ООО «Веб-Сервер», ОГРН 1227700436578, ИНН 9704151517
ул. Вятская, д. 27, стр. 7, г. Москва, 127015


# Описание функциональных характеристик

*версия 1.2*

ООО «Веб-Сервер»

сент. 25, 2026

# Оглавление

ГЛАВА 1

# Аннотация

Настоящий документ содержит описание функциональных характеристик Angie ADC.

Angie ADC — программное обеспечение класса "контроллер доставки приложений", которое представляет собой систему балансировки, включающее DNS-балансировку, а также позволяющее маршрутизировать и балансировать сетевые запросы, используя протоколы маршрутизации внешнего и внутреннего шлюза.

ГЛАВА 2

# Обзор решения и возможности

Angie ADC — комплексное программное решение для балансировки нагрузки и управления сетевым трафиком, включающее балансировку на уровнях L4-L7, интеллектуальную доставку приложений и маршрутизацию, а также глобальную DNS-балансировку (GSLB) и решения для обеспечения высокой доступности.

Angie ADC поддерживает мониторинг и управление через веб-интерфейс и командную строку (CLI), а также предоставляет API для интеграции с внешними системами.

Решение Angie ADC поставляется в виде аппаратного балансировщика и виртуального устройства (Virtual Appliance) и развертывается в *кластере*.

## 2.1 Функциональность Angie ADC

В таблице ниже приведены основные функции и возможности Angie ADC, а также ссылки на соответствующие разделы документации.

| Функции | Документация |
|---|---|
| Локальная балансировка нагрузки | *Балансировщик нагрузки* |
| Глобальная балансировка нагрузки (GSLB) | *Глобальная балансировка* |
| Решения для обеспечения высокой доступности и маршрутизации | *IP-маршрутизация*: настройки высокой доступности для standalone-устройств Angie ADC<br>*Пара высокой доступности*: развертывание высокодоступного решения на базе двух Angie ADC |
| Инструменты для мониторинга и диагностики | *Мониторинг работы балансировщика нагрузки*<br>*Мониторинг системных метрик Angie ADC*<br>*Мониторинг доступности узлов и бэкендов*<br>*Диагностика сети*<br>*Syslog*, *SNMP* |
| Инструменты для управления и безопасности | Управление через веб-интерфейс и CLI<br>*Кластер управления*: администрирование нескольких Angie ADC<br>*Резервное копирование и восстановление системы*<br>*Ролевое разграничение доступа к системе*<br>*Управление сертификатами*<br>*Парольная политика*<br>*Настройка учетной записи*<br>*Аудит действий пользователей* |
| Поддерживаемые протоколы и механизмы | *BGP*, *OSPF*, *VRRP*, *RHI*, RIP, *BFD* |

ГЛАВА 3

# Управление трафиком и сервисы балансировки

Здесь описывается настройка балансировщика нагрузки L4-L7 и глобальной DNS-балансировки (GSLB), а также приведены примеры решения типовых задач.

В этом разделе:

- *Балансировщик нагрузки*
- *Глобальная балансировка*
- *Типовые задачи и примеры*

## 3.1 Балансировщик нагрузки

В этом разделе:

- *Настройка конфигурации*
- *Управление резервными конфигурациями*
- *Методы балансировки*
- *Предоставление доступов на работу с конфигурацией*
- *Проверки работоспособности серверов*
- *Резервирование проксируемых серверов*
- *Аутентификация клиентов*
- *Управление сертификатами*
- *Режим отладки*

См. также *Типовые задачи и примеры*.

### Балансировка

Балансировщик нагрузки позволяет распределять клиентские запросы по серверам на основании выбранного алгоритма балансировки, данных статистики или обратной связи от сервера (см. *Методы балансировки*). В идеале клиентские запросы равномерно распределяются по всем серверам, что увеличивает совокупную производительность системы. Доступна балансировка на двух уровнях: L7 (протокол HTTP) и L4 (TCP и UDP).

#### Высокая доступность

Высокая доступность при балансировке обеспечивается за счет *проверок работоспособности апстрим-серверов* и *резервирования проксируемых серверов и групп*.

> **Примечание**
>
> Отказоустойчивость самого балансировщика нагрузки обеспечивается отдельно, например с помощью *пары высокой доступности*.

#### Дополнительная настройка

Помимо алгоритмов распределения запросов, в HTTP- и TCP/UDP-балансировке могут использоваться смежные модули:

- Модуль `queue` создает очередь запросов. Если балансировщик не смог выбрать апстрим-сервер для обработки, то запрос попадает в очередь ожидания. Также можно задать приоритизацию запросов в очереди. Приоритетные запросы будут обрабатываться в первую очередь, что позволит в условиях критической перегрузки сохранить обслуживание наиболее важных клиентов, отсекая второстепенные запросы и повышая устойчивость к DDoS-атакам.
- Модуль `keepalive` позволяет держать открытые неактивные соединения с апстрим-серверами, т. е. не закрывать соединения при завершении обработки запроса.
- Директива `bind_conn`, реализованная в модуле `keepalive`, позволяет привязать клиентское соединение к серверному соединению.
- Модуль `sticky` также позволяет привязывать клиентское соединение к апстрим-серверу, но на основе информации, содержащейся в самом запросе.
- Модуль `zone` обеспечивает доступ с использованием REST API. API позволяет получать набор метрик, в том числе относящихся к балансировке.
- Модуль `upstream_probe` позволяет обеспечить активные проверки апстрим-серверов.

Подробнее см. в справочнике HTTP-модуля Upstream и потокового модуля Upstream.

#### Хранение версий конфигурации

Все версии конфигурации балансировщика нагрузки сохраняются. Вы можете просмотреть историю изменений, а также *восстановить конфигурацию* из резервной копии.

#### Справочная информация

В разделе adc_lb-reference представлены сведения о встроенных и сторонних модулях, примеры их настройки, а также поддерживаемые ими директивы и переменные.

### 3.1.1 Настройка конфигурации

В этой статье:

- *Общая информация*;
- *Текстовый конфигуратор*;
- *Визуальный конфигуратор*;
- *Перезапуск балансировщика*;
- *Логи балансировщика*;
- *Принцип конфигурации*;
- *Использование портов*;
- *Настройка простого прокси-сервера*;
- *Настройка балансировки нагрузки*.

#### 3.1.1.1 Общая информация

Конфигурацию балансировщика нагрузки можно просмотреть и отредактировать в веб-интерфейсе Angie ADC (`Управление трафиком` → `Балансировщик нагрузки`).

В левой части экрана отображается список файлов конфигурации, к которым есть *доступ* пользователя. В правой части экрана отображается конфигурация каждого файла. Поддерживаются *текстовый* и *визуальный* конфигураторы, между ними можно переключаться.

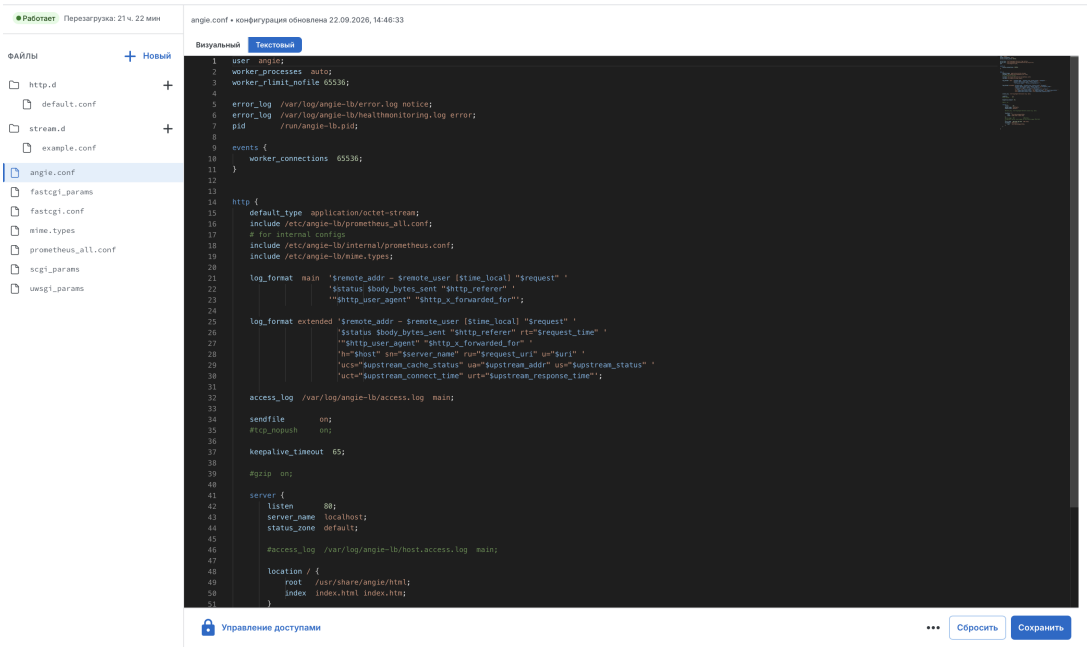
По умолчанию конфигурация балансировщика содержит следующие файлы и папки:

- папка `http.d` — содержит конфигурации веб-сайтов и веб-приложений, работающих по протоколам HTTP/HTTPS:
  - файл `default.conf` — пример настройки виртуального хоста; этот файл не используется.
- папка `stream.d` — настройки проксирования трафика на уровне транспортного протокола (TCP и UDP):
  - файл `example.conf` — пример настройки потокового (TCP/UDP) проксирования; этот файл не используется.
- файл `angie.conf` — главный конфигурационный файл веб-сервера; запускает процессы, определяет пути к логам и подключает все остальные папки и файлы;
- файл `fastcgi.conf` — настройки для работы с PHP (через PHP-FPM) и другими FastCGI-приложениями;
- файл `fastcgi_params` — аналог fastcgi.conf, содержащий базовые переменные окружения для FastCGI;
- файл `mime.types` — карта соответствия расширений файлов и их типов данных (например .html → text/html, .jpg → image/jpeg);
- файл `prometheus_all.conf` — файл настройки экспорта метрик в формате Prometheus для мониторинга (например, в Grafana);
- файл `scgi_params` — параметры для работы по протоколу SCGI;
- файл `uwsgi_params` — параметры для интеграции с приложениями на Python.

Администратор может добавлять новые файлы в папки `http.d` и `stream.d`, нажав на значок добавления напротив соответствующей папки, и удалять любые существующие файлы, нажав на три точки в нижней части экрана соответствующего файла и выбрав пункт `Удалить файл`.

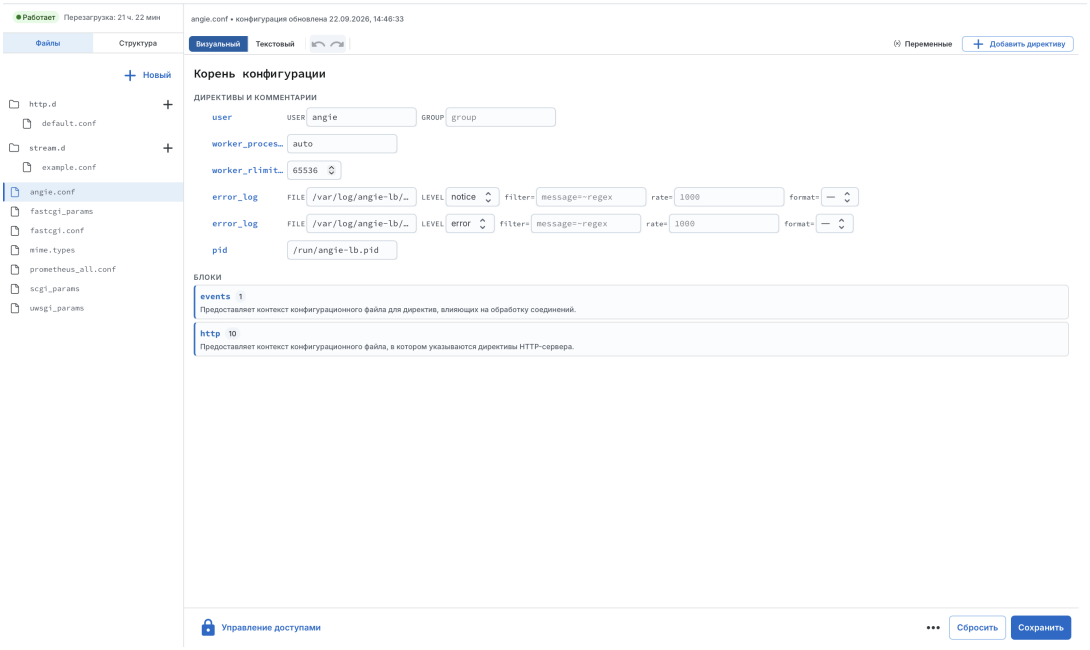
При сохранении изменений конфигурация обновляется и применяется сразу. Если при сохранении файла возникает конфликт (другой пользователь уже внес изменения в файл), то система подсвечивает те места, где правки конфликтуют, и предлагает выбрать, какую версию файла сохранить: пользовательскую (`Сохранить мою версию`) или серверную (`Принять серверную версию`).

### 3.1.1.2 Текстовый конфигуратор




В текстовом конфигураторе конфигурация редактируется как обычный текст: директивы Angie вводятся и изменяются вручную. Содержимое выбранного файла отображается с подсветкой синтаксиса и нумерацией строк.

### 3.1.1.3 Визуальный конфигуратор




Визуальный конфигуратор позволяет задать конфигурацию, выбирая директивы и параметры из меню, а не вводя их вручную. На основе заданных значений конфигуратор формирует итоговую конфигурацию балансировщика; просмотреть и при необходимости отредактировать ее вручную можно, переключившись в текстовый режим.

#### 3.1.1.4 Перезапуск балансировщика

В верхней части экрана отображается статус балансировщика:

- `Работает` — балансировщик запущен;
- `Не работает` — балансировщик не запустился после перезагрузки конфигурации.

Также отображается время, прошедшее с последней перезагрузки (`Перезагрузка`) и количество перезагрузок при наведении на время (`Поколение`).

> **Совет**
>
> Если у балансировщика отображается статус `Не работает`, необходимо проверить журнал ошибок балансировщика (см. ниже) и устранить ошибки.
>
> Также можно попробовать перезапустить балансировщик вручную. Для этого откройте любой файл конфигурации балансировщика и сохраните его без внесения изменений.

