятся в пространстве имен `listener.<address>.tcp_stats.*` (для downstream) и `cluster.<name>.tcp_stats.*` (для upstream).

Примечание: Эти метрики предоставляются операционной системой. Из-за различий в доступных метриках операционной системы и методологии измерений значения могут быть не

согласованы между различными операционными системами или версиями одной и той же операционной системы.

Метрика	Тип	Описание
*.tcp_stats.cx_tx_segments	Counter	Общее количество переданных TCP сегментов
*.tcp_stats.cx_rx_segments	Counter	Общее количество полученных TCP сегментов
*.tcp_stats.cx_tx_data_segments	Counter	Общее количество переданных TCP сегментов с ненулевой длиной данных
*.tcp_stats.cx_rx_data_segments	Counter	Общее количество полученных TCP сегментов с ненулевой длиной данных
*.tcp_stats.cx_tx_retransmitted_segments	Counter	Общее количество повторно переданных TCP сегментов
*.tcp_stats.cx_rx_bytes_received	Counter	Общее количество полученных байт полезной нагрузки, для которых были отправлены TCP подтверждения
*.tcp_stats.cx_tx_bytes_sent	Counter	Общее количество переданных байт полезной нагрузки (включая повторно переданные байты)
*.tcp_stats.cx_tx_unsent_bytes	Gauge	Байты, которые FlxGate DataPlane отправил в операционную систему, но которые еще не были отправлены
*.tcp_stats.cx_tx_unacked_segments	Gauge	Сегменты, которые были переданы, но еще не подтверждены
*.tcp_stats.cx_tx_percent_retransmitted_segments	Histogram	Процент сегментов на соединении, которые были повторно переданы
*.tcp_stats.cx_rtt_us	Histogram	Сглаженная оценка времени кругового пути в микросекундах
*.tcp_stats.cx_rtt_variance_us	Histogram	Оценка дисперсии в микросекундах времени кругового пути. Более высокие значения указывают на большую изменчивость

7.1.8 Метрики QUIC

Метрики QUIC доступны при использовании QUIC stats debug visitor и находятся в пространстве имен `listener.<address>.quic_stats.*`:

Метрика	Тип	Описание
listener.<address>.quic_stats.cx_tx_packets_total	Counter	Общее количество переданных пакетов
listener.<address>.quic_stats.cx_tx_packets_retransmitted_total	Counter	Общее количество повторно переданных пакетов
listener.<address>.quic_stats.cx_tx_amplification_throttling_total	Counter	Общее количество пакетов, ограниченных во время ответа handshake сервера. Часто указывает на то, что цепочка TLS сертификатов слишком длинная для передачи без дополнительного сетевого кругового пути
listener.<address>.quic_stats.cx_rx_packets_total	Counter	Общее количество полученных пакетов
listener.<address>.quic_stats.cx_path_degrading_total	Counter	Количество раз, когда было обнаружено ухудшение сетевого пути
listener.<address>.quic_stats.cx_forward_progress_after_path_degrading_total	Counter	Количество раз, когда был достигнут прогресс после ухудшения пути
listener.<address>.quic_stats.cx_rtt_us	Histogram	Сглаженная оценка времени кругового пути в микросекундах
listener.<address>.quic_stats.cx_tx_estimated_bandwidth	Histogram	Оценка пропускной способности соединения в байтах в секунду
listener.<address>.quic_stats.cx_tx_percent_retransmitted_packets	Histogram	Процент пакетов на соединении, которые были повторно переданы

Метрика	Тип	Описание
listener.<address>.quic_stats.cx_tx_mtu	Histogram	Максимальный размер пакета, который будет отправлен для соединения
listener.<address>.quic_stats.cx_rx_mtu	Histogram	Размер самого большого пакета, полученного от пира

7.1.9 Метрики менеджера слушателей

Метрики менеджера слушателей находятся в пространстве имен `listener_manager.*`:

