Программное обеспечение

«Servicepipe Application Delivery Controller»

Руководство пользователя

Москва - 2026 г.

Оглавление

1 Общие сведения

1.1 Назначение документа

Настоящий документ является руководством пользователя программного обеспечения «Servicepipe Application Delivery Controller» (далее - ПО «Servicepipe ADC», ADC, программное обеспечение).

Документ описывает назначение программного обеспечения, общую модель работы с трафиком, порядок использования разделов панели управления Application delivery, правила создания и изменения объектов конфигурации, а также типовые сценарии эксплуатации и диагностики.

Руководство предназначено для применения после установки и первичной настройки программного обеспечения. Порядок установки, обновления и удаления компонентов Servicepipe ADC описывается в отдельной инструкции по установке программного обеспечения.

1.2 Сведения о программном обеспечении

ПО «Servicepipe ADC» предназначено для публикации прикладных сервисов и API, маршрутизации и балансировки L4/L7-трафика, включая TCP/UDP и HTTP/HTTPS, терминации и проксирования TLS-соединений, централизованного управления конфигурацией, а также применения правил обработки трафика.

Servicepipe ADC размещается перед backend-сервисами заказчика. Программное обеспечение принимает входящий трафик, сопоставляет его с правилами обработки, применяет политики маршрутизации, TLS, балансировки, доступности, логирования и обработки HTTP-трафика, после чего передает запрос или соединение в соответствующий backend-пул.

Управление основными функциями ADC выполняется через административный контур и раздел Application delivery панели управления. В зависимости от прав пользователя и состава поставки в интерфейсе доступны функции статистики, управления трафиком, идентификации клиентов API, управления TLS, секретами, плагинами и вспомогательными объектами конфигурации.

1.3 Категории пользователей

Руководство рассчитано на сотрудников заказчика и организации, выполняющей эксплуатацию ADC, которым предоставлен доступ к административному контуру программного обеспечения.

Категория пользователя	Основные задачи
Администратор ADC	Создание и изменение сервисов, маршрутов, upstream-пулов, сертификатов, секретов, глобальных правил и конфигураций плагинов. Контроль корректности конфигурации.
Инженер эксплуатации / SRE	Мониторинг состояния трафика, анализ ошибок и задержек, настройка проверок здоровья, диагностика проблем с backend-узлами и сетевыми соединениями.
DevOps-инженер	Публикация приложений и API, настройка маршрутизации, подключение плагинов, подготовка типовых конфигураций для команд разработки.

Категория пользователя	Основные задачи
Владелец сервиса / приложения	Просмотр параметров публикации сервиса, проверка статистики, согласование правил маршрутизации, лимитов и TLS-настроек.

Примечание. Набор доступных действий зависит от роли пользователя, прав доступа и политики администрирования, принятой в конкретной инсталляции Servicepipe ADC.

1.4 Термины и сокращения

Термин	Описание
ADC	Application Delivery Controller - программный контроллер доставки приложений, принимающий L4/L7-трафик и применяющий политики маршрутизации, TLS, балансировки, доступности и обработки запросов.
Application delivery	Раздел панели управления Servicepipe ADC, предназначенный для работы со статистикой, трафиком, идентификацией, безопасностью и конфигурацией gateway-уровня.
Route / Маршрут	L7-правило сопоставления HTTP-запроса по URI, Host, HTTP-методу, IP-адресу и дополнительным условиям.
Stream Route / Потоковый маршрут	L4-правило сопоставления TCP, TLS, UDP или другого stream-трафика по адресу сервера, порту, remote address, SNI и связанным настройкам.
Service / Сервис	Логическая сущность, объединяющая общие настройки обработки трафика: hosts, WebSocket, script, Upstream, health checks и плагины.
Upstream	Пул backend-узлов и параметры соединения с ними: схема, алгоритм балансировки, retries, таймауты, keepalive, TLS и health checks.
Node / Нода	Конкретный backend-адрес внутри Upstream: host, port, weight и priority.
Consumer / Потребитель	Идентификатор клиента API или приложения, используемый для применения плагинов аутентификации, авторизации, лимитов и логирования.
Consumer Group / Группа потребителей	Группа потребителей с общими настройками плагинов и политик обработки трафика.
Plugin / Плагин	Модуль расширения обработки трафика: аутентификация, логирование, rate limiting, rewrite, observability, интеграции и другие функции.
Plugin Config / Конфигурация плагинов	Переиспользуемый набор плагинов, который можно подключать к маршрутам и потоковым маршрутам по идентификатору.
Global Rule / Глобальное правило	Конфигурация плагинов, применяемая ко всему трафику ADC независимо от конкретного маршрута или сервиса.
SSL-сертификат	Серверный или клиентский сертификат с приватным ключом и параметрами TLS/SNI.
Secret / Секрет	Ссылка на внешний менеджер секретов или конфигурацию доступа к нему, используемую для хранения чувствительных значений.
Proto	Содержимое .proto-файла, используемое в сценариях обработки gRPC/protobuf-трафика.
SNI	Server Name Indication - имя домена, передаваемое клиентом при установлении TLS-соединения.