#### 3.1.1.5 Логи балансировщика

В веб-интерфейсе можно просмотреть журнал ошибок балансировщика нагрузки. Для этого перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` и в нижней части экрана щелкните многоточие (`...`) и выберите `Логи`.

Также см. *Журналы событий и экспорт* и adc_cli-diagnostics.

#### 3.1.1.6 Принцип конфигурации

Для настройки балансировщика нагрузки используется декларативная конфигурация. Балансировщик нагрузки состоит из модулей, которые настраиваются директивами, заданными в конфигурации. Директивы могут быть простыми и блочными. Простая директива состоит из имени и параметров, разделенных пробелами, и оканчивается точкой с запятой `;`. В блочной директиве вместо точки с запятой после имени и параметров следует набор дополнительных инструкций внутри фигурных скобок `{` и `}`. Блочные директивы называются контекстами.

Примеры:

- контекст events (общая обработка соединений);
- контекст http (конфигурация HTTP-балансировки);
- контекст stream (конфигурация TCP/UDP балансировки).

Контексты `events`, `http`, `stream` располагаются в контексте main. Другие директивы, размещенные вне контекстов `events`, `http`, `stream`, считаются также находящимися в контексте `main`.

В контекстах http и stream размещаются:

- блоки `server`, в которых настраивается проксирование трафика;
- блоки `upstream`, в которых настраиваются группы upstream-серверов для балансировки нагрузки.

В конфигурации в веб-интерфейсе Angie ADC уже предустановлены некоторые общие параметры. Вам необходимо дополнить конфигурацию с учетом ваших целей.

Подробнее о конфигурации и обработке соединений см. adc__configfile и adc__processing.

> **Предупреждение**
>
> Строки конфигурации, приведенные ниже, нельзя удалять или менять (они используются для обеспечения работы соответствующего функционала Angie ADC).
>
> Контекст `main`:

```
error_log  /var/log/angie-lb/healthmonitoring.log error;
pid        /run/angie-lb.pid;
```

Контекст `http`:

```
include /etc/angie-lb/prometheus_all.conf;
include /etc/angie-lb/internal/prometheus.conf;
```

#### 3.1.1.7 Использование портов

Если вы планируете использовать в конфигурации другие порты помимо 80/443, вы можете открыть их с помощью команды open через CLI Angie ADC.

**Предупреждение**

При настройке конфигурации не занимайте следующие порты (используются внутренними сервисами Angie ADC):

TCP: 22, 199, 2022, 2222, 2601, 2602, 2603, 2604, 2605, 2606, 2615, 2616, 2617, 2619, 2623, 3050, 3051, 3053, 3054, 3060, 3111, 3122, 3123, 3301, 3302, 3380, 3391, 4460, 5044, 5355, 5432, 8010, 8080, 8443, 9090, 9100, 9120, 9600, 9633, 9898, 9900, 60080

- Порты 8080 и 8443 можно использовать, если у вас выделен отдельный *интерфейс управления*.

UDP: 53, 123, 161, 323, 546, 3784, 3785, 4784, 5355, 7784

#### 3.1.1.8 Настройка простого прокси-сервера

Angie ADC можно использовать в качестве прокси-сервера, который принимает запросы, перенаправляет их на проксируемые серверы, получает от них ответы и отправляет их клиенту.

В примере ниже настраивается базовый прокси-сервер, который будет обслуживать запросы изображений из локального каталога и отправлять все остальные запросы на проксируемый сервер.

1. Создайте сервер, который будет использоваться в качестве бэкенда для прокси (проксируемый сервер ), добавив блок `server` в контекст `http`:

   ```
   server {
       listen 8081;
       root /static/www;

       location / {
       }
   }
   ```

   Здесь сервер слушает на порту `8081` и сопоставляет URI запросов с файлами в каталоге `/static/www` (как настроить загрузку и раздачу статических файлов см. соотв. статью). Директива root находится в контексте `server` и будет использоваться, если директива location, выбранная для выполнения запроса, не содержит собственной директивы `root`.

   **Примечание**

   В общем случае конфигурационный файл может содержать несколько блоков `server`, отличающихся портами, на которых они слушают (listen), и именами сервера (server_name). Определив, какой `server` будет обрабатывать запрос, Angie ADC сравнивает URI запроса с параметрами location, заданными в блоке `server`, и принимает решение.

2. Добавьте конфигурацию прокси-сервера:

```
server {
    listen 80;
    location / {
        proxy_pass http://localhost:8081/;
    }

    location ~ \.(gif|jpg|png)$ {
        root /static/images;
    }
}
```

В первом блоке `location` добавлена директива adc__proxy_pass с указанием протокола, имени и порта проксируемого сервера. Во втором блоке `location` задано регулярное выражение для всех URI, оканчивающихся на `.gif`, `.jpg` или `.png`. Соответствующие запросы будут обслуживаться из каталога `/static/images`.

> **Примечание**
>
> При выборе блока `location`, который будет обслуживать запрос, сначала проверяются директивы `location`, задающие префиксы, и запоминается `location` с самым длинным подходящим префиксом. Затем проверяются блоки `location`, заданные регулярными выражениями, в порядке их объявления в конфигурации и используется первое совпадение. Иначе берется `location`, сохраненный до этого.

3. Сохраните изменения в конфигурации, нажав на кнопку `Сохранить`.

Получившийся прокси-сервер будет обслуживать запросы к файлам `.gif`, `.jpg` или `.png` из каталога `/static/images`, а все остальные запросы будет проксировать на сервер `localhost:8081`.

#### 3.1.1.9 Настройка балансировки нагрузки

Чтобы настроить балансировку нагрузки для HTTP-трафика, добавьте блок upstream в контекст `http` и укажите в нем бэкенд-серверы, которые будут использоваться для балансировки нагрузки. Затем настройте проксирование запросов на эту группу серверов в блоке `server` с помощью директивы location.

Пример:

```
upstream backend {
    zone backend 1m;
    server backend1.example.com;
    server backend2.example.com;
}

server {
    listen 80;
    server_name localhost;

    location /api {
        proxy_pass http://backend;
    }
}
```

В приведенном примере задана группа серверов `backend`. Все запросы на порт `80` с URI, начинающимися с `/api`, проксируются в эту группу. Метод балансировки нагрузки не указан явно, поэтому по умолчанию используется метод `round-robin` (запросы распределяются по серверам последовательно).

### 3.1.2 Управление резервными конфигурациями

В Angie ADC реализовано хранение всех версий конфигурационных файлов балансировщика нагрузки. При сохранении изменений в любом файле конфигурации балансировщика Angie ADC автоматически применяет новую конфигурацию и сохраняет предыдущую версию в репозитории. Можно просмотреть список сохраненных версий конфигурации и применить нужную версию. Также поддерживается *выбор внутреннего или внешнего репозитория* для хранения конфигураций.

#### 3.1.2.1 Просмотр списка сохраненных конфигураций

Чтобы просмотреть список сохраненных конфигураций:

1. Откройте веб-интерфейс Angie ADC.
2. Выберите `Управление трафиком` → `Балансировщик нагрузки`.
3. В нижней части экрана щелкните многоточие (`...`) и выберите `История`.

Откроется таблица со списком сохраненных конфигураций. Для каждой конфигурации указано время ее создания, версия Angie ADC, статус (рабочая или нерабочая конфигурация), пользовательский комментарий, теги (`angie-lb` и UUID) и дата последнего изменения.

#### 3.1.2.2 Изменение рабочего статуса хранимой конфигурации

Чтобы пометить конфигурацию как рабочую или нерабочую:

1. Откройте веб-интерфейс Angie ADC.
2. Выберите `Управление трафиком` → `Балансировщик нагрузки`.
3. В нижней части экрана щелкните многоточие (`...`) и выберите `История`.

   Откроется таблица со списком сохраненных конфигураций.
4. Щелкните многоточие (`...`) напротив конфигурации, статус которой вы хотите изменить, и нажмите `Пометить как нерабочую` или `Пометить как рабочую`.
5. Если вы устанавливаете нерабочий статус, укажите причину поломки и нажмите `Сохранить`.

   Статус конфигурации будет изменен.

#### 3.1.2.3 Восстановление последней рабочей версии конфигурации

Чтобы применить последнюю рабочую версию конфигурации:

1. Откройте веб-интерфейс Angie ADC.
2. Выберите `Управление трафиком` → `Балансировщик нагрузки`.
3. В нижней части экрана щелкните многоточие (`...`) и выберите `Применить последнюю рабочую версию`.

   Будет применена последняя рабочая конфигурация.

#### 3.1.2.4 Восстановление конфигурации из резервной копии

Чтобы применить произвольную конфигурацию из резервной копии:

1. Откройте веб-интерфейс Angie ADC.
2. Выберите `Управление трафиком` → `Балансировщик нагрузки`.
3. В нижней части экрана щелкните многоточие (`...`) и выберите `История`.

   Откроется таблица со списком сохраненных конфигураций.
4. Щелкните многоточие (`...`) напротив конфигурации, которую вы хотите использовать, и нажмите кнопку `Применить`.

   Конфигурация будет применена сразу.

#### 3.1.2.5 Выбор репозитория для хранения конфигураций

В Angie ADC поддерживается два типа репозитория для хранения *резервных копий конфигурации балансировщика нагрузки*, локальный (`filesystem`) и внешний (S3). Для надежного хранения данных рекомендуется использовать внешний репозиторий. Резервная копия в любом репозитории обновляется автоматически при внесении изменений и сохранении новой конфигурации балансировщика нагрузки.

*Локальный репозиторий* `filesystem` подключен по умолчанию. При использовании локального репозитория резервные копии хранятся в файловой системе внутри Angie ADC.

> **Примечание**
>
> При подключении нового репозитория (локального или внешнего) резервная копия в локальном репозитории автоматически удаляется.

*Внешний репозиторий* подключается по протоколу S3. Мы рекомендуем использовать внешний S3-совместимый репозиторий для хранения резервных копий конфигурации. В отличие от локального репозитория, при отключении внешнего репозитория резервные копии в нем сохраняются.

**Просмотр информации о текущем репозитории:**

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → многоточие (`...`) → `История` → `Управление резервным копированием`.

   Откроется страница с информацией о текущем репозитории.

**Подключение локального репозитория:**

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → многоточие (`...`) → `История` → `Управление резервным копированием`.

   Откроется страница с информацией о текущем репозитории.

2. Перейдите на вкладку `Filesystem`.
3. Нажмите `Подключить`.

Будет подключен локальный репозиторий и создана резервная копия конфигурации балансировщика.

> **Примечание**
>
> Если у вас до этого уже была сохранена резервная копия в `filesystem`, она будет удалена.

**Подключение внешнего репозитория (S3):**

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → многоточие (`...`) → `История` → `Управление резервным копированием`.

   Откроется страница с информацией о текущем репозитории.

2. Перейдите на вкладку `S3`.
3. Заполните поля для подключения репозитория:

   - `Конечная точка`
   - `Имя корзины`
   - `Ключ доступа`

- `Секретный токен`

4. Нажмите `Подключить`.

   Если вы подключаете репозиторий, в котором уже была резервная копия конфигурации, то будет использована эта копия. Если подключается новый репозиторий, будет создана резервная копия текущей конфигурации балансировщика.

> **Примечание**
>
> Если у вас до этого уже была сохранена резервная копия в `filesystem`, она будет удалена.

### 3.1.3 Методы балансировки

> **Примечание**
>
> Для каждой upstream-группы можно задать только один метод балансировки.

| Алго-ритм | Описание | HT- | TCP/UDP |
|---|---|---|---|
| round-rc | Запросы распределяются по серверам последовательно (используется по умолчанию). Поддерживается учет веса серверов (`weight`), медленный старт (`slow_start`), backup-группы серверов. | | |
| hash | Задает метод балансировки нагрузки для группы, при котором соответствие клиента серверу определяется при помощи хэшированного значения ключа. Поддерживается учет веса серверов (`weight`). Подробнее см. TCP/UDP, HTTP. | | |
| ip_hash | Сервер выбирается с помощью хэш-функции на основе IP-адреса клиента, что обеспечивает "липкость сессии". Частный случай `hash`. В качестве ключа для хэширования используются первые три октета IPv4-адреса клиента или IPv6-адрес клиента целиком. В итоге запрос конкретного клиента всегда попадает на один и тот же сервер, если он работоспособен. Поддерживается учет веса серверов (`weight`). Подробнее см. HTTP. | | |
| least_cc | Новый запрос направляется на сервер с минимальным количеством активных соединений (наименее загруженный). Поддерживается учет веса серверов (`weight`), медленный старт (`slow_start`), backup-группы серверов. Подробнее см. TCP/UDP, HTTP. | | |
| random | Запросы распределяются по серверам случайным образом. При большом числе запросов они будут распределяться по upstream-серверам равномерно, что хорошо подходит, например, для кластера балансировщиков. Поддерживается учет веса серверов (`weight`) и опциональный параметр `two` (выбор из двух случайных серверов по `least_conn`). Подробнее см. TCP/UDP, HTTP. | | |
| least_ti | Балансировка нагрузки на основе наименьшего времени ответа. Запросы распределяются на те серверы, которые обрабатывают их быстрее. Уменьшает время ожидания для пользователей. Если серверы имеют разное оборудование или настройки, помогает уравновесить нагрузку. Поддерживаются backup-группы серверов. Подробнее см. TCP/UDP, HTTP. | | |
| feedback | Балансировка нагрузки по обратной связи (по произвольному параметру из ответа на основной или проверочный запрос). Поддерживается учет веса серверов (`weight`), медленный старт (`slow_start`), backup-группы серверов. Подробнее см. TCP/UDP, HTTP. | | |
| least_ba | Балансировка на основе наименьшего потребления полосы пропускания. Режимы: upstream, downstream, both. Поддерживается учет веса серверов (`weight`), медленный старт (`slow_start`), backup-группы серверов. Подробнее см. TCP/UDP, HTTP. | | |
| least_pa | Балансировка на основе наименьшего числа пакетов в единицу времени. Может использоваться в случаях, когда оборудование, расположенное за балансировщиком, не способно "переварить" слишком большой поток пакетов. Режимы: upstream, downstream, both. Поддерживается учет веса серверов (`weight`), медленный старт (`slow_start`), backup-группы серверов. Подробнее см. TCP/UDP. | | |

#### 3.1.3.1 Примеры HTTP-балансировки нагрузки

##### Базовая настройка

Самая простая конфигурация для балансировки нагрузки с помощью Angie ADC может выглядеть следующим образом:

```
http {

    upstream myapp1 {

        server srv1.example.com;
        server srv2.example.com;
```

```
        server srv3.example.com;
    }
    server {

        listen 80;
        location / {

            proxy_pass http://myapp1;
        }
    }
}
```

В приведенном примере:

- Группа серверов `myapp1` содержит три сервера `srv1`, `srv2` и `srv3`.
- На этих серверах работают три экземпляра одного и того же приложения.
- Используется метод балансировки `round-robin` (по умолчанию).
- Все запросы на порт `80` перенаправляются через группу серверов `myapp1`.