Метрика	Тип	Описание
listener_manager.listener_added	Counter	Общее количество добавленных слушателей (либо через статическую конфигурацию, либо через LDS)
listener_manager.listener_modified	Counter	Общее количество измененных слушателей (через LDS)
listener_manager.listener_removed	Counter	Общее количество удаленных слушателей (через LDS)
listener_manager.listener_stopped	Counter	Общее количество остановленных слушателей
listener_manager.listener_create_success	Counter	Общее количество успешных добавлений объектов слушателей к рабочим потокам
listener_manager.listener_create_failure	Counter	Общее количество неудачных добавлений объектов слушателей к рабочим потокам
listener_manager.listener_in_place_updated	Counter	Общее количество объектов слушателей, созданных для выполнения пути обновления цепочки фильтров
listener_manager.total_filter_chains_draining	Gauge	Количество в настоящее время draining цепочек фильтров
listener_manager.total_listeners_warming	Gauge	Количество в настоящее время прогревающихся слушателей
listener_manager.total_listeners_active	Gauge	Количество в настоящее время активных слушателей
listener_manager.total_listeners_draining	Gauge	Количество в настоящее время draining слушателей
listener_manager.workers_started	Gauge	Булево значение (1, если запущено, и 0 в противном случае), которое указывает, были ли слушатели инициализированы на рабочих потоках

7.1.10 Метрики менеджера кластеров

Метрики менеджера кластеров находятся в пространстве имен `cluster_manager.*`:

Метрика	Тип	Описание
cluster_manager.cluster_added	Counter	Общее количество добавленных кластеров (либо через статическую конфигурацию, либо через CDS)
cluster_manager.cluster_modified	Counter	Общее количество измененных кластеров (через CDS)
cluster_manager.cluster_removed	Counter	Общее количество удаленных кластеров (через CDS)
cluster_manager.cluster_updated	Counter	Общее количество обновлений кластеров
cluster_manager.cluster_updated_via_merge	Counter	Общее количество обновлений кластеров, примененных как объединенные обновления
cluster_manager.update_merge_cancelled	Counter	Общее количество объединенных обновлений, которые были отменены и доставлены досрочно
cluster_manager.update_out_of_merge_window	Counter	Общее количество обновлений, которые прибыли вне окна объединения
cluster_manager.active_clusters	Gauge	Количество в настоящее время активных (прогретых) кластеров
cluster_manager.warming_clusters	Gauge	Количество в настоящее время прогревающихся (не активных) кластеров

7.1.10.1 Метрики по рабочим потокам

Метрики по рабочим потокам находятся в пространстве имен `thread_local_cluster_manager.<worker_id>.*`:

Метрика	Тип	Описание
<code>thread_local_cluster_manager.<worker_id>.clusters_inflated</code>	Gauge	Количество кластеров, которые рабочий поток инициализировал. При использовании отложенного создания кластеров это число должно быть \leq (<code>cluster_added</code> - <code>clusters_removed</code>)

7.1.11 Использование метрик

7.1.11.1 Доступ к метрикам через Prometheus

Метрики доступны через административный интерфейс FlxGate DataPlane:

```
# Получить все метрики в формате Prometheus
curl http://localhost:9901/stats/prometheus

# Получить только используемые метрики
curl http://localhost:9901/stats/prometheus?usedonly

# Получить метрики с text_readouts
curl http://localhost:9901/stats/prometheus?text_readouts

# Получить гистограммы в формате summary
curl http://localhost:9901/stats/prometheus?histogram_buckets=summary
```

7.1.11.2 Конфигурация Prometheus

Пример конфигурации Prometheus для сбора метрик FlxGate DataPlane:

```
scrape_configs:
- job_name: 'flxgate-dp'
  metrics_path: /stats/prometheus
  static_configs:
  - targets: ['localhost:9901']
```

7.1.11.3 Интеграция с Zabbix

7.1.11.3.1 Использование Zabbix для мониторинга Prometheus

Если у вас уже настроен Prometheus, можно использовать Zabbix для мониторинга самого Prometheus:

1. **Настройте сбор метрик из Prometheus** через HTTP Agent в Zabbix
2. **Используйте PromQL запросы** через API Prometheus
3. **Создайте шаблон** для автоматического обнаружения метрик

7.1.11.3.2 Примеры ключей для Zabbix

После настройки можно использовать следующие ключи:

```
# Получить конкретную метрику
flxgate.metrics[flxgate_server_uptime]

# Получить все метрики
```

```
flxgate.metrics.all
```

```
# Примеры важных метрик для мониторинга
flxgate.metrics[flxgate_server_live]
flxgate.metrics[flxgate_listener_downstream_cx_active]
flxgate.metrics[flxgate_http_downstream_rq_total]
flxgate.metrics[flxgate_http_downstream_rq_5xx]
```