2 Общая модель работы с ADC

2.1 Назначение и область применения

Servicepipe ADC используется как единая точка публикации HTTP(S), gRPC, TCP, TLS и UDP-сервисов в инфраструктуре заказчика или private cloud. Программное обеспечение позволяет отделить внешний трафик от backend-инфраструктуры, централизовать TLS-терминацию, управлять маршрутизацией запросов, выполнять проверки здоровья backend-узлов и подключать плагины обработки трафика без изменения кода приложений.

В прикладных сценариях ADC применяется для публикации нескольких приложений через единый входной контур, балансировки нагрузки между backend-нодами, ограничения нагрузки на backend, ведения журналов, диагностики трафика, управления идентификацией клиентов API и подключения дополнительных политик обработки запросов.

2.2 Модель обработки HTTP/L7-трафика

При обработке HTTP(S) или gRPC-трафика ADC анализирует признаки прикладного запроса и выбирает подходящую конфигурацию маршрута.

1. Клиент отправляет HTTP(S) или gRPC-запрос на адрес ADC.
2. ADC ищет подходящий маршрут по URI, Host, HTTP-методу, IP-адресу клиента и дополнительным условиям.
3. Если маршрут привязан к сервису, используется конфигурация соответствующего сервиса. Если сервис не задан, используется Upstream маршрута.
4. Применяются глобальные правила, плагины сервиса, маршрута, конфигурации плагинов, потребителя и группы потребителей с учетом правил приоритета.
5. ADC выбирает backend-ноду в Upstream по заданному алгоритму балансировки и отправляет запрос.
6. Ответ backend-а возвращается клиенту; пассивные health checks могут обновить состояние backend-ноды на основании результата реального запроса.

2.3 Модель обработки L4/stream-трафика

Для TCP, TLS, UDP и других stream-сценариев ADC работает с соединением без обязательного разбора HTTP-запроса. В этом режиме используется сущность Stream Route.

1. Клиент устанавливает соединение с адресом и портом, опубликованными потоковым маршрутом.
2. ADC сопоставляет соединение по server address, server port, remote address и, при наличии TLS, по SNI.
3. Если для stream-подсистемы настроены плагины, ADC применяет их к соединению.
4. Соединение передается в Service или Upstream в соответствии с настройками потокового маршрута.

2.4 Основные сущности управления

Конфигурация ADC строится из взаимосвязанных сущностей. Пользователь может создавать самостоятельные объекты и переиспользовать их в маршрутах, сервисах и потоковых маршрутах.

Сущность	Назначение	Где используется
Upstream	Определяет backend-пул и параметры соединения с ним.	В Service, Route и Stream Route.
Service	Содержит общие настройки обработки трафика и может	В HTTP/L7-маршрутах и stream-сценариях при необходимости.