### least_conn

Балансировка нагрузки по наименее загруженному серверу.

Модуль балансировки нагрузки `least_conn` в модуле HTTP позволяет распределять нагрузку более равномерно, направляя новые запросы на серверы с наименьшим количеством активных соединений. Этот метод рекомендуется, если время обработки запросов может сильно варьироваться.

Пример конфигурации:

```
http {

    upstream backend {

        least_conn;

        server backend1.example.com;
        server backend2.example.com;
        server backend3.example.com;
    }

    server {

        listen 80;
        location / {

            proxy_pass http://backend;
        }
    }
}
```

В этом примере:

- Директива `least_conn` включает метод балансировки нагрузки по наименьшему числу соединений.
- Для балансировки нагрузки определены три проксируемых сервера.
- Входящие запросы к серверу проксируются в группу `backend`.

Преимущества метода:

- Динамическая балансировка: непрерывно отслеживает количество активных соединений для адаптации к текущей нагрузке на каждом сервере.
- Простота: работает без необходимости дополнительных параметров или сложных настроек.
- Совместимость: совместим с другими конфигурациями проксируемых серверов, включая веса, максимальное количество сбоев и время ожидания после сбоя.

### ip_hash

Балансировка нагрузки с сохранением сессий.

Методы `round-robin` и `least-conn` не гарантируют, что запросы от одного клиента будут обрабатываться одним и тем же сервером. Если важно, чтобы клиент всегда направлялся на один сервер (например, при использовании сессионных данных), можно использовать метод `ip_hash`.

Пример конфигурации:

```
upstream myapp1 {

    ip_hash;
    server srv1.example.com;
    server srv2.example.com;
    server srv3.example.com;
}
```

Метод `ip_hash` гарантирует, что запросы от одного клиента будут попадать на один сервер, если этот сервер доступен.

### weight

Взвешенная балансировка нагрузки.

С помощью директивы `weight` можно указать, какой сервер должен обрабатывать больше запросов. Этот способ рекомендуется, если серверы имеют разную производительность. По умолчанию вес сервера равен `1`.

Пример:

```
upstream myapp1 {

    server srv1.example.com weight=3;
    server srv2.example.com;
    server srv3.example.com;
}
```

В этой конфигурации каждые пять запросов будут распределены следующим образом: три запроса попадут на `srv1`, один запрос — на `srv2` и еще один — на `srv3`.

Метод `weight` может использоваться в комбинации с методами `round-robin` и `least-conn`.

### least_time

Балансировка нагрузки на основе наименьшего времени ответа.

Модуль балансировки нагрузки `least_time` задает для группы метод балансировки нагрузки, при котором вероятность передачи соединения активному серверу обратно пропорциональна среднему времени его ответа; чем оно меньше, тем больше соединений будет получать сервер.

Когда использовать `least_time`?

- Переменная нагрузка: запросы распределяются на серверы, которые обрабатывают их быстрее.
- Высокая доступность: уменьшается время ожидания для пользователей.

- Гетерогенные серверы: если серверы имеют разное оборудование или настройки, `least_time` помогает уравновесить нагрузку.

Пример конфигурации:

```
http {

    upstream backend {

        least_time header;
        server backend1.example.com;
        server backend2.example.com;
        server backend3.example.com;
    }
    server {

        listen 80;
        location / {

            proxy_pass http://backend;
        }
    }
}
```

В этом примере `least_time header`: балансировка основана на времени отклика до получения заголовка ответа от сервера.

### feedback

Балансировка по произвольному параметру из ответа на основной или проверочный запрос.

Модуль балансировки нагрузки `feedback` Задает в `upstream` механизм балансировки нагрузки по обратной связи. Он динамически корректирует решения при балансировке, умножая вес каждого проксируемого сервера на среднее значение обратной связи, которое меняется с течением времени в зависимости от значения переменной и подчиняется необязательному условию. По умолчанию параметр `factor` равен 90.

Пример конфигурации:

```
upstream backend {

    zone backend 1m;

    feedback $feedback_value factor=80 account=$condition_value;

    server backend1.example.com;
    server backend2.example.com;
}

map $upstream_http_custom_score $feedback_value {

    "high" 100;
    "medium" 75;
    "low" 50;
    default 10;
}

map $upstream_probe $condition_value {

    "high_priority" "1";
```

```
    "low_priority" "0";
    default "1";
}
```

Эта конфигурация категоризирует ответы серверов по уровням обратной связи на основе определенных оценок из полей заголовков ответа, а также добавляет условие на `$upstream_probe`, чтобы учитывать только ответы от активной проверки `high_priority` или ответы на обычные клиентские запросы.

#### 3.1.3.2 Примеры TCP/UDP-балансировки нагрузки

##### Конфигурация для TCP

```
stream {

    upstream tcp_backend {

        server 192.168.1.10:3306 weight=2;
        server 192.168.1.11:3306;
        server 192.168.1.12:3306;
    }

    server {

        listen 3306;
        proxy_pass tcp_backend;
        proxy_timeout 10s;
        proxy_connect_timeout 5s;
    }
}
```

В этой конфигурации задана группа серверов, участвующих в балансировке (`tcp_backend`). Настроен вес для серверов, чтобы более мощный сервер получал больше запросов (`weight=2`), Сервер слушает входящие соединения на порту `3306` (используется для MySQL). Трафик перенаправляется к группе серверов `tcp_backend`. Настроены тайм-ауты:

- максимальное время ожидания ответа от сервера `10s`;
- максимальное время для установления соединения с сервером `5s`.

##### Конфигурация для UDP

Для приложений, работающих через протокол UDP, конфигурация аналогична TCP, но с настройками для UDP-трафика.

```
stream {

    upstream udp_backend {

        server 192.168.1.20:53;
        server 192.168.1.21:53;

    }
    server {

        listen 53 udp;
        proxy_pass udp_backend;

        proxy_responses 1;
```

```
        proxy_timeout 10s;
    }
}
```

### least_bandwidth

Балансировка на основе наименьшего потребления полосы пропускания.

Алгоритм периодически вычисляет среднее использование полосы пропускания для каждого апстрим-сервера, используя формулу скользящего среднего для сглаживания колебаний со временем. Когда начинается новая сессия, модуль балансировки нагрузки `least_bandwidth` выбирает проксируемый сервер с наименьшим средним использованием пропускной способности, скорректированным с учетом веса сервера, если он указан. Этот механизм гарантирует, что сессии направляются на серверы, которые в настоящее время используют меньше пропускной способности, что способствует равномерному распределению сетевой нагрузки.

Пример конфигурации:

```
stream {

    upstream backend {

        least_bandwidth both factor=80;
        server backend1.example.com:12345;
        server backend2.example.com:12345;
    }
    server {

        listen 12345;
        proxy_pass backend;
    }
}
```

В этом примере:

- В директиве `least_bandwidth` настроен учет как входящей, так и исходящей пропускной способности.
- Коэффициент сглаживания установлен равным 80, что влияет на скорость адаптации среднего значения к изменениям в использовании пропускной способности.
- Для балансировки нагрузки определены два проксируемых сервера.

Преимущества метода:

- Динамическая балансировка: непрерывно отслеживает использование пропускной способности для адаптации к текущей нагрузке на каждом сервере.
- Коэффициент сглаживания `factor` контролирует отзывчивость расчета скользящего среднего; более высокий коэффициент означает более медленную адаптацию к изменениям.
- Режимы: позволяет выбирать балансировку на основе входящей пропускной способности, исходящей пропускной способности или обоих вариантов, в зависимости от потребностей приложения.
- Совместим с другими конфигурациями проксируемых серверов, включая веса, максимальное количество сбоев и время ожидания после сбоя.

### least_packets

Балансировка на основе наименьшего числа пакетов в единицу времени.

Средняя скорость пакетов для каждого проксируемого сервера периодически проверяется с использованием формулы скользящего среднего для сглаживания колебаний со временем. Когда инициируется новый сеанс, модуль балансировки нагрузки `least_packets` в потоковом модуле выбирает проксируемый сервер с наименьшей средней скоростью передачи пакетов, скорректированной с учетом веса сервера. Этот процесс гарантирует, что сеансы направляются на серверы, которые в настоящее время наименее загружены с точки зрения обработки пакетов, что способствует равномерному распределению сетевой нагрузки.

Пример конфигурации:

```
stream {

    upstream backend {

        least_packets both factor=80;
        server backend1.example.com:12345;
        server backend2.example.com:12345;
    }
    server {

        listen 12345;
        proxy_pass backend;
    }
}
```

В этом примере:

- В директиве `least_packets` настроен учет как входящих, так и исходящих пакетов.
- Коэффициент сглаживания установлен равным 80, что влияет на то, как быстро среднее значение адаптируется к изменениям скорости пакетов.
- Определены два проксируемых сервера для балансировки нагрузки.

Преимущества метода:

- Динамическая балансировка: непрерывно отслеживает скорость пакетов для адаптации к текущей нагрузке на каждом сервере.
- Параметр `factor` контролирует отзывчивость расчета скользящего среднего; более высокий коэффициент означает более медленную адаптацию к изменениям.
- Режимы: позволяет выбирать балансировку на основе числа входящих пакетов, исходящих пакетов или обоих значений.
- Совместим с другими конфигурациями проксируемых серверов, включая веса, максимальное количество сбоев и время ожидания после сбоя.

### hash

Контекстное хэширование.

Поддерживается метод `hash` (например, по IP или произвольному значению). Если сервер из пула станет недоступным или будет добавлен новый сервер, минимальное количество клиентов перенаправится на другие серверы. Используется для "липких сессий", когда важно, чтобы запросы от одного клиента всегда обрабатывались одним и тем же сервером.

Пример:

```
upstream tcp_backend {
```

```
    hash $remote_addr consistent;
    server 192.168.1.10:3306;
    server 192.168.1.11:3306;
}
```

#### Ограничение IP

Для защиты можно использовать ограничения IP через `allow` и `deny`.

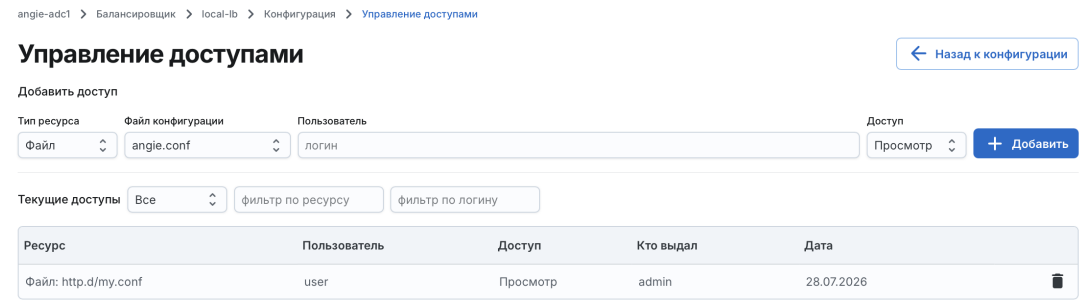
Пример:

```
server {

    listen 3306;
    allow 192.168.1.0/24;
    deny all;
    proxy_pass tcp_backend;
}
```

В этом примере разрешена только локальная сеть, остальным доступ запрещен.

### 3.1.4 Предоставление доступов на работу с конфигурацией




Всем пользователям Angie ADC (локальным и внешним) назначаются роли в системе, определяющие их доступ к конфигурации балансировщика нагрузки.

- **Администратор** обладает полными правами на все действия в системе, включая редактирование всех файлов конфигурации балансировщика нагрузки Angie ADC.
- **Супервизор** имеет ограниченные права: может только просматривать информацию в системе, например проводить аудит, в том числе может просматривать все файлы конфигурации балансировщика нагрузки.
- **Оператор** имеет ограниченные права: может редактировать файлы конфигурации балансировщика, к которым ему предоставлен доступ, и просматривать статистику зон, к которым ему предоставлен доступ. Доступы предоставляет администратор.

#### 3.1.4.1 Предоставление прав на просмотр и редактирование

Чтобы *оператор мог настраивать балансировку для сервера*, ему необходимо разрешить редактирование файла конфигурации, в котором описан этот сервер. Права на просмотр и редактирование файлов предоставляет администратор через веб-интерфейс Angie ADC.

**Примечание**

Рекомендуется давать операторам доступ только к файлам в папках `http.d` и `stream.d`.

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → `Управление доступами`.
2. В открывшемся окне заполните поля формы для добавления нового доступа:
   - Чтобы предоставить доступ к файлу, выберите тип ресурса `Файл` и в выпадающем списке `Файл конфигурации` выберите нужный файл. Затем в поле `Доступ` укажите уровень доступа: `Просмотр` или `Редактирование`.
   - Чтобы предоставить доступ к зоне, выберите тип ресурса `Зона` и введите имя зоны в поле `Зона`. Для типа ресурса `Зона` всегда выдается доступ на редактирование.

   **Примечание**

   Зону предварительно необходимо настроить.
3. В поле `Пользователь` укажите логин пользователя, для которого настраивается доступ.
4. Нажмите `Добавить`.

Доступ будет добавлен и отобразится в таблице доступов.

#### 3.1.4.2 Просмотр выданных доступов

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → `Управление доступами`.
2. В открывшемся окне в таблице `Текущие доступы` отобразится список выданных доступов.