7.1.11.3.3 Создание шаблона Zabbix

Рекомендуется создать шаблон Zabbix с предустановленными элементами данных для ключевых метрик:

1. **Важные метрики для мониторинга:**
2. `flxgate_server_uptime` - время работы сервера
3. `flxgate_server_live` - статус сервера (1 = активен, 0 = draining)
4. `flxgate_listener_downstream_cx_active` - активные соединения
5. `flxgate_http_downstream_rq_total` - общее количество запросов
6. `flxgate_http_downstream_rq_5xx` - количество ошибок 5xx
7. `flxgate_cluster_upstream_cx_active` - активные upstream соединения
8. **Триггеры для оповещений:**
9. Сервер недоступен (`server.live = 0`)
10. Высокий процент ошибок 5xx
11. Превышение лимита соединений
12. Отсутствие активных upstream соединений

7.1.11.3.4 Настройка через Zabbix API

Пример создания элемента данных через Zabbix API:

```
curl -X POST -H 'Content-Type: application/json' \
-d '{
  "jsonrpc": "2.0",
  "method": "item.create",
  "params": {
    "name": "FlxGate Server Uptime",
    "key": "flxgate.metrics[flxgate_server_uptime]",
    "hostid": "10084",
    "type": 0,
    "value_type": 3,
    "units": "s",
    "delay": "30s"
  },
  "auth": "YOUR_AUTH_TOKEN",
  "id": 1
}' \
http://zabbix-server/api_jsonrpc.php
```

7.1.11.4 Интеграция с Grafana

Ниже описаны шаги по настройке Grafana для мониторинга метрик FlxGate DataPlane.

7.1.11.4.1 Добавление источника данных Prometheus

1. Войдите в **Grafana** и перейдите в **Configuration** → **Data Sources**
2. Нажмите **"Add data source"** и выберите **Prometheus**
3. Настройте подключение:
4. **URL:** `http://prometheus-server:9090` (адрес вашего Prometheus сервера)
5. **Access:** Server (default) или Browser (если Grafana может напрямую обращаться к Prometheus)
6. **HTTP Method:** GET
7. **Scrape interval:** 15s (рекомендуется)
8. Нажмите **"Save & Test"** для проверки подключения

7.1.11.4.2 Создание дашборда

Пример 1: Обзор сервера

Создайте панель для мониторинга основных метрик сервера:

Панель "Uptime": - **Query:** `flxgate_server_uptime` - **Visualization:** Stat - **Unit:** seconds (s)

Панель "Server Status": - **Query:** `flxgate_server_live` - **Visualization:** Stat - **Thresholds:** - Green: 1 - Red: 0

Панель "Memory Usage": - **Query:** `flxgate_server_memory_allocated` - **Visualization:** Graph - **Unit:** bytes (B)

Пример 2: Мониторинг HTTP запросов

Панель "Request

Rate": - **Query:** `rate(flxgate_http_downstream_rq_total[5m])` - **Visualization:** Graph - **Unit:** reqps (requests/sec)

Панель "HTTP Status Codes": - Query:

```
sum(rate(flxgate_http_downstream_rq_2xx[5m])) by (instance)
sum(rate(flxgate_http_downstream_rq_4xx[5m])) by (instance)
sum(rate(flxgate_http_downstream_rq_5xx[5m])) by (instance)
- Visualization: Time series - Legend: {{status_code}}
```

Панель "Error Rate": - Query:

```
sum(rate(flxgate_http_downstream_rq_5xx[5m])) /
sum(rate(flxgate_http_downstream_rq_total[5m])) * 100
- Visualization: Graph - Unit: percent (%)
```

Панель "Response Time (p99)": - Query:

```
histogram_quantile(0.99, rate(flxgate_http_downstream_rq_time_bucket[5m]))
- Visualization: Graph - Unit: milliseconds (ms)
```


Пример 3: Мониторинг соединений

Панель "Active

Connections": - Query: `flxgate_listener_downstream_cx_active` - **Visualization:** Graph
- **Unit:** short

Панель "Connection

Rate": - Query: `rate(flxgate_listener_downstream_cx_total[5m])` - **Visualization:** Graph
- **Unit:** conmps (connections/sec)

Панель "Rejected

Connections": - Query: `rate(flxgate_listener_downstream_cx_overflow[5m])` - **Visualization:** Graph
- **Unit:** conmps