Сущность	Назначение	Где используется
	переиспользоваться несколькими маршрутами.	
Route	Публикует HTTP/L7-трафик и направляет запросы в Service или Upstream.	Для HTTP(S), gRPC и других L7-сценариев.
Stream Route	Публикует L4/stream-трафик.	Для TCP, TLS, UDP и stream-протоколов.
Consumer	Идентифицирует клиента API.	В плагинах аутентификации, лимитов и политик доступа.
SSL	Хранит TLS-сертификаты, приватные ключи, SNI и TLS-протоколы.	Для входящего TLS, mTLS и TLS до upstream.
Secret	Хранит параметры подключения к внешним менеджерам секретов.	В конфигурациях, где нужны чувствительные значения.
Plugin Config	Позволяет переиспользовать набор плагинов.	В Route и Stream Route.
Global Rule	Применяет плагины ко всему трафику.	На gateway-уровне независимо от конкретного маршрута.
Proto	Хранит .proto-описания.	В gRPC/protobuf и отдельных протокольных сценариях.

3 Работа с панелью управления

3.1 Переход в раздел Application delivery

Для работы с функциями ADC пользователь открывает панель управления и переходит в раздел Application delivery. Раздел содержит группы меню для просмотра статистики, управления трафиком, идентификацией, безопасностью и конфигурацией gateway-уровня.

После перехода в раздел пользователь выбирает нужную группу меню и подраздел. Доступность подразделов и операций зависит от роли пользователя и настроек доступа.

3.2 Навигация раздела

Группа меню	Раздел	Назначение
Статистика	Обзор	Просмотр ключевых метрик, графиков запросов, ошибок, задержек, пропускной способности и активных соединений.
Трафик	Сервисы	Управление логическими сервисами и их общей конфигурацией.
Трафик	Маршруты	Управление HTTP/L7-правилами сопоставления запросов.
Трафик	Потоковые маршруты	Управление L4/stream-маршрутами для TCP, TLS, UDP и связанных протоколов.
Трафик	Upstreams	Управление backend-пулами, алгоритмами балансировки, таймаутами, TLS и health checks.
Идентификация	Потребители	Управление идентификаторами клиентов API.
Идентификация	Группы потребителей	Группировка потребителей и применение общих плагинов.
Безопасность	SSL-сертификаты	Управление сертификатами, приватными ключами, SNI и TLS-протоколами.
Безопасность	Секреты	Подключение внешних менеджеров секретов.
Конфигурация	Глобальные правила	Управление плагинами, выполняемыми глобально для всего трафика.

Группа меню	Раздел	Назначение
Конфигурация	Конфигурации плагинов	Управление переиспользуемыми наборами плагинов.
Конфигурация	Метаданные плагинов	Управление глобальными параметрами отдельных плагинов.
Конфигурация	Protos	Хранение .proto-описаний для gRPC/protobuf-сценариев.

3.3 Работа со списками объектов

Большинство подразделов Application delivery содержит табличный список объектов. Таблица используется для просмотра текущей конфигурации, перехода к созданию нового объекта, редактирования существующих объектов и удаления неиспользуемых сущностей.

- Кнопка «+ Добавить» открывает форму создания нового объекта.
- Колонка «Действия» содержит операции редактирования и удаления.
- В нижней части таблицы может отображаться пагинация и выбор размера страницы.
- При удалении объекта интерфейс выводит подтверждение, так как действие может привести к изменению обработки трафика.
- После успешного создания, изменения или удаления объекта интерфейс отображает уведомление о результате операции.

3.4 Работа с формами создания и редактирования

Формы создания и редактирования открываются в боковой панели. Внутри формы поля сгруппированы по блокам, а в левой части формы может отображаться навигация по разделам формы.

В нижней части формы расположены кнопки сохранения и отмены изменений. В режиме редактирования часть идентификаторов может быть заблокирована, так как изменение таких значений может нарушить связи между объектами конфигурации.

> **Примечание.** Перед сохранением изменений необходимо проверить, что объект имеет корректный статус, обязательные поля заполнены, JSON-поля валидны, а ссылки на другие сущности указывают на существующие объекты.

3.5 Заполнение типовых полей

Тип поля	Правило заполнения	Назначение
Метки	Вводятся в формате key:value, после ввода значения нажимается Enter.	Используются для группировки, поиска и сопровождения конфигураций.
Списки значений	Каждое значение вводится отдельно и подтверждается Enter.	Используются для Host, URI, удаленных адресов, HTTP-статусов и других множественных параметров.
JSON-поля	Заполняются корректным JSON-объектом или JSON-массивом.	Используются для Vars, конфигураций плагинов, протокольных параметров и метаданных.
Переключатели	Включают или отключают дополнительные блоки формы.	Например, Health Checks, существующий Upstream, keepalive pool, WebSocket.
Статус	Обычно принимает значения «Включён» или «Отключён».	Включенный объект участвует в обработке трафика; отключенный сохраняется в конфигурации, но не применяется.