   Для удобства поиска поддерживается фильтрация.

#### 3.1.4.3 Редактирование выданных доступов

Редактирование записей не поддерживается.

Чтобы изменить запись, необходимо удалить старую запись и добавить новую с измененными параметрами.

#### 3.1.4.4 Удаление выданных доступов

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` → `Управление доступами`.
2. В открывшемся окне в таблице `Текущие записи` отобразится список выданных доступов.
3. Нажмите на значок удаления напротив нужной записи.

Доступ будет удален.

### 3.1.5 Проверки работоспособности серверов

Angie ADC включает пассивные и активные проверки работоспособности серверов (health probes). Если сервер начинает отвечать с ошибками, Angie ADC временно исключает его из пула доступных серверов.

#### 3.1.5.1 Пассивные проверки

Пассивные проверки работают автоматически на основе реальных клиентских запросов.

Пример:

```
upstream myapp1 {

   server srv1.example.com max_fails=2 fail_timeout=10s;
   server srv2.example.com;
   server srv3.example.com;
}
```

В этой конфигурации сервер помечается как недоступный после двух неудачных запросов подряд (`max_fails=2`). Через десять секунд сервер снова проверяется на доступность (`fail_timeout=10s`). Если он восстановился, запросы снова начинают отправляться на этот сервер.

#### 3.1.5.2 Активные проверки

Активные проверки позволяют оценить доступность сервера без участия клиентов. Поддерживаются активные проверки для HTTP и TCP / UDP. Проверки задаются с помощью директивы `upstream_probe`. Для одного апстрима можно определить несколько проверок. Можно гибко настраивать поведение проверок. Поддерживается отправка ICMP echo-запросов (ping). Подробнее см. справочные материалы по активным проверкам HTTP и stream.

##### Пример для HTTP

```
upstream backend {
    zone backend 1m;

    server backend1.example.com;
    server backend2.example.com;
}

map $upstream_status $good {
    200     "1";
}

server {
    listen ...;

    location /backend {
        ...
        proxy_pass http://backend;

        upstream_probe backend_probe
            uri=/probe
            port=10004
            interval=5s
            test=$good
            essential
            persistent
            fails=3
            passes=3
            max_body=10m
            mode=idle;
    }
}
```

Детали работы:

- Изначально сервер не получает клиентские запросы, пока не пройдет *все* заданные для него проверки с параметром `essential` (пропуская помеченные как `persistent`, если конфигурация перезагружена и до этого сервер считался работающим). Если таких проверок нет, сервер

считается работающим.

- Сервер считается неработающим и не получает клиентские запросы, если *какая-либо* заданная для него проверка достигает своего порога `fails` или сам сервер достигает порога max_fails.
- Чтобы неработающий сервер снова мог считаться работающим, *все* заданные для него проверки должны достичь своего порога `passes`; после этого учитывается порог max_fails.

### Пример для TCP/UDP

```
upstream backend {
    zone backend 1m;

    server a.example.com;
    server b.example.com;
}

map $upstream_probe_response $good {
    ~200    "1";
    default  "";
}

server {
    listen ...;

    # ...
    proxy_pass backend;
    upstream_probe_timeout 1s;

    upstream_probe backend_probe
        port=12345
        interval=5s
        test=$good
        essential
        persistent
        fails=3
        passes=3
        max_response=512k
        mode=onfail
        "send=data:GET / HTTP/1.0\r\n\r\n";
}
```

**Примечание**

Согласно спецификации RFC 2616 (HTTP/1.1) и RFC 9110 (HTTP Semantics), заголовки HTTP должны разделяться последовательностью CRLF (`\r\n`), а не просто `\n`.

Детали работы:

- Изначально сервер не получает клиентские запросы, пока не пройдет *все* заданные для него проверки с параметром `essential` (пропуская помеченные как `persistent`, если конфигурация перезагружена и до этого сервер считался работающим). Если таких проверок нет, сервер считается работающим.
- Сервер считается неработающим и не получает клиентские запросы, если *какая-либо* заданная для него проверка достигает своего порога `fails` или сам сервер достигает порога max_fails.
- Чтобы неработающий сервер снова мог считаться работающим, *все* заданные для него проверки должны достичь своего порога `passes`; после этого учитывается порог max_fails.

- Параметр `send` - отправляемые для проверки данные, а переменная `$upstream_probe_response` – ответ бекенда.

#### Пример ICMP echo-запроса для HTTP

```
http {
    upstream u {
        zone z 1k;

        server 127.0.0.1:8081;
        server 127.0.0.1:8082;
    }

    server {
        listen ...;

        location / {
            proxy_pass http://u/;

            upstream_probe simple1
                 interval=3s
                 ping;

            upstream_probe simple2
                 interval=3s
                 ping ping_timeout=3s;
        }
    }
}
```

Детали работы:

- Используется проверка методом ICMP Echo вместо HTTP-запроса.
- ICMP-проверка проверяет доступность IP-адреса `127.0.0.1`. Порты `upstream` игнорируются.
- Сервер считается неработающим и не получает клиентские запросы, если не отвечает на запрос ICMP Echo в течение заданного таймаута. Первая проверка использует таймаут ожидания по умолчанию (1 секунда). Вторая проверка задает таймаут ожидания ответа явно (3 секунды).
- Единственным критерием прохождения проверки является успешное получение ICMP-ответа в пределах таймаута.

### 3.1.6 Резервирование проксируемых серверов и групп

Для повышения отказоустойчивости системы вы можете дополнительно резервировать проксируемые серверы, чтобы они включались в работу при выходе основных серверов из строя.

В Angie ADC доступно два механизма резервирования:

- *Запасные серверы* (spare-серверы).

  Запасные серверы (spare-серверы) включаются в работу при выходе из строя основных серверов в рамках одной группы. Это позволяет плавно подхватывать трафик без переключения на другую группу серверов. Запасные серверы можно использовать для поддержания производительности системы, например при выводе отдельных серверов на техническое обслуживание.

- *Резервные группы серверов* (backup-группы).

  Резервные группы (backup-группы) серверов включаются в работу при отказе всех серверов активной группы и поддерживают отказоустойчивость системы при выходе из строя групп

серверов целиком. С помощью backup-групп можно реализовать многоуровневое резервирование большой инфраструктуры.

Можно использовать оба механизма одновременно для обеспечения максимальной надежности.

**Примечание**

Отказоустойчивость самого балансировщика нагрузки обеспечивается отдельно, например с помощью *пары высокой доступности*.

#### 3.1.6.1 Запасные серверы (spare-серверы)

При настройке апстрим-групп часть серверов в группе можно выделить как запасные, назначив им параметр spare в блоке `server`. Также можно объявить серверы запасными динамически с помощью API-интерфейса.

Пример:

```
http {

    upstream u {
        zone z 1m;
        server 127.0.0.1:8081 max_fails=1 fail_timeout=2s weight=2;
        server 127.0.0.1:8082 spare;
        server 127.0.0.1:8083 spare;
    }
```

По умолчанию запасные (spare) серверы находятся в режиме ожидания и имеют статус `idle`. В случае выхода из строя одного или нескольких основных серверов группы они включаются в балансировку и переходят в статус `up`. При восстановлении основных серверов замещающие spare-серверы снова переводятся в режим ожидания. При этом поддерживаются sticky-сессии: запасные серверы, выведенные из активного пула, будут по-прежнему использоваться для таких сессий. Если на запасном сервере настроены активные проверки работоспособности, то сервер не будет включаться в работу, пока проверки не будут пройдены. Неработоспособные запасные серверы будут иметь такие же статусы, как неработоспособные основные.

Решение о вводе spare-сервера в балансировку определяется разницей между совокупным весом всех серверов, используемых в группе в качестве основных `main_weight`, и совокупным весом всех активных серверов `active_weight`. Если общий вес активных серверов становится меньше веса всех основных серверов — spare-серверы вводятся в работу, если больше — spare-серверы выводятся в режим ожидания. При этом всегда используется минимально необходимое количество spare-серверов. При введении дополнительного spare-сервера будет выбран сервер с максимальным весом, а при выведении из строя лишних spare-серверов сначала будет выбран сервер с минимальным весом.

*В режиме отладки* через REST API доступны значения счетчиков `main_weight` и `active_weight`. Для upstream-сервера отображается поле `spare` со значением `idle` (сервер не используется) или `running` (сервер активен). Поле `spare_num` отображает количество серверов, считающихся запасными.

#### 3.1.6.2 Резервные группы серверов (backup-группы)

Резервные группы серверов (backup-группы), в отличие от spare-серверов, представляют собой отдельные группы серверов, выделенные только для резервирования. Upstream-сервер может входить только в одну резервную группу.

Если балансировщик не может найти ни один рабочий сервер в основной группе, то он переходит к серверам backup-группы. Можно добавить одну или несколько backup-групп серверов с разным уровнем резервирования. Тогда балансировщик будет сначала переходить к backup-группе с наименьшим уровнем. Если групп много, можно указать для них произвольные уровни, например для удобства сортировки.

Уровень резервирования backup-группы можно задать с помощью директивы `backup=n`:

- `backup=1` (или просто `backup`) — группа резервных серверов первого уровня (в API отображается как `true`);
- `backup=2` — группа резервных серверов второго уровня (в API отображается как соответствующее число);
- `backup=3` — группа резервных серверов третьего уровня.

Пример:

```
upstream my_backend {
    server primary1.example.com;
    server primary2.example.com;

    server backend.example.com service=http resolve backup=prio;

    server backup1a.example.com backup;
    server backup1b.example.com backup;

    server backup2a.example.com backup=2;
    server backup2b.example.com backup=2;

    backup_switch permanent=2m;
}
```

Если upstream-сервер настраивался с помощью SRV-записи, то уровень backup-группы можно определить динамически через поле приоритета (`priority`). Для этого используется:

- Абсолютное значение приоритета (директива `backup=prio`).

  В этом случае значение приоритета `priority=n` из SRV-записи будет использовано в качестве уровня резервной группы `backup=n`.

- Относительное значение приоритета `priority=n` из SRV-записи.

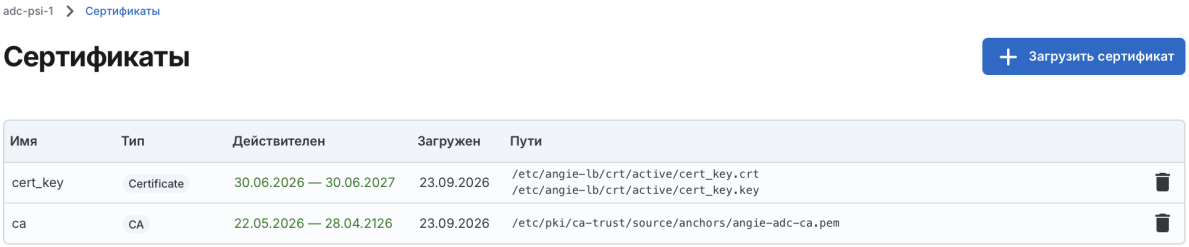
  В этом случае директива `backup=n` задает базовый уровень отсчета (например `backup=3`), а относительные уровни резервирования отсчитываются от этого уровня. Сервер с наименьшим значением `priority` получит уровень `backup=3`, следующий за ним — `backup=4`, и так далее. Если уровень отсчета `backup=n` не задан, отсчет будет идти от нуля: сервер с наименьшим значением `priority` получит `backup=0` и станет основным, за ним — `backup=1`, резервный, и так далее.

По умолчанию балансировщик всегда начинает поиск подходящего сервера с основной группы, даже если в данный момент активна резервная. Чтобы этого избежать, можно использовать директиву `backup_switch permanent`, задав для нее значение, например `permanent=2m`. В этом случае балансировщик нагрузки будет пытаться вернуться на основную группу не чаще, чем раз в две минуты, что позволит избежать избыточных переключений группы.

Если указать параметр `permanent` без значения, то выбранная backup-группа будет сохраняться в качестве активной даже после того, как серверы из основной группы снова станут доступными. В случае отказа текущей backup-группы балансировщик будет переключаться на следующую по уровню backup-группу, и, только дойдя до максимального уровня, вернется на основную группу.

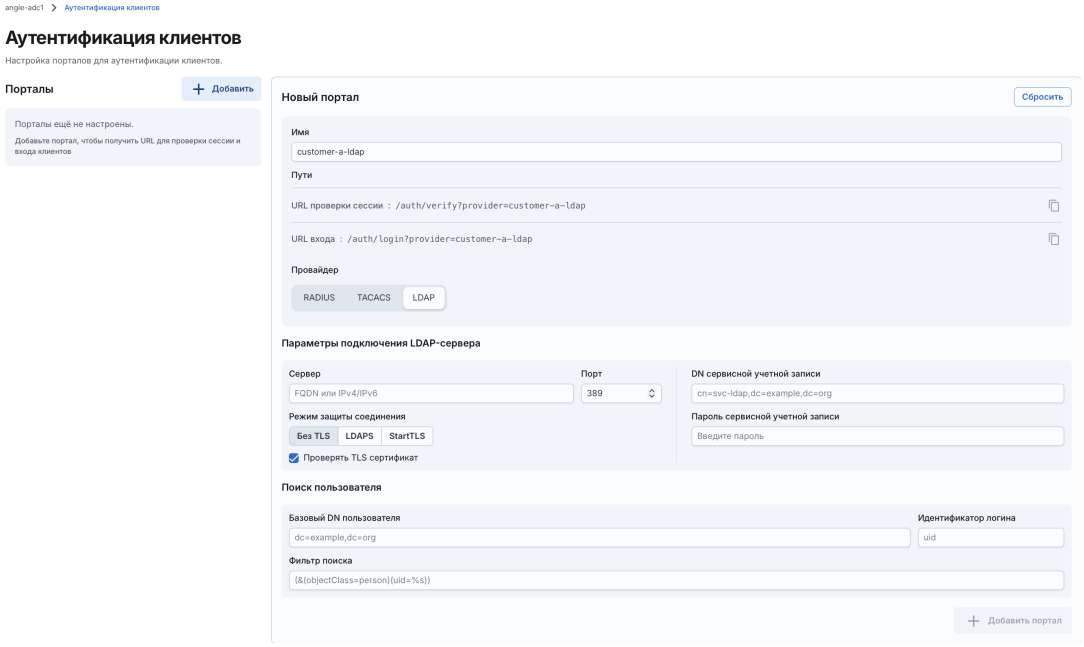
### 3.1.7 Управление сертификатами

В Angie ADC можно загружать сертификаты для обработки TLS-соединений балансируемого трафика и доверенные корневые и промежуточные сертификаты (`Система` → `Сертификаты`). Набор загруженных сертификатов распространяется на все узлы кластера.