Пример 4: Мониторинг кластеров

Панель "Upstream Active

Connections": - Query: `flxgate_cluster_upstream_cx_active` - **Visualization:** Graph - **Unit:** short

Панель "Upstream Request

Rate": - Query: `rate(flxgate_cluster_upstream_rq_total[5m])` - **Visualization:** Graph - **Unit:** reqps

Панель "Upstream Error Rate": - Query:

`sum(rate(flxgate_cluster_upstream_rq_5xx[5m])) / sum(rate(flxgate_cluster_upstream_rq_total[5m])) * 100`
- **Visualization:** Graph - **Unit:** percent (%)

7.1.11.4.3 Создание алертов

Grafana позволяет создавать алерты на основе метрик:

1. **Перейдите в панель** и нажмите на иконку редактирования
2. **Откройте вкладку "Alert"**
3. **Настройте правило алерта:**

Пример алерта: Высокий процент ошибок - Condition: `WHEN avg() OF query(A, 5m, now) IS ABOVE 5` - **Query A:** `sum(rate(flxgate_http_downstream_rq_5xx[5m])) / sum(rate(flxgate_http_downstream_rq_total[5m])) * 100` - **Message:** Error rate is above 5%

Пример алерта: Сервер недоступен - Condition: `WHEN last() OF query(A, 1m, now) IS BELOW 1` - **Query A:** `flxgate_server_live` - **Message:** FlxGate server is in draining state

7.1.11.4.4 Полезные переменные для дашбордов

Создайте переменные для удобной фильтрации:

1. **Перейдите в Dashboard Settings → Variables**
2. **Добавьте переменные:**

Пример переменной "instance": - Name: `instance` - **Type:** Query
- **Query:** `label_values(flxgate_server_uptime, instance)` - **Multi-value:** Enable

Пример переменной "listener": - Name: listener - Type: Query

- Query: label_values(flxgate_listener_downstream_cx_active, listener) - Multi-value: Enable

Затем используйте переменные в запросах:

```
flxgate_listener_downstream_cx_active{listener="$listener"}
```

7.1.11.5 Примеры PromQL запросов

Rate ошибок соединений

```
rate(flxgate_listener_downstream_cx_overflow[5m])
```

Успешность запросов

```
rate(flxgate_http_downstream_rq_2xx[5m]) / rate(flxgate_http_downstream_rq_total[5m])
```

Время ответа (p99)

```
histogram_quantile(0.99, rate(flxgate_http_downstream_rq_time_bucket[5m]))
```

Активные соединения

```
flxgate_listener_downstream_cx_active
```

Процент ошибок upstream

```
rate(flxgate_cluster_upstream_rq_5xx[5m]) / rate(flxgate_cluster_upstream_rq_total[5m]) * 100
```

7.1.12 Примечания

1. **Типы метрик:**
2. **Counter:** Счетчик, который только увеличивается
3. **Gauge:** Значение, которое может увеличиваться и уменьшаться
4. **Histogram:** Распределение значений с процентиями
5. **Пространства имен:** Метрики организованы по пространствам имен, где <stat_prefix>, <address>, <name> и другие заменяются на реальные значения из конфигурации.
6. **Динамические метрики:** Некоторые метрики (например, HTTP коды ответа) создаются динамически на основе фактических значений в запросах/ответах.
7. **Теги:** В Prometheus метрики могут иметь теги (labels), которые добавляются автоматически на основе конфигурации Flxgate Data Plane.

7.1.13 Дополнительные ресурсы

- Prometheus документация <https://prometheus.io/docs/>
- PromQL запросы <https://prometheus.io/docs/prometheus/latest/querying/basics/>
- Grafana документация <https://grafana.com/docs/>