Тип поля	Правило заполнения	Назначение
Ссылки на объекты	Заполняются выбором существующей сущности или вводом ее идентификатора.	Используются для связи маршрутов с сервисами, upstream-пулами и конфигурациями плагинов.

3.6 Работа с плагинами

Плагины расширяют обработку трафика и могут подключаться на разных уровнях: глобальные правила, сервисы, маршруты, потоковые маршруты, потребители, группы потребителей и конфигурации плагинов.

Общий порядок добавления плагина:

1. Откройте форму объекта, к которому должен быть подключен плагин.
2. В блоке «Плагины» нажмите «+ Добавить».
3. В окне выбора найдите нужный плагин через поиск.
4. Выберите плагин и откройте его настройки.
5. Заполните JSON-конфигурацию плагина.
6. Сохраните форму объекта.

Примечание. Один и тот же плагин может быть задан на нескольких уровнях. Перед добавлением плагина проверьте, не настроен ли он уже в глобальном правиле, сервисе, маршруте, Plugin Config, Consumer или Consumer Group. Ошибочная конфигурация на глобальном уровне может повлиять на весь трафик.

4 Статистика и мониторинг

4.1 Назначение раздела статистики

Раздел «Статистика -> Обзор» предназначен для просмотра сводных показателей и графиков по трафику Application delivery. С его помощью пользователь контролирует интенсивность запросов, долю ошибок, задержки, пропускную способность, число активных соединений и состояние upstream-пулов.

Статистика используется для оперативной диагностики, оценки влияния изменений конфигурации, выявления проблем backend-а и анализа нагрузки на опубликованные приложения.

4.2 Фильтры и показатели

Элемент	Описание
Период	Выбор временного диапазона, за который отображаются агрегированные данные.
Обновление	Обновление данных выбранного периода.
Все сервисы	Фильтр по сервисам, позволяющий смотреть статистику выбранного сервиса.
Все роуты	Фильтр по маршрутам, позволяющий смотреть статистику выбранного маршрута.
Requests / sec	Интенсивность запросов.
Error rate	Доля ошибочных ответов, используемая для контроля 4xx/5xx.
P95 latency	95-й процентиль задержки, отражающий время обслуживания большинства запросов.
Bandwidth	Пропускная способность.
Active connections	Количество активных соединений.
Upstreams	Состояние upstream-пулов и признаки проблем backend-а.

4.3 Графики и интерпретация данных

График	Назначение
Request Rate	Показывает динамику запросов по времени. Серии могут разделяться по классам HTTP-статусов 2xx, 3xx, 4xx и 5xx.
Error Rate	Показывает долю ошибок по времени и помогает разделять клиентские и серверные ошибки.
Latency Percentiles	Показывает процентили задержек p50, p90, p95, p99. Может использоваться для анализа задержек request, upstream и ADC.
Latency Breakdown	Показывает разложение задержки по компонентам: requests, upstream, adc.

При анализе проблем рекомендуется сопоставлять графики запросов, ошибок и задержек. Например, одновременный рост Error Rate и Latency Percentiles может указывать на деградацию backend-а, сетевые проблемы или некорректные таймауты.

5 Управление трафиком

5.1 Upstream и backend-ноды

Upstream определяет backend-пул, в который ADC направляет трафик после выбора маршрута или сервиса. Upstream может быть самостоятельной сущностью или inline-настройкой внутри сервиса, маршрута либо потокового маршрута.

Самостоятельный Upstream рекомендуется создавать, если один backend-пул используется более чем одним маршрутом или сервисом. Это снижает дублирование и упрощает сопровождение конфигурации.