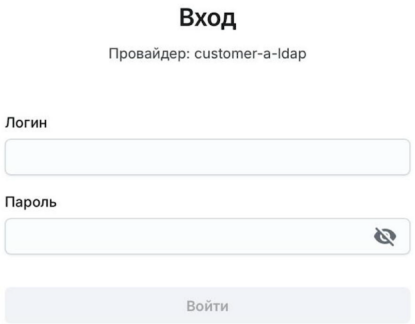



Подробнее см. *Управление сертификатами*.

### 3.1.8 Аутентификация клиентов




Angie ADC позволяет настроить аутентификацию клиентов при обращении к сервисам через балансировщик нагрузки. При доступе к защищенным ресурсам клиент будет попадать на страницу логина (портал аутентификации):




При успешной аутентификации клиент будет перенаправлен на запрашиваемую страницу. Для аутентификации используется внешний сервер (LDAP, TACACS+ или RADIUS). Настройки аутентификации клиентов распространяются на все узлы *кластера*.

#### 3.1.8.1 Настройка аутентификации клиентов

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Аутентификация клиентов`.
2. В открывшемся окне нажмите `Добавить` и укажите имя нового портала для аутентификации. Имя должно быть уникальным и валидным для использования в URL: нижний регистр, начинается с буквы; допускаются буквы, цифры и дефисы, не допускаются пробелы и специальные символы.

   В блоке `Пути` отобразятся URL проверки сессии и URL входа. Эти пути необходимо в дальнейшем указать в конфигурации балансировщика нагрузки (см. пункт 5).
3. Выберите тип провайдера, LDAP, TACACS+ или RADIUS, и задайте параметры подключения к серверу:
   - Для LDAP:
     - `Сервер`: IP-адрес или доменное имя LDAP-сервера.
     - `Порт`: порт подключения к LDAP-серверу (по умолчанию `389`).
     - `Режим защиты соединения` (по умолчанию `Без TLS`, также доступны варианты `LDAPS` и `StartTLS`).
     - `DN сервисной учетной записи`: логин сервисной учетной записи, от имени которой система подключается к LDAP.
     - `Пароль сервисной учетной записи`: пароль сервисной учетной записи для подключения к LDAP-серверу.
     - `Базовый DN пользователя`: задает ветку на LDAP-сервере, в которой система будет искать пользователя, например `ou=users,dc=example,dc=com`.
     - `Идентификатор логина`: поле в LDAP, которое содержит логин, например `uid`, `sAMAccountName` или `userPrincipalName`.
     - `Фильтр поиска`: LDAP-фильтр для поиска пользователя, например `(&(objectClass=user)(sAMAccountName=%s))`.

     **Примечание**

     Фильтр LDAP определяет, в каком формате пользователь должен вводить логин при входе в систему:

     - если используется фильтр `(&(objectClass=user)(sAMAccountName=%s))`, то логин будет без доменного имени, например `ivanov`;
     - если используется фильтр `(&(objectClass=user)(userPrincipalName=%s)`, то логин будет полным: `ivanov@domain.com`.
   - Для RADIUS:
     - `Сервер`: IP-адрес или доменное имя RADIUS-сервера.
     - `Порт`: порт подключения к RADIUS-серверу (по умолчанию `1812`).
     - `Протокол`: протокол передачи данных: `UDP` или `TCP` (по умолчанию: `UDP`).
     - `NAS-идентификатор`: идентификатор NAS (Network Access Server), передаваемый серверу RADIUS.
     - `Метод аутентификации`: поддерживается только `PAP`.
     - `Секрет`: общий секрет для шифрования паролей.
   - Для TACACS+:
     - `Адрес`: IP-адрес или доменное имя TACACS-сервера.

- `Порт`: порт подключения к TACACS-серверу (по умолчанию 49).
- `Секрет`: общий секрет для шифрования паролей.

4. Нажмите `Добавить портал`.
5. Перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки` и укажите пути портала аутентификации для нужного `location`, а также страницу ошибки в случае непрохождения аутентификации.

   Например:

```
include /etc/angie-lb/internal/adc-ui-portal-locations.inc;

location /app/ {
    auth_request /internal-verify;
    error_page 401 = @auth_login;
    proxy_pass http://<backend-address>;
    proxy_set_header Host              $host;
    proxy_set_header X-Forwarded-Proto <http или https>;
}

location = /internal-verify {
    internal;
    proxy_set_header X-Angie-Auth-Request 1;
    proxy_pass http://unix:/run/angie-adc-auth-dataplane/auth-dataplane.sock:/
↪auth/verify?provider=<portal-name>;
}

location @auth_login {
    return 302 /auth/portal/login?provider=<portal-name>&redirect_url=$request_
↪uri;
}
```

#### 3.1.8.2 Изменение параметров портала аутентификации

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Аутентификация клиентов`.
2. В открывшемся окне в списке порталов выберите портал, параметры которого вы хотите изменить.
3. Внесите изменения и нажмите `Сохранить портал`.

> **Важно**
>
> Если портал использовался для аутентификации в конфигурации балансировщика нагрузки, не забудьте отредактировать конфигурацию соответствующим образом.

#### 3.1.8.3 Удаление портала аутентификации

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Аутентификация клиентов`.
2. В открывшемся окне в списке порталов выберите портал, который вы хотите удалить, и нажмите `Удалить`.

Портал будет удален.

**Важно**

Если портал использовался для аутентификации в конфигурации балансировщика нагрузки, не забудьте удалить его из конфигурации, иначе пользователи будут по-прежнему перенаправляться на страницу аутентификации, но не смогут ее пройти, т.к. портал удален.

### 3.1.9 Режим отладки

Режим отладки следует включать перед самостоятельной диагностикой или по рекомендации технической поддержки. Чтобы включить режим отладки, выполните команду `angie debug` в CLI. Для отключения режима отладки выполните команду `no angie debug`.

**Примечание**

Использование режима отладки может снизить производительность, а также увеличить расход места на диске.

#### 3.1.9.1 Отладочный лог

Чтобы включить отладочный лог, задайте в конфигурации уровень `debug` с помощью директивы adc__error_log:

```
error_log /path/to/log debug;
```

Если выключить режим отладки, но оставить уровень `debug` в директиве adc__error_log, Angie ADC будет добавлять записи в лог на уровне `info`.

Если переопределить adc__error_log в конфигурации, но не указать в ней уровень `debug`, отладочный лог будет отключен. Здесь переопределение лога на уровне adc__server отключает отладочный лог для отдельного сервера:

```
error_log /path/to/log debug;

http {
   server {
     error_log /path/to/log;
    # ...
```

Во избежание этого уберите строку, переопределяющую adc__error_log, либо задайте в ней уровень `debug`:

```
error_log /path/to/log debug;

http {
   server {
     error_log /path/to/log debug;
   #  ...
```

#### 3.1.9.2 Расположение директивы

Расположение директивы adc__error_log влияет на полноту собираемой отладочной информации.

Директива, указанная на более низком уровне конфигурации (например, внутри блока `server` или `location`), заменяет настройки логирования, заданные на более высоком уровне (например, на основном уровне конфигурации или внутри блока `http`).

### Отключение отладочного лога для конкретного сервера

Если глобально отладочный лог включен, но для отдельного сервера adc__error_log указан без уровня `debug`, то для этого сервера отладочная информация собираться не будет.

```
error_log /var/log/angie-lb/error.log debug; # Глобальный отладочный лог

http {

    server {

        listen 80;
        server_name example.com;

        error_log /var/log/angie-lb/example.com.error.log;
        # Отладочный лог для example.com отключен, в файле - уровень info

        # ...
    }

    server {

        listen 80;
        server_name another.com;

        # Для этого сервера будет использоваться глобальный отладочный лог
        # ...
    }
}
```

### Сохранение отладочного лога для сервера

Чтобы сохранить сбор отладочной информации для конкретного сервера, но направить ее в другой файл, необходимо также указать уровень `debug`:

```
error_log /var/log/angie-lb/error.log debug; # Глобальный отладочный лог

http {

    server {

        listen 80;
        server_name example.com;

        error_log /var/log/angie-lb/example.com.error.log debug;
        # Отладочный лог для example.com включен, но пишется в отдельный файл

        # ...
    }
}
```

Таким образом, чтобы включить отладочный лог глобально, но переопределить файл лога для отдельных блоков, также укажите уровень `debug` в этих переопределениях. Иначе, если в директиве adc__error_log не указан уровень логирования, по умолчанию будет использоваться уровень `error` и отладочная информация для этих блоков будет потеряна.

#### 3.1.9.3 Лог для отдельных адресов

Можно включить отладочный лог только для указанных клиентских адресов:

```
error_log /path/to/log;

events {
  debug_connection 192.168.1.1;
  debug_connection 192.168.10.0/24;
}
```

## 3.2 Глобальная балансировка

Глобальная балансировка, также GSLB (Global Server Load Balancing) – это сервис, который позволяет распределять запросы клиентов между несколькими дата-центрами на основе их доступности и производительности.

Для работы глобальной балансировки собирается кластер, настраивается конфигурация GSLB и делегирование зон на GSLB-серверы Angie ADC. Локальный DNS-сервер будет отправлять клиентские запросы для этих зон на один из GSLB-серверов. Сервер GSLB решает, какой дата-центр лучше обслужит запрос, выбирает сервер и передает его IP-адрес DNS-серверу. DNS-сервер отправляет этот IP-адрес клиенту, после чего клиент направляет свой запрос на новый сервер по полученному адресу.

Поддерживаются динамические записи типа A, AAAA и CNAME и статические записи типа A, AAAA, CNAME, MX, TXT.

Для динамических записей поддерживаются следующие типы балансировки:

- `round_robin` — запросы распределяются по серверам последовательно.
- `weighted round_robin` — запросы распределяются по серверам последовательно с учетом весов серверов (на серверы с большим весом запросы будут попадать чаще).
- `failover` — запрос направляется на доступный сервер с наивысшим приоритетом.

> **Примечание**
>
> Доступность бэкендов контролируется *мониторами доступности*. Мониторы рекомендуется настроить заранее.

Подробнее см: *Настройка конфигурации*.

Возможности GSLB в Angie ADC:

- Гибкая настройка балансировки: поддержка статических и динамических записей, поддержка нескольких записей для одного домена.
- Масштабируемость: простота добавления новых доменов и записей через веб-интерфейс.
- Отказоустойчивость: непрерывная доступность приложений даже при сбоях в одном из дата-центров, возможность настройки TTL.
- Кластеризация: упрощение администрирования нескольких GSLB-модулей в разных дата-центрах.

В этом разделе:

- *Настройка сервиса GSLB*
- *Настройка конфигурации*

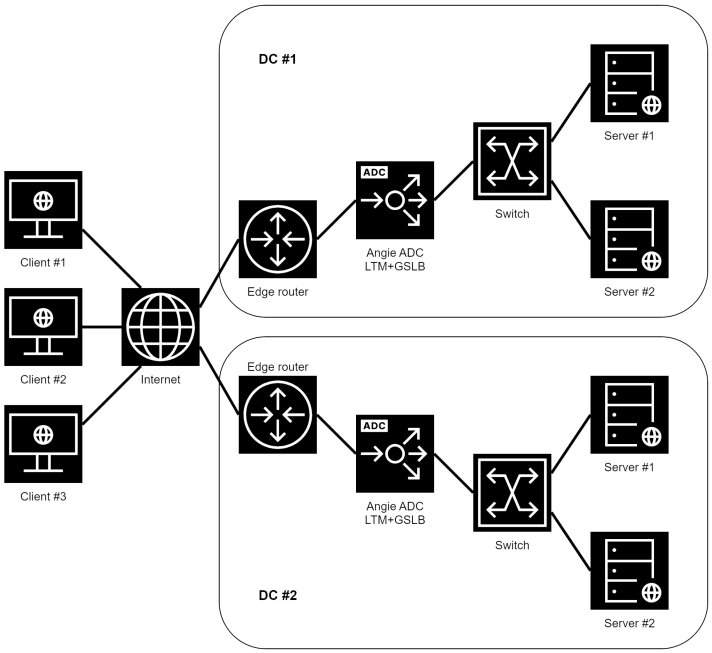



Рис. 1: Рис. 1: Пример развертывания кластера из двух Angie ADC с GSLB в двух дата-центрах

### 3.2.1 Настройка сервиса GSLB

Настройка GSLB состоит из следующих шагов:

1. Настройка *мониторов доступности бэкендов*. Доступность бэкендов учитывается при GSLB-балансировке для обеспечения отказоустойчивости.
2. Настройка *конфигурации GSLB* в веб-интерфейсе (на любом из узлов кластера):
   - Включение функции глобальной балансировки.
   - Настройка IP-адресов ADNS, если необходимо.
   - Добавление доменов, для которых требуется выполнять глобальную балансировку.
   - Добавление DNS-записей для каждого домена.
   - Применение конфигурации и включение функции глобальной балансировки.

   Синхронизация конфигурации GSLB на другие узлы кластера происходит автоматически.
3. Настройка делегирования заданных доменов на GSLB-узлы Angie ADC на первичном DNS-сервере.