- Zabbix документация <https://www.zabbix.com/documentation>

8 Термины, сокращения и определения

Термин	Определение
API	Application Programming Interface. Программный интерфейс
CA	Certification Authority. Центр сертификации, удостоверяющий центр
CLI	Command Line Interface. Интерфейс командной строки
CRL	Certificate Revocation List. Список отзыва сертификатов
DHCP	Dynamic Host Configuration Protocol. Протокол динамического конфигурирования узлов
DNS	Domain Name System. Система доменных имен
flxGATE	Платформа управления сетевым трафиком
HA	High Availability. Высокая доступность
HTTP	HyperText Transfer Protocol. Протокол передачи гипертекста
ICMP	Internet Control Message Protocol. Протокол межсетевых управляющих сообщений. Передает сообщения об ошибках и других исключительных ситуациях, которые возникают при передаче данных
ID	Идентификатор
IP	Интернет-протокол
IPMI	Intelligent Platform Management Interface. Интеллектуальный интерфейс управления платформой
ISO-образ	Архивный файл, который содержит идентичную копию (образ) данных
JSON	JavaScript Object Notation. Текстовый формат обмена данными, основанный на JavaScript
LDAP	Lightweight Directory Access Protocol. Протокол быстрого доступа к каталогам
LLDP	Link Layer Discovery Protocol. Протокол канального уровня, который позволяет сетевым устройствам анонсировать в сеть информацию о себе и о своих возможностях, а также собирать информацию о соседних устройствах
MAC-адрес	Media Access Control Address. Уникальный идентификатор. Присваивается каждому сетевому оборудованию
MTU	Maximum Transmission Unit. Максимальная единица передачи
NFS	Network File System. Сетевая файловая система
NIC	Network Interface Card. Сетевая интерфейсная карта
OpenLDAP	OpenLDAP. Протокол облегченного доступа к каталогам с открытым исходным кодом

Термин	Определение
QoS	Quality of Service. Набор технологических решений для оптимизации сетевого трафика с помощью назначаемых приоритетов передачи информации
RBAC	Role-Based Access Control. Управление доступом на основе ролей
REST API	Способ доступа к веб-сервисам без какой-либо обработки
SCTP	Stream Control Transmission Protocol. Протокол транспортного уровня в компьютерных сетях
SDN	Software-Defined Networking. Программно-определяемые сети
SFTP	Secure File Transfer Protocol. Протокол безопасной передачи файлов через сеть
SR-IOV	Single Root Input/Output Virtualization. Виртуализация ввода-вывода. Применяется для виртуализации ресурсов ввода-вывода для отдельных серверов
SSH	Secure Shell. Безопасная оболочка – сетевой протокол прикладного уровня. Позволяет удаленно управлять операционной системой и туннелировать TCP-соединения
SSL	Secure Sockets Layer. Протокол безопасности, который создает зашифрованное соединение между веб-сервером и веб-браузером
TCP	Transmission Control Protocol. Протокол управления передачей данных
TLS	Transport Layer Security. Криптографический протокол обеспечения безопасной передачи данных
UDP	User Datagram Protocol. Сетевой протокол транспортного уровня. Использует IP для передачи данных от одного устройства к другому. Данные (датаграммы), которые вносятся в пакет UDP, включают порты назначения, источник, контрольную сумму и длину пакета
URI	Uniform Resource Identifier. Унифицированный идентификатор ресурса
UUID	Универсально уникальный идентификатор. 128-битная метка, используемая для идентификации информации
vCPU	Virtual Central Processing Unit. Виртуализированный вариант физического CPU – центральные блоки управления в виртуальных машинах и облачных средах
VF	Virtual Function. Виртуальная функция
VIP	Virtual IP address. Виртуальный IP-адрес – компонент сетевой и интернет-инфраструктуры, который обеспечивает балансировку нагрузки, высокую доступность и эффективное распределение ресурсов в вычислительной среде. Это уникальная числовая метка, присвоенная виртуальной машине или службе, а не физическому устройству
VLAN	Virtual Local Area Network. Виртуальная локальная сеть
VNC	Virtual Network Computing. Метод удаленного доступа к рабочему столу компьютера по сети

Термин	Определение
vNIC	Virtual Network Interface Card. Виртуальный сетевой интерфейс, основанный на физических сетевых картах узла
VoIP	Voice over Internet Protocol. Технология передачи голосовых сообщений в локальных сетях или в сети Интернет с использованием протокола IP
VRRP	Virtual Router Redundancy Protocol. Сетевой протокол, предназначенный для увеличения доступности маршрутизаторов, которые выполняют роль шлюза по умолчанию
VXLAN	Virtual Extensible LAN. Технология сетевой виртуализации для решения проблем масштабируемости в больших системах облачных вычислений
YAML	Yet Another Markup Language. Формат сериализации данных. Используется при управлении конфигурацией, а также для хранения данных в структурированном формате
ОЗУ	Оперативная память
ОС	Операционная система
ПО	Программное обеспечение
ЦПУ	Центральный процессор