Блок / поле	Назначение
Основная информация	Имя, описание и метки upstream-пула.
Источник Upstream	Выбор статических нод или Service Discovery.
Ноды	Список backend-адресов с host, port, weight и priority.
Схема	Протокол соединения от ADC к backend: HTTP, HTTPS, gRPC, gRPCs, TCP, TLS, UDP.
Тип балансировки	Алгоритм распределения трафика: Round Robin, CHash, Least Conn, EWMA и другие поддерживаемые варианты.
Hash On и ключ	Параметры consistent hashing для sticky-сценариев.
Pass Host	Определяет, какой Host будет отправлен backend-у: исходный Host, host ноды или значение rewrite.
Повторные попытки	Включают retries и задают число повторов и общий таймаут повторов.
Таймауты	Connect, Send и Read для соединения с backend-ом.
Keepalive Pool	Параметры пула постоянных соединений к backend-узлам.
TLS	Настройки TLS и клиентского сертификата для HTTPS, gRPCs и TLS-сценариев.

Примечание. Не задавайте чрезмерно большое число повторных попыток. При аварии backend-а это может усилить нагрузку и увеличить время ответа клиенту.

5.2 Проверки здоровья

Health checks используются для временного исключения неработающих backend-нод из балансировки и возврата их в работу после восстановления. Проверки могут быть активными и пассивными.

Тип проверки	Описание	Когда использовать
Активная проверка	ADC самостоятельно выполняет тестовые HTTP(S) или TCP-проверки backend-нод.	Для production-сервисов, где нужен контролируемый и регулярный мониторинг доступности backend-а.
Пассивная проверка	ADC анализирует реальные ответы backend-а на пользовательский трафик.	Как дополнительный механизм оценки состояния backend-а по фактическому трафику.

Для активных проверок задаются тип проверки, таймаут, параллельность, Host, порт, HTTP-путь, проверка HTTPS-сертификата, заголовки запроса, интервалы и критерии Healthy/Unhealthy. Для пассивных проверок задаются критерии оценки реальных ответов backend-а.

- Для production-сервисов рекомендуется использовать отдельный lightweight endpoint, например /health или /ready.
- Пассивные проверки полезны как дополнение, но не всегда могут самостоятельно вернуть ноду в healthy-состояние без активных проверок.
- Интервалы проверок не должны быть чрезмерно короткими, так как это увеличивает нагрузку на backend.
- HTTP-статусы healthy и unhealthy должны соответствовать фактической логике приложения.

5.3 Сервисы

Сервис является логической сущностью, которая описывает общий backend-сервис и может переиспользоваться несколькими маршрутами. Сервис позволяет вынести общие настройки Upstream, Host, WebSocket, script и плагины из отдельных маршрутов.

Операция	Порядок выполнения
Создание сервиса	Перейдите в «Трафик -> Сервисы», нажмите «+ Добавить», заполните основную информацию и настройки сервиса, затем сохраните форму.
Подключение Upstream	Выберите существующий Upstream либо задайте inline-настройки backend-пула внутри формы сервиса.
Настройка WebSocket	Включите WebSocket, если сервис использует upgrade-соединения.
Подключение плагинов	Добавьте плагины уровня сервиса, если они должны применяться ко всем маршрутам, использующим этот сервис.
Редактирование сервиса	Откройте действие редактирования в таблице сервисов, измените параметры и сохраните форму.

Используйте сервисы, если несколько маршрутов должны вести в один backend, если нужно централизованно менять Upstream для набора маршрутов или если backend имеет общие плагины и Host/WebSocket-настройки.

5.4 HTTP/L7-маршруты

Маршруты являются основным механизмом публикации HTTP/L7-трафика. Маршрут определяет, какие клиентские запросы должен принять ADC и куда они должны быть направлены.

Блок / поле	Назначение
Основная информация	Имя, описание, метки, приоритет и статус маршрута.
HTTP-методы	Ограничение маршрута выбранными методами: GET, POST, PUT, DELETE, PATCH, HEAD, OPTIONS, CONNECT, TRACE, PURGE.
WebSocket	Разрешение WebSocket upgrade на маршруте.