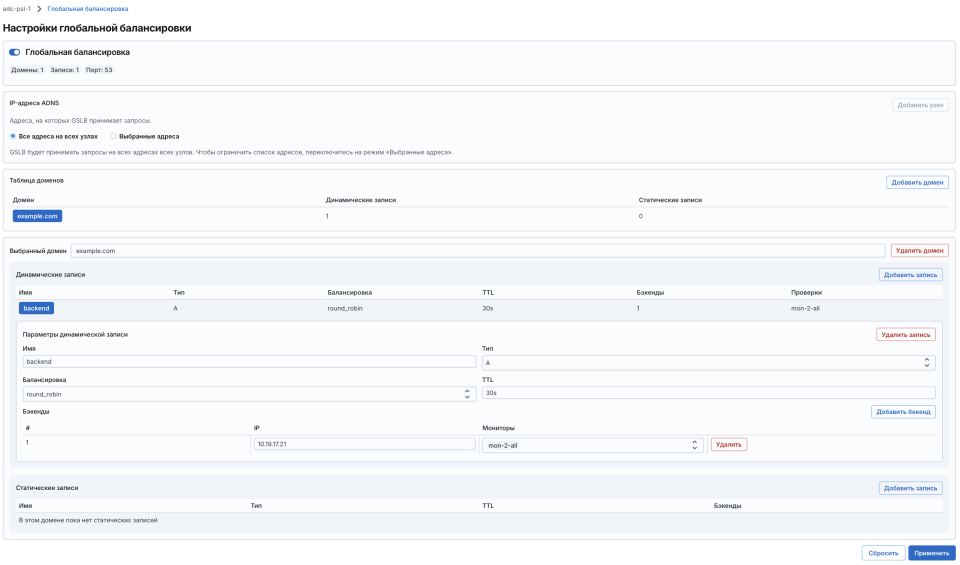
   Пример для `www.example.com` — делегирование зоны `example.com` на GSLB-узлы Angie ADC в родительской зоне `com`:

```
www.example.com.       IN NS adc-dc1.example.com.
www.example.com.       IN NS adc-dc2.example.com.

; Glue-записи - IP-адреса GSLB-узлов Angie ADC
adc-dc1.example.com.  IN A  <IP-адрес dc1>
adc-dc2.example.com.  IN A  <IP-адрес dc2>
```

### 3.2.2 Настройка конфигурации

Конфигурация GSLB настраивается в веб-интерфейсе Angie ADC (`Управление трафиком` → `Глобальная балансировка`).




#### 3.2.2.1 Предварительные шаги

Перед настройкой конфигурации GSLB необходимо настроить *мониторы бэкендов*. Эти мониторы будут использоваться в динамических записях для проверки доступности бэкендов и обеспечения отказоустойчивости.

#### 3.2.2.2 Настройка IP-адресов ADNS

По умолчанию компонент GSLB принимает DNS-запросы на всех адресах всех узлов кластера Angie ADC. Список адресов можно ограничить: для каждого узла можно задать список локальных прослушиваемых IP-адресов.

Чтобы задать список прослушиваемых IP-адресов:

1. Откройте веб-интерфейс Angie ADC и перейдите в раздел `Управление трафиком` → `Глобальная балансировка`.

   Откроется окно настройки глобальной балансировки.

2. В блоке `IP-адреса ADNS` выберите режим `Выбранные адреса`.

3. Нажмите `Добавить узел`.

4. Укажите имя узла и хотя бы один IP-адрес (IPv4 или IPv6).

   > **Примечание**
   >
   > Проверяется только формат адреса. Не предусмотрена проверка наличия адреса на узле.

5. Повторите операцию для каждого узла, список адресов которого вы хотите ограничить.

   > **Примечание**
   >
   > На узлах, не добавленных в таблицу, будут по-прежнему прослушиваться все адреса.

6. Завершите настройку GSLB, добавив необходимые домены и записи (см. ниже).
7. Нажмите `Применить` в нижней части экрана, чтобы применить конфигурацию.

Статус применения конфигурации будет отображаться в верхней части экрана.

> **Примечание**
>
> При переключении на режим `Все адреса на всех узлах` и применении конфигурации список ранее настроенных IP-адресов для каждого узла будет сброшен.

#### 3.2.2.3 Добавление домена и создание записей

Чтобы добавить домен и настроить для него глобальную балансировку, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC и перейдите в раздел `Управление трафиком` → `Глобальная балансировка`.

   Откроется окно настройки глобальной балансировки.

2. В блоке `Таблица доменов` нажмите `Добавить домен` и укажите имя домена, например `example.com`. Имя домена должно быть корректным DNS-именем без завершающей точки.
3. Настройте записи для добавленного домена. Выберите тип записи (статическая или динамическая) и укажите необходимые параметры.

> **Совет**
>
> - Чтобы настроить балансировку для домена, а не только для записей внутри него, создайте динамическую запись типа `A` с именем `@` и укажите необходимые параметры балансировки.
> - Чтобы задать общее правило балансировки для разных поддоменов, создайте динамическую запись типа `A` с именем `*` и укажите необходимые параметры балансировки.

### Параметры динамической записи типа A и AAAA

| | |
|---|---|
| `Имя` | Имя записи. Должно быть уникальным в рамках одного домена. |
| `Тип` | Тип записи. Возможные значения:<br>• `A` — сопоставляет доменное имя с IPv4-адресом.<br>• `AAAA` — сопоставляет доменное имя с IPv6-адресом. |
| `Балансировка` | Тип балансировки. Возможные значения:<br>• `round_robin` — запросы распределяются по серверам последовательно.<br>• `weighted round_robin` — запросы распределяются по серверам последовательно с учетом весов серверов (на серверы с большим весом запросы будут попадать чаще).<br>• `failover` — запрос направляется на доступный сервер с наивысшим приоритетом. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. Все последующие запросы в течение времени TTL будут идти строго на выбранный адрес. По умолчанию установлено значение `30s`. |
| `Бэкенды` | Таблица бэкенд-серверов. Необходимо задать как минимум один бэкенд. Для каждого бэкенд-сервера задаются параметры:<br>• `IP`: IP-адрес сервера. Поддерживаются IPv4 и IPv6-адреса.<br>• `Мониторы`: один или несколько мониторов доступности, привязанных к бэкенду. Выбираются из списка *настроенных мониторов*. Если задано несколько мониторов, бэкенд считается доступным, когда исправны все заданные мониторы.<br>• `Приоритет`: приоритет сервера (необходим для режима `failover`). Чем меньше значение, тем выше приоритет сервера, которому это значение назначено.<br>• `Вес`: вес сервера (необходим для режима `weighted round_robin`). На серверы с большим весом запросы будут попадать чаще. |

### Параметры динамической записи типа CNAME

| | |
|---|---|
| `Имя` | Имя записи. |
| `Тип` | Тип записи `CNAME` — задает псевдоним для другого доменного имени. |
| `Балансировка` | Тип балансировки. Возможные значения:<br>• `round_robin` — система проверяет доступность серверов и распределяет запросы по серверам последовательно.<br>• `weighted round_robin` — система проверяет доступность серверов и распределяет запросы по серверам последовательно с учетом веса (на серверы с большим весом запросы будут попадать чаще).<br>• `failover` — система проверяет доступность серверов и выбирает лучший по приоритету. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. Все последующие запросы в течение времени TTL будут идти строго на выбранный домен. По умолчанию установлено значение `30s`. |
| `Бэкенды` | Таблица бэкенд-серверов.<br>• `Host`: доменное имя, на которое будет указывать CNAME-запись.<br>• `Мониторы`: один или несколько мониторов доступности, привязанных к бэкенду. Выбираются из списка *настроенных мониторов*. Если задано несколько мониторов, бэкенд считается доступным, когда исправны все заданные мониторы.<br>• `Приоритет`: приоритет сервера (необходим для режима `failover`). Чем меньше значение, тем выше приоритет сервера, которому это значение назначено.<br>• `Вес`: вес сервера (необходим для режима `weighted round_robin`). На серверы с большим весом запросы будут попадать чаще. |

### Параметры статической записи типа A и AAAA

| | |
|---|---|
| `Имя` | Имя записи. |
| `Тип` | Тип записи. Возможные значения:<br>• `A` — сопоставляет доменное имя с IPv4-адресом.<br>• `AAAA` — сопоставляет доменное имя с IPv6-адресом. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. По умолчанию установлено значение `300s`. |
| `Бэкенды` | Таблица бэкенд-серверов. Для каждого бэкенд-сервера задаются его параметры:<br>• `IP`: IP-адрес сервера. |

#### Параметры статической записи типа CNAME

| | |
|---|---|
| `Имя` | Имя записи. |
| `Тип` | Тип записи `CNAME` — задает псевдоним для другого доменного имени. Необходимо указать ровно один хост. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. По умолчанию установлено значение `300s`. |
| `Бэкенды` | Таблица бэкенд-серверов. Необходимо указать ровно один бэкенд:<br>• `Host`: доменное имя, на которое будет указывать CNAME-запись. |

#### Параметры статической записи типа MX

| | |
|---|---|
| `Имя` | Имя записи. |
| `Тип` | Тип записи `MX` — указывает почтовый сервер для домена. Необходимо указать хотя бы один почтовый сервер. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. По умолчанию установлено значение `300s`. |
| `Бэкенды` | Таблица бэкенд-серверов. Для каждого бэкенд-сервера задаются его параметры:<br>• `Host`: доменное имя почтового сервера.<br>• `Приоритет`: приоритет сервера. Чем меньше значение, тем выше приоритет. Необязательный параметр. |

#### Параметры статической записи типа TXT

| | |
|---|---|
| `Имя` | Имя записи. |
| `Тип` | Тип записи `TXT` — содержит произвольный текст. |
| `TTL` | Значение TTL (Time to Live) с указанием единицы времени: `ms` (миллисекунды), `s` (секунды), `m` (минуты) или `h` (часы). Определяет, как долго DNS-запись может храниться в кэше. По умолчанию установлено значение `300s`. |
| `Бэкенды` | Таблица записей бэкенд-серверов. Для каждой записи бэкенд-сервера задаются его параметры:<br>• `Текст`: TXT-запись должна содержать непустой текст длиной не более 4096 символов. |

4. Нажмите `Применить` в нижней части экрана, чтобы применить конфигурацию.

Статус применения конфигурации будет отображаться в верхней части экрана.

### 3.2.2.4 Включение и выключение глобальной балансировки

1. Откройте веб-интерфейс Angie ADC и перейдите в раздел `Управление трафиком` → `Глобальная балансировка`.

   Откроется окно настройки глобальной балансировки.

2. Включите или выключите переключатель `Глобальная балансировка`.

## 3.3 Типовые задачи и примеры

В этом разделе собраны примеры настройки Angie ADC под разные задачи:

- *Настройка IPv6*
- *Настройка пула SNAT*
- *Настройка Transparent Proxy для TCP- и UDP-трафика*
- *Балансировка трафика на основе набора шифров*
- *Загрузка и раздача статических файлов*
- *Делегирование сервиса оператору*
- *Настройка ACME*
- *Настройка пользовательских метрик*
- adc_keepalive-oneconnect
- *Глобальная балансировка с учетом мониторов доступности*

### 3.3.1 Настройка IPv6

IPv6 в Angie ADC настраивается на трех уровнях:

- Настройка доступа к веб-интерфейсу Angie ADC (management plane).
- Настройка сетевых протоколов маршрутизации (control plane).
- Настройки для передачи и обработки клиентского трафика (data plane).

#### 3.3.1.1 Доступ к веб-интерфейсу Angie ADC

Веб-интерфейс Angie ADC одновременно поддерживает IPv4 и с IPv6. Для доступа к веб-интерфейсу с IPv6-адресом `2001:db8::10` введите в браузере:

- `http://[2001:db8::10]:8080`
- `https://[2001:db8::10]:8443` (если требуется безопасное подключение).

Если есть DNS-запись для адреса `2001:db8::10`, можно обращаться к веб-интерфейсу по имени.

#### 3.3.1.2 Настройка сетевых протоколов маршрутизации

Для маршрутизации IPv6 подходят все стандартные и вспомогательные протоколы маршрутизации:

- BGPv4;
- OSPFv3 (для IPv6 только OSPFv3, в отличие от IPv4);
- BFD;
- VRRPv3.

> **Примечание**
>
> Сосед для протокола BGP и пир для BFD указываются в формате IPv6-адреса.

Пример настройки OSPFv3:

```
router ospf6
  ospf6 router-id 2.2.2.2
  log-adjacency-changes
```

```
  redistribute connected
exit
```

Также необходимо включить OSPFv3 на всех интерфейсах:

```
interface ens33
  description internal
  ipv6 address 2001:db8::2/64
  ipv6 ospf6 area 0
exit
```

#### 3.3.1.3 Передача и обработка клиентского трафика

Для поддержки IPv6:

- Бэкенд-серверы должны быть настроены на прием подключений по протоколу IPv6.
- В конфигурации балансировщика нагрузки необходимо включить прием трафика по IPv6:

  ```
  server {
      listen 80;
      listen [::]:80;
      location / {
  ```

  После этого балансировщик сможет принимать подключения по IPv6.
- Также необходимо добавить AAAA-записи в DNS, чтобы клиенты могли находить балансировщик по IPv6-адресу.
- Для маршрутизации трафика к бэкенд-серверам в секции `upstream` должны быть указаны IPv6-адреса.

  В примере ниже один сервер задан адресом IPv4, а второй – IPv6:

  ```
  http {
      upstream myapp1 {
          server 192.168.0.128:80;
          server [2001:db8::4]:80;
      }
  }
  ```

#### 3.3.1.4 Поддержка смешанных подключений IPv4 и IPv6 в Angie ADC

Angie ADC поддерживает гибкую маршрутизацию трафика между клиентами и бэкенд-серверами с разными конфигурациями.

Бэкенд-серверы могут быть трех типов:

- только IPv4;
- только IPv6;
- dual-stack (одновременно поддерживаются оба протокола).

Пример:

К некоторому ресурсу подключаются IPv4-клиенты (отмечены зеленым цветом) и IPv6-клиенты (отмечены красным цветом).