Блок / поле	Назначение
URI и URI (множественные)	Основные пути и шаблоны сопоставления HTTP-запросов.
Host и Hosts	Ограничение маршрута конкретными виртуальными хостами.
Удаленный адрес	Ограничение маршрута источником клиентского IP или CIDR.
Vars (JSON)	Дополнительные условия сопоставления: заголовки, параметры, cookie, переменные gateway.
Функция фильтрации	Продвинутый механизм кастомного сопоставления. Используется только при наличии проверенного кода.
Сервис	Привязка маршрута к существующему сервису.
Upstream	Привязка к существующему Upstream или inline-настройка backend-пула.
Плагины	Подключение Plugin Config или inline-плагинов маршрута.

Пример Vars (JSON) для сопоставления заголовка:

```
[
  ["http_x_env", "==", "prod"]
]
```

Примечание. Приоритет маршрута следует использовать осторожно. Если несколько маршрутов подходят под один запрос, более высокий приоритет может изменить выбранную конфигурацию обработки трафика.

5.5 Потоковые маршруты

Потоковые маршруты применяются для L4/stream-трафика: TCP, TLS, UDP и протоколов, которые не маршрутизируются как обычный HTTP. Потоковый маршрут сопоставляет входящее соединение по адресу и порту сервера, remote address и SNI, затем передает соединение в Service или Upstream.

Блок / поле	Назначение
Адрес сервера	Локальный адрес, на котором ADC принимает входящие соединения.
Порт сервера	Локальный порт приема stream-трафика.
Удаленный адрес	Ограничение маршрута клиентскими IP-адресами или CIDR.
SNI	Имя SNI для TLS-соединений и выбора маршрута.
ID сервиса	Ссылка на сервис, если stream-маршрут должен использовать готовую сервисную конфигурацию.
ID Upstream	Ссылка на существующий backend-пул.
Inline-настройки Upstream	Полная настройка backend-пула внутри потокового маршрута.
Плагины	Stream-плагины, применяемые к потоку.
Протокол	Дополнительная протокольная конфигурация: имя протокола, Superior ID, JSON-конфигурация и логгер.

6 Управление идентификацией клиентов API

6.1 Потребители

Потребитель является идентификатором клиента API или приложения. Потребители используются вместе с плагинами, которые определяют клиента и применяют к нему индивидуальные политики: key-auth, jwt-auth, basic-auth, ldap-auth, rate limiting, логирование и другие механизмы.

Поле	Назначение
Имя пользователя	Уникальный идентификатор потребителя. В режиме редактирования обычно заблокирован.
Описание	Назначение клиента, владелец, интеграция или канал поддержки.
Метки	Группировка и фильтрация потребителей.
ID группы потребителей	Связь потребителя с группой, в которой заданы общие плагины.
Плагины	Индивидуальная конфигурация плагинов потребителя.

- Используйте стабильное имя потребителя, не зависящее от временных ключей доступа.
- В описании фиксируйте владельца потребителя и назначение интеграции.
- Общие настройки выносите в группы потребителей, а индивидуальные - в потребителя.

6.2 Группы потребителей

Группа потребителей объединяет нескольких потребителей и позволяет применять к ним общий набор плагинов. Это удобно для тарифных планов, клиентских сегментов, окружений или команд.

Поле	Назначение
ID	Уникальный идентификатор группы. В режиме редактирования поле заблокировано.
Описание	Назначение группы, например premium-customers или internal-services.
Метки	Группировка и фильтрация групп.
Плагины	Общие плагины группы, применяемые к связанным потребителям.

7 Управление TLS и секретами

7.1 SSL-сертификаты

Раздел «Безопасность -> SSL-сертификаты» предназначен для управления TLS-сертификатами, приватными ключами, SNI и разрешенными TLS-протоколами. Сертификаты используются для входящего TLS-трафика, TLS-терминации, TLS bridging, mTLS-сценариев и TLS до upstream.

Поле	Назначение
Описание	Назначение сертификата, домены, владелец, срок действия или дата перевыпуска.
Метки	Группировка сертификатов и поиск.
Тип	Серверный сертификат для входящего TLS или клиентский сертификат для mTLS к backend-у.
Статус	Включение или отключение сертификата.
TLS-протоколы	Список разрешенных версий TLS, например TLSv1.2 и TLSv1.3.
SNI	Основное имя сервера и дополнительные SNI-имена.
Сертификат (PEM)	PEM-сертификат или цепочка сертификатов в корректном порядке.
Приватный ключ (PEM)	PEM-приватный ключ. Значение является чувствительным.

- Проверяйте соответствие SNI и CN/SAN сертификата.
- Не используйте просроченные сертификаты и заранее планируйте перевыпуск.

- Для публичных сервисов используйте актуальные версии TLS, например TLSv1.2/TLSv1.3.
- Разграничивайте серверные и клиентские сертификаты по типу и меткам.

7.2 Секреты и внешние менеджеры секретов

Раздел «Безопасность -> Секреты» используется для подключения внешних менеджеров секретов. Секреты позволяют не хранить чувствительные значения непосредственно в обычных конфигурациях плагинов или gateway-сущностей.

Менеджер	Основные параметры
HashiCorp Vault	URI, префикс, токен, namespace.
AWS Secrets Manager	Access Key ID, Secret Access Key, Session Token, регион, Endpoint URL.
Google Cloud Secret Manager	SSL-проверка, файл авторизации JSON, Client Email, приватный ключ, Project ID, Token URI, Entries URI, Scope.

Примечание. Токены, приватные ключи и secret access key являются чувствительными значениями. Доступ к ним должен предоставляться только уполномоченным администраторам.

8 Управление конфигурацией и плагинами

8.1 Глобальные правила

Глобальные правила позволяют включить плагины для всего трафика ADC. Плагины в глобальном правиле выполняются независимо от конкретного маршрута или сервиса. Такой механизм удобно использовать для централизованного логирования, трассировки, базовых security-заголовков или других общих политик.

Поле	Назначение
ID	Уникальный идентификатор глобального правила. В режиме редактирования заблокирован.
Плагины	Набор плагинов, которые будут выполняться глобально.

Примечание. В глобальные правила следует выносить только действительно общие политики. Ошибка в глобальном правиле может затронуть все маршруты и сервисы.

8.2 Конфигурации плагинов

Конфигурации плагинов представляют собой переиспользуемые наборы плагинов. Их можно подключать к маршрутам и потоковым маршрутам через поле ID конфигурации плагинов.

Поле	Назначение
ID	Уникальный идентификатор конфигурации.
Имя	Человекочитаемое название набора плагинов.
Описание	Назначение конфигурации и область применения.
Метки	Группировка и поиск конфигураций.
Плагины	Набор плагинов и их JSON-конфигурации.

Используйте Plugin Config, если один и тот же набор плагинов нужно применять к нескольким маршрутам. Например, единая конфигурация логирования, CORS, rate limiting или заголовков безопасности.

8.3 Метаданные плагинов

Метаданные плагинов задают глобальные параметры конкретного плагина. Такой механизм используется для плагинов, которым требуются общие настройки, не относящиеся к одному маршруту или сервису.

Поле	Назначение
Имя плагина	Плагин, для которого задаются metadata. В режиме создания выбирается из списка поддерживаемых плагинов.
Конфигурация (JSON)	JSON-конфигурация метаданных плагина.

8.4 Proto-файлы

Раздел Protos хранит содержимое .proto-файлов. Эти файлы используются в сценариях, где plugin/protocol-конфигурация должна знать структуру protobuf/gRPC-сообщений.

Поле	Назначение
ID	Идентификатор proto-файла. В режиме редактирования отображается и заблокирован.
Содержимое (.proto)	Текст .proto-файла.

Пример содержимого .proto-файла:

```
syntax = "proto3";
package example;

message Hello {
  string name = 1;
}
```

9 Типовые сценарии использования

9.1 Публикация HTTP API

Для публикации HTTP API через ADC выполните следующие действия:

1. Перейдите в «Трафик -> Upstreams».
2. Создайте Upstream и добавьте backend-ноды приложения.
3. Выберите схему HTTP или HTTPS, алгоритм балансировки и Pass Host.
4. Включите активные проверки здоровья, например по пути /health со статусом 200.
5. Сохраните Upstream.
6. Перейдите в «Трафик -> Маршруты».
7. Создайте маршрут, укажите имя, приоритет и статус «Включён».
8. В правилах маршрутизации задайте URI, например /api/v1/*, Host и HTTP-методы.
9. В блоке Upstream выберите созданный backend-пул.
10. При необходимости подключите Plugin Config или inline-плагины.
11. Сохраните маршрут и проверьте доступность API.