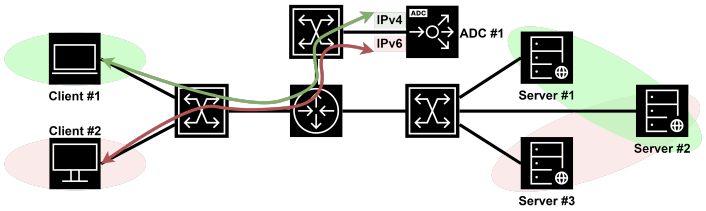



Рис. 1

### Только IPv4

IPv4-сессии установлены между Angie ADC и бэкенд-серверами с поддержкой IPv4:

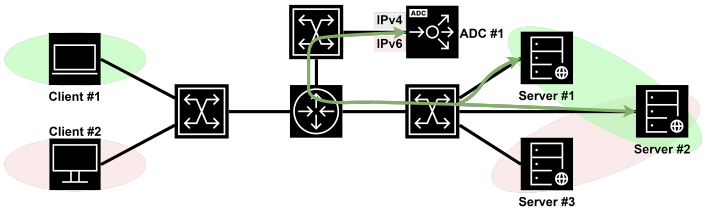



Рис. 2

### Только IPv6

IPv6-сессии установлены между Angie ADC и бэкенд-серверами с поддержкой IPv6:

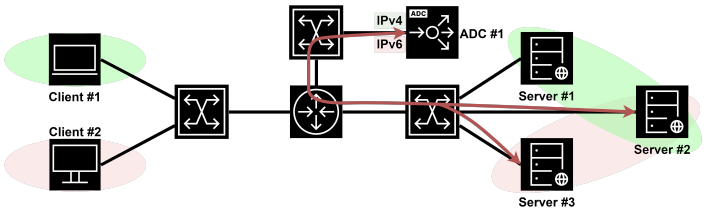



Рис. 3

### Смешанные подключения

Angie ADC может проксировать входящие IPv4-подключения в любые сессии (как IPv4, так и IPv6) в сторону бэкенд-серверов. Аналогично и для IPv6-подключений.

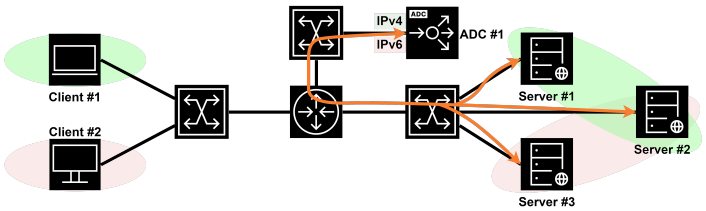



Рис. 4

Таким образом, Angie ADC позволяет гибко проксировать трафик между различными протоколами.

Преимущества такой схемы:

- Практически мгновенная публикация в сети IPv6-сервиса без поддержки IPv6 на бэкенд-серверах.

- Постепенный переход бэкенд-инфраструктуры на IPv6 без потери доступности сервисов снаружи по IPv4, т.е. возможна миграция с IPv4 на IPv6 с минимальными изменениями инфраструктуры.

#### 3.3.1.5 Настройка ip6tables

Правила управления доступом для IPv6 похожи на правила для IPv4, но должны настраиваться отдельно вручную.

Пример типовой настройки `iptables`:

```
[root@angie-va angie-va]# iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2022 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 179 -j ACCEPT
-A INPUT -p ospf -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 520 -j ACCEPT
-A INPUT -p eigrp -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
```

Для IPv6 соответствующая настройка производится с помощью `ip6tables`:

```
[root@angie-va angie-va]# ip6tables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 2022 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 179 -j ACCEPT
-A INPUT -p ipv6-icmp -j ACCEPT
-A INPUT -p ospf -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 521 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp6-adm-prohibited
```

### 3.3.2 Настройка пула SNAT (SNAT Pool)

#### 3.3.2.1 Введение

Пул SNAT (SNAT Pool, Source Network Address Translation Pool) — это набор IP-адресов, которые Angie ADC использует для подмены исходных IP-адресов клиентов при передаче трафика на серверы. SNAT-пулы применяются в высоконагруженных средах с большим количеством одновременно открытых сессий. Они позволяют обойти ограничение на количество одновременных подключений (около 65000 на один IP-адрес). Ограничение связано с количеством доступных клиентских сокетов

(TCP- и UDP-портов), используемых для установки соединений. SNAT-пулы могут использоваться как для L4-, так и для L7-балансировки.

**Примечание**

Также возможен подход, основанный на горизонтальном масштабировании за счет установки нескольких Angie ADC и распределения трафика между ними. Этот подход позволяет решить проблему не только с ограничением количества сессий, но и снять ограничение на производительность системы, например когда при больших объемах трафика необходимо обрабатывать данные на скоростях более 10 Гбит/с.

#### 3.3.2.2 Ручная настройка SNAT-пулов

Настройка вручную создаваемых SNAT-пулов происходит в три этапа:

1. Выбор пула адресов и их настройка.
2. Настройка маршрутизации.
3. Настройка правил балансировки.

Пример простой схемы с одним Angie ADC представлен на рисунке ниже:

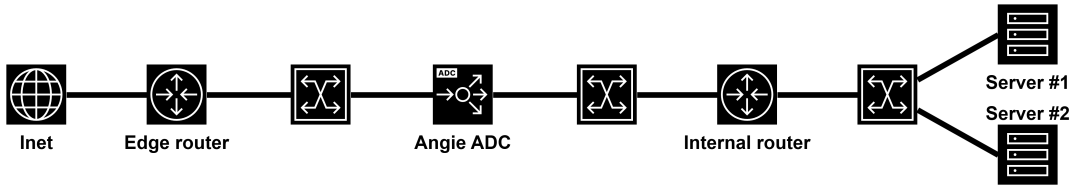



Рис. 2

##### 1. Выбор пула адресов и их настройка

Размер пула адресов напрямую зависит от количества сессий, которые требуется одновременно поддерживать. Чем больше сессий должно работать одновременно, тем больше IP-адресов должно входить в пул.

Выбор адресов условно свободный, можно использовать адреса из разных подсетей. Мы не рекомендуем использовать чужие валидные адреса, чтобы избежать случайных пересечений с адресами клиентов. Также лучше выбирать адреса не из произвольного диапазона, а выделить конкретную подсеть, которая будет полностью использоваться в пуле. Это упростит дальнейшую настройку маршрутизации.

В примере ниже мы будем использовать подсеть `192.168.12.20/30`, включающую в себя четыре адреса. Так как все четыре адреса будут использоваться в качестве адресов отправителя пакетов на участке сети между Angie ADC и апстримами, то необходимо все эти четыре адреса назначить на сетевой интерфейс системы, например на loopback-интерфейс.

Пример конфигурации представлен ниже. Адреса назначаются с маской `/32`. Это нужно, чтобы система обрабатывала возвращающийся от серверов трафик и устанавливала полноценные соединения.

```
angie-va#
angie-va# conf t
angie-va(config)# int lo
angie-va(config-if)# ip add 192.168.12.20/32
angie-va(config-if)# ip add 192.168.12.21/32
angie-va(config-if)# ip add 192.168.12.22/32
angie-va(config-if)# ip add 192.168.12.23/32
```

**Примечание**

В этом примере между апстримами и Angie ADC размещен маршрутизатор. Ситуация, когда апстримы и один из интерфейсов Angie ADC размещены в одной подсети, будет рассмотрена отдельно (см. *Заключение* ниже).

### 2. Настройка маршрутизации

В этом примере мы будем использовать протокол BGP для динамической маршрутизации. Настройка других протоколов динамической маршрутизации выполняется аналогично.

В BGP можно анонсировать все четыре адреса из пула независимо (четыре префикса с маской /32). Однако BGP позволяет производить суммаризацию (агрегирование) анонсируемых префиксов, поэтому есть способ лучше — выполнить анонс всего пула целиком в виде одного префикса с маской /30.

Создадим агрегированный маршрут для всей подсети, используемой в качестве пула SNAT:

```
angie-va# conf t
angie-va(config)# ip route 192.168.12.20/30 Null0
angie-va(config)# exi
angie-va# sho ip ro sta
Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, F - PBR,
f - OpenFabric, t - Table-Direct,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure
S>* 192.168.12.20/30 [1/0] unreachable (blackhole), weight 1, 00:01:21
```

Получившийся маршрут необходимо анонсировать по протоколу BGP:

```
angie-va# conf t
angie-va(config)# router bg 1
angie-va(config-router)# add ipv4 un
angie-va(config-router-af)# red sta
angie-va(config-router-af)# end
angie-va#
```

Настройку BGP-соседей опустим для краткости (подробнее о настройке BGP см. *BGP-маршрутизация*).

**Примечание**

На представленной выше схеме BGP-соседом должен стать маршрутизатор Internal router, так как трафик от серверов будет возвращаться через него.

Убедимся, что BGP-соседи получают корректный маршрут:

```
internal_router#sho ip bg
BGP table version is 6, local router ID is 192.168.0.150
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
```

```
*>i 192.168.12.20/30 192.168.0.147 0 100 0 ?
internal_router#
```

### 3. Настройка правил балансировки

Для распределения клиентов по адресам из пула можно использовать различные модули, например Geo, Geo IP, Map, Split Clients. Рассмотрим несколько типовых сценариев с описанием используемых модулей.

#### Сценарий A: использование модуля Split Clients

Стандартная ситуация, когда на Angie ADC попадают подключения пользователей из интернета напрямую. Предполагается, что пользователи распределяются равномерно по всему пространству адресов. В этом случае для настройки можно использовать модуль Split Clients (применяется в контексте `http`). `Split_clients` работает как при L7-, так и при L4-балансировке.

Пример конфигурации:

```
split_clients "${remote_addr}AAA" $variant {
    25% 192.168.12.20;
    25% 192.168.12.21;
    25% 192.168.12.22;
    25% 192.168.12.23;
}
```

**Примечание**

К адресу клиента добавлен текст `AAA`. Это нужно для изменения значения, выдаваемого хэш-функцией. Если потребуется поменять распределение клиентов по адресам из пула, можно поменять добавляемый текст.

#### Сценарий B: использование модуля Map

Возможна ситуация, когда перед Angie ADC с точки зрения трафика, идущего от клиентов, расположен другой балансировщик:

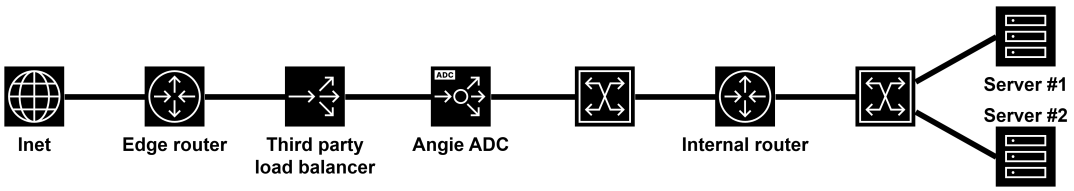



Рис. 3

Если этот балансировщик использует SNAT-пул, то переменная `remote_addr` может не обеспечить достаточного разнообразия. Это может привести к коллизиям — перекосу в распределении клиентов по адресам из пула SNAT. Решить указанную проблему можно, например, используя модуль Map, с помощью которого можно в явном виде задать соответствие «старого» и «нового» адресов отправителя. Метод `map` работает как при L7-, так и при L4-балансировке.

Пример конфигурации (восемь адресов из предыдущего пула привязываются к четырем адресам пула SNAT):

```
map $remote_addr $variant {
    "10.10.10.0" "192.168.12.20";
```

```
    "10.10.10.1" "192.168.12.20";
    "10.10.10.2" "192.168.12.21";
    "10.10.10.3" "192.168.12.21";
    "10.10.10.4" "192.168.12.22";
    "10.10.10.5" "192.168.12.22";
    "10.10.10.6" "192.168.12.23";
    "10.10.10.7" "192.168.12.23";
}
```

#### Сценарий C: модуль Split Clients и хэш переменной $request_id

Для балансировки HTTP-трафика можно использовать подход, при котором не важно, используется ли какой-либо NAT/PAT или прокси между реальным клиентом и Angie ADC. Для распределения клиентов будет использоваться модуль Split Clients и хэш от переменной `$request_id`. Значение переменной генерируется автоматически и, по сути, является случайным числом. Этот метод имеет ограниченную область применения — L7-балансировка для протокола HTTP.

Пример конфигурации:

```
split_clients "${request_id}" $variant {
    25% 192.168.12.20;
    25% 192.168.12.21;
    25% 192.168.12.22;
    25% 192.168.12.23;
 }
```

**Примечание**

При использовании этого метода разные сессии одного и того же клиента могут привязываться к различным адресам из пула. Это не должно быть большой проблемой, так как современные веб-системы используют заголовок `X-Forwarded-For` для определения адреса клиента и не смотрят на адрес отправителя в IP-пакете.

Теперь с помощью директивы proxy_bind укажем, где и как использовать указанное распределение клиентов. Директиву можно задать в контекстах `http`, `server` или `location`.

Пример:

```
location / {
proxy_pass http://backend ;
proxy_bind ${variant};
}
```

### 3.3.2.3 Заключение

Предложенные выше способы ручного создания SNAT-пула позволят уверенно перешагнуть ограничение в 65000 одновременных сессий как для L4-, так и для L7-балансировки. Однако следует все же понимать, что ограничение количества одновременно поддерживаемых сессий зависит не только от количества сокетов, но и от объемов использования целого ряда других системных ресурсов.

Чтобы адаптировать используемый подход к ситуации, когда один из интерфейсов Angie ADC расположен в одной подсети с апстрим-серверами, потребуется следующее:

- не использовать динамическую маршрутизацию;
- настроить адреса из пула на физическом интерфейсе, а не на loopback-интерфейсе (адреса должны быть из той же IP-подсети, в которой расположены серверы);

- для обеспечения отказоустойчивости можно использовать несколько Angie ADC, а IP-адреса из пула указать как виртуальные дополнительные адреса для VRRP, вместо того чтобы назначать их непосредственно на физический интерфейс.

### 3.3.3 Настройка Transparent Proxy для TCP- и UDP-трафика

Angie ADC работает как обратный прокси для TCP- и UDP-трафика. Апстрим-серверы видят, что весь трафик приходит с IP-адреса самого прокси Angie ADC:

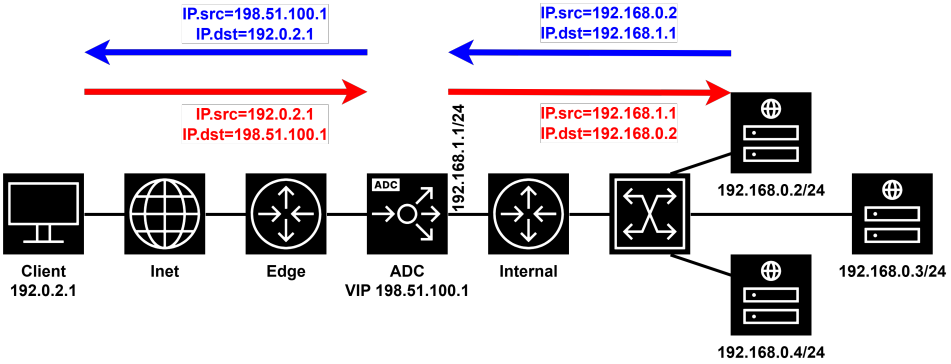



Рис. 1: красным цветом обозначен поток трафика от клиента к сервису, синим — обратный трафик.