9.2 Использование общего backend-пула

Если несколько маршрутов должны вести в один backend, рекомендуется использовать общий Upstream или Service.

1. Создайте самостоятельный Upstream с backend-нодами.
2. Создайте сервис и привяжите его к этому Upstream.
3. Добавьте общие плагины на уровне сервиса, если они нужны всем маршрутам.
4. Создайте несколько маршрутов и в каждом включите привязку к существующему сервису.
5. Для каждого маршрута задайте собственный URI, Host, методы и дополнительные условия.

Такой подход снижает риск расхождения backend-настроек между маршрутами и упрощает сопровождение публикаций.

9.3 Подключение TLS-сертификата

1. Перейдите в «Безопасность -> SSL-сертификаты».
2. Нажмите «+ Добавить».
3. Укажите тип «Серверный» и статус «Включён».
4. Добавьте разрешенные TLS-протоколы, например TLSv1.2 и TLSv1.3.
5. В поле SNI укажите домен, например api.example.com.
6. Вставьте сертификат в формате PEM и приватный ключ в формате PEM.
7. Сохраните сертификат.
8. Создайте или обновите маршрут с Host, соответствующим SNI сертификата.

9.4 Публикация TCP/TLS-сервиса

1. Перейдите в «Трафик -> Потоковые маршруты».
2. Нажмите «+ Добавить».
3. В блоке сервера задайте адрес сервера, порт сервера, при необходимости удаленный адрес и SNI.
4. Укажите ID сервиса, ID Upstream или настройте inline-Upstream.
5. Если backend доступен по TLS, выберите соответствующую схему соединения и настройте TLS.
6. При необходимости добавьте stream-плагины и протокольную конфигурацию.
7. Сохраните потоковый маршрут и проверьте подключение.

10 Диагностика и рекомендации по эксплуатации

10.1 Диагностика типовых проблем

Симптом	**Что проверить**
Запрос не попадает в маршрут	URI, Host, HTTP-методы, удаленный адрес, Vars (JSON), приоритеты маршрутов и статус маршрута.
Получается 404 или маршрут не найден	Наличие подходящего маршрута и корректность wildcard-паттерна URI.
Ошибка соединения с backend	Upstream-ноды, порт, схема соединения, DNS, network ACL и таймаут Connect.
Частые 502/503/504	Состояние backend-а, health checks, таймауты Read/Send, retries и плагины, влияющие на доступность.
Трафик идет не на тот backend	Привязку маршрута к сервису или Upstream, Pass Host, приоритеты маршрутов и условия match rules.
TLS-соединение не устанавливается	SSL-сертификат, SNI, приватный ключ, TLS-протоколы и цепочку сертификатов.

Симптом	Что проверить
WebSocket не работает	Переключатель WebSocket на маршруте или сервисе, заголовки Upgrade/Connection и поддержку WebSocket на backend-е.
Плагин не сработал	Уровень подключения плагина, приоритеты Consumer/Group/Route/Plugin Config/Service, корректность JSON-конфигурации и subsystem для stream.
Пассивная проверка не возвращает ноду в healthy	Порог passive healthy и наличие активной проверки восстановления.

10.2 Общие рекомендации

- Для каждого production-сервиса задавайте понятные имена объектов и метки, отражающие приложение, среду и владельца.
- Выносите общие backend-настройки в самостоятельные Upstream и Service, если они используются несколькими маршрутами.
- Включайте активные health checks для критичных backend-пулов и используйте отдельные lightweight endpoint-ы.
- Документируйте назначение глобальных правил и плагинов, так как они могут влиять на большой объем трафика.
- Проверяйте JSON-конфигурации перед сохранением и храните согласованные шаблоны для типовых плагинов.
- Не храните чувствительные значения в открытом виде, если доступна интеграция с менеджером секретов.
- Перед изменением маршрутов, сертификатов или глобальных правил оценивайте влияние на работающие публикации.
- После изменения конфигурации проверяйте статистику, Error Rate, Latency Percentiles и состояние Upstream.