Для определения и передачи настоящего IP-адреса клиента апстрим-серверу доступны следующие два метода:

- *IP Transparency* — апстрим-серверы видят соединение как исходящее от реального клиента (для TCP- и UDP-трафика).
- *Direct Server Return (DSR)* — дополнительно позволяет отправлять ответы от апстрим-серверов напрямую клиентам, минуя промежуточный балансировщик (только для UDP-трафика). Метод может быть реализован через NAT.

#### 3.3.3.1 IP Transparency

При использовании IP Transparency апстрим-серверы будут видеть настоящий IP-адрес клиента, с которого приходят пакеты. IP Transparency может использоваться как с TCP-, так и с UDP-протоколами.

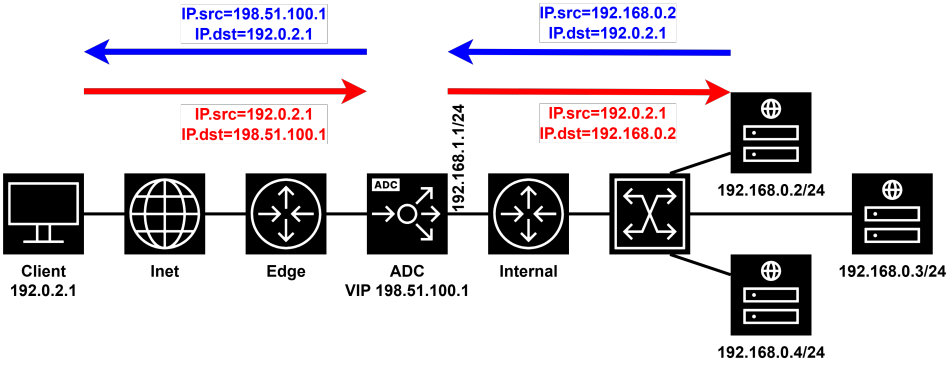



Рис. 2: пример сети, в которой на системе балансировки Angie ADC включена опция IP Transparency.

##### Предусловия

Должно выполняться одно из следующих условий:

- Балансировщик и апстрим-серверы находятся в одной подсети, балансировщик является маршрутизатором по умолчанию (default gateway) для апстрим-серверов.

- Балансировщик находится на пути передачи трафика между клиентом и апстрим-серверами (трафик проходит через балансировщик в обоих направлениях).

### Пример настройки балансировщика

```
# in the 'http' context
upstream http_upstreams {
     server 192.168.0.2;
     server 192.168.0.3;
     server 192.168.0.4;
}

server {
    listen 80;

    location / {
        proxy_bind $remote_addr transparent;
        proxy_pass http://http_upstreams;
    }
}
```

В этом примере:

- Настроен простой виртуальный HTTP-сервер, распределяющий трафик между группой апстрим-серверов.
- Исходный адрес для трафика, направляемого на апстрим-серверы, подменяется с помощью параметра `transparent` в директиве adc__proxy_bind. Чаще всего в качестве исходного адреса указывается адрес удаленного клиента.

### Настройка маршрутизации

При использовании IP Transparency трафик должен проходить через Angie ADC и обрабатываться им. Для этого необходимо настроить корректную маршрутизацию.

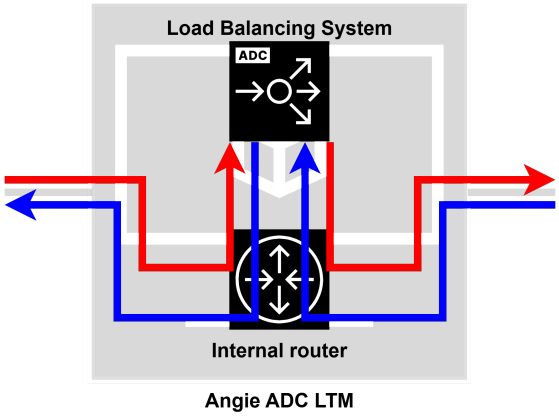



Рис. 3: движение трафика через Angie ADC с настроенным IP Transparency.

Маршрутизация настраивается на основе политик PBR (Policy-Based Routing):

1. Подключитесь к интерфейсу командной строки на порту 2222.
2. Добавьте маршрутизацию всего трафика в локальный интерфейс `lo` в таблице `100`:

   ```
   ip route add local 0.0.0.0/0 dev lo table 100
   ```

3. Добавьте правила: весь трафик с апстрим-серверов необходимо обрабатывать с помощью таблицы маршрутизации `100`:

```
ip rule add from 192.168.0.2 table 100
ip rule add from 192.168.0.3 table 100
ip rule add from 192.168.0.4 table 100
```

Вместо добавления отдельных серверов можно добавить правило для подсети, если это необходимо (например, `ip rule add from 10.21.20.0/24 table 100`).

4. Проверьте, что весь трафик обрабатывается в таблице `100` и что есть только один маршрут — весь трафик уходит в локальный интерфейс `lo`:

```
# ip rule
0: from all lookup local
32764: from 192.168.0.2 lookup 100
32765: from 192.168.0.3 lookup 100
32766: from 192.168.0.4 lookup 100
32767: from all lookup main
32768: from all lookup default

# ip route show table 100
local default dev lo scope host
```

#### 3.3.3.2 Direct Server Return (DSR)

**Примечание**

Применение DSR возможно только для UDP-трафика.

При использовании Direct Server Return (DSR) апстрим-серверы будут видеть настоящий IP-адрес клиента, с которого приходят пакеты, и будут отправлять ответ непосредственно удаленному клиенту (т. е. пакеты с ответом полностью обходят балансировщик нагрузки). DSR может дать небольшой прирост производительности:

- Позволяет не удерживать открытыми UDP-сокеты в ожидании пакета ответа (повышает масштабируемость).
- Пакеты с ответами могут полностью обходить L7-обработку (снижает задержку).

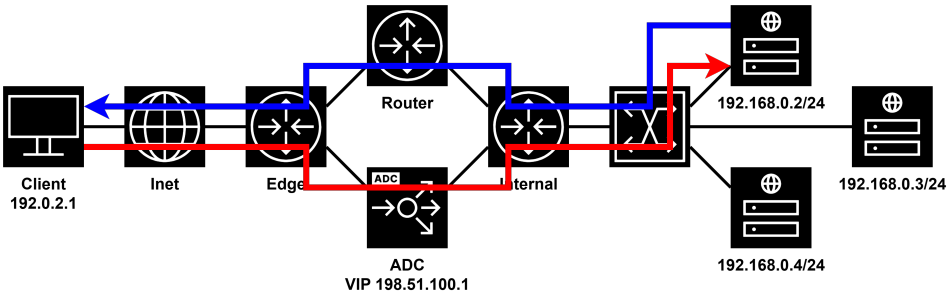



Рис. 4: движение трафика через Angie ADC с DSR.

Ограничения при использовании DSR:

- Балансировщик нагрузки не видит пакетов с ответами, поэтому не может определить, отвечает ли апстрим-сервер или вышел из строя.
- Балансировщик не может анализировать запрос глубже первого пакета, чтобы выбрать апстрим-сервер, поэтому его возможности по принятию решений о балансировке (например, маршрутизация на основе содержимого) ограничены.
- Балансировщик не может участвовать в процессах согласования или обработке состояний, таких как SSL/TLS.

- Такие функции, как кэширование и логирование, недоступны при использовании DSR.

Кроме того при использовании DSR необходимо:

- переписывать исходный адрес ответа, чтобы он соответствовал публичному адресу балансировщика нагрузки;
- настроить активные проверки для апстрим-серверов.

### Пример настройки балансировщика

```
# in the 'stream' context
server {
    listen 53 udp;

    proxy_bind $remote_addr:$remote_port transparent;
    proxy_responses 0;
    #proxy_timeout 1s;
}

upstream dns_upstreams {
    server 192.168.0.2:53;
    server 192.168.0.3:53;
    server 192.168.0.4:53;
}
```

В этом примере:

- Создан кластер из трех DNS-серверов с балансировкой нагрузки. Каждый из DNS-серверов должен возвращать разные IP-адреса при запросах `www.example.com`.
- Настроена простая конфигурация обратного прокси, которая распределяет нагрузку между этими DNS-серверами.
- Параметр `transparent` в директиве `proxy_bind` позволяет апстрим-серверу видеть исходный IP-адрес и порт (переменные `$remote_addr` и `$remote_port`).
- Балансировщик не ожидает ответ от апстрим-серверов (директива `proxy_responses=0`). Ответ будет отправлен клиенту напрямую.

### Переписывание IP-адреса и порта при ответе

Апстрим-сервер генерирует ответные пакеты, в которых в качестве исходного адреса используется его локальный IP-адрес. Этот исходный адрес необходимо переписать на IP-адрес и порт балансировщика нагрузки, к которому изначально подключался клиент. Есть два способа для переписывания исходящих пакетов:

- Router NAT – переписывание на промежуточном маршрутизаторе.
- Origin NAT – переписывание при отправке пакетов с каждого апстрим-DNS-сервера.

### Router NAT

Вы можете изменять (переписывать) исходящие пакеты на промежуточном маршрутизаторе, который является маршрутом по умолчанию для апстрим-серверов.

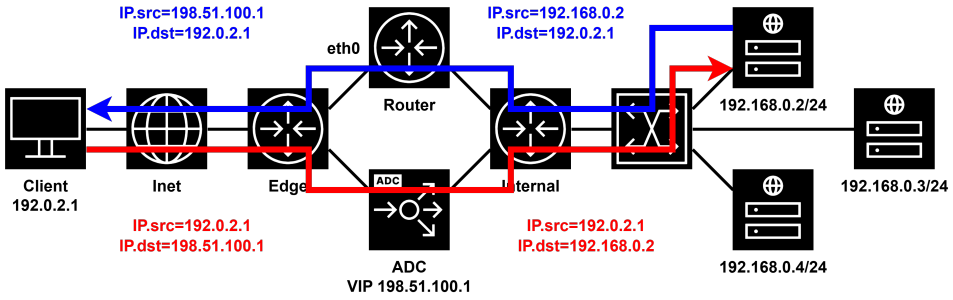



Рис. 5: движение трафика с Router NAT

Для этого настройте на маршрутизаторе stateless NAT-переписывание:

Например:

```
# tc qdisc add dev eth0 root handle 10: htb
# tc filter add dev eth0 parent 10: protocol ip prio 10 u32 match ip src 192.168.0.2␣
↪match ip sport 53 action nat egress 192.168.0.2 198.51.100.1
# tc filter add dev eth0 parent 10: protocol ip prio 10 u32 match ip src 192.168.0.3␣
↪match ip sport 53 action nat egress 192.168.0.3 198.51.100.1
# tc filter add dev eth0 parent 10: protocol ip prio 10 u32 match ip src 192.168.0.4␣
↪match ip sport 53 action nat egress 192.168.0.4 198.51.100.1
```

Убедитесь, что вы указали корректный исходящий интерфейс (например, `eth0`) и правильные IP-адреса для каждого апстрим-сервера. В зависимости от конфигурации можно сократить количество команд `tc filter`, объединив их с помощью CIDR-масок для параметров `src` и `egress old`.

Чтобы вывести текущую конфигурацию `tc filter`, выполните:

```
# tc filter show dev eth0
```

### Origin NAT

Вариант с переписыванием исходящих пакетов при их отправке с каждого апстрим-DNS-сервера применим, если у вас есть возможность настраивать сеть на апстрим-серверах, особенно если они напрямую подключены к интернету.

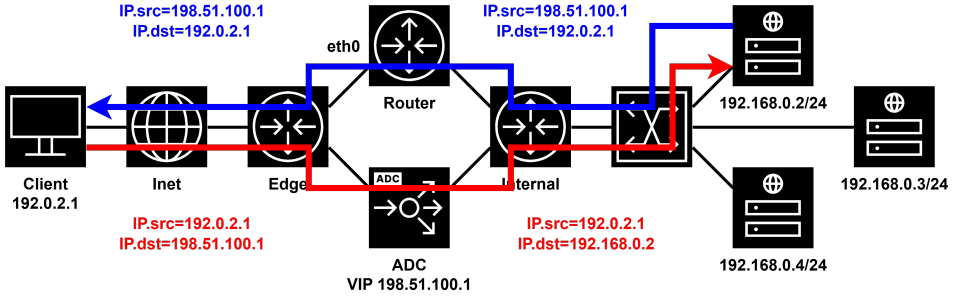



Рис. 6: движение трафика с Origin NAT

Необходимо настроить stateless NAT-переписывание на каждом апстрим-сервере.

Пример:

```
# tc qdisc add dev eth0 root handle 10: htb
# tc filter add dev eth0 parent 10: protocol ip prio 10 u32 match ip src 192.168.0.2␣
↪match ip sport 53 action nat egress 192.168.0.2 198.51.100.1
```

Такую конфигурацию необходимо применить на каждом апстрим-сервере, убедившись, что выбран корректный интерфейс и корректные IP-адреса.

### 3.3.4 Балансировка трафика на основе набора шифров

В этой статье рассматривается балансировка трафика на основе набора шифров. На один VIP-адрес балансировщика нагрузки поступает два типа трафика:

- TLS-трафик с шифрованием ГОСТ;
- TLS-трафик без ГОСТ.

Для стандартного TLS выполняется Offload и балансировка на Angie ADC. Трафик ГОСТ направляется на отдельный бэкенд-сервер без расшифровки.

#### 3.3.4.1 Настройка

Конфигурация основного бэкенд-сервера:

```
http {
    default_type text/plain;

    server {
        listen 80;
        server_name localhost;

        location / {
            return 200 "Hello, this is default backend\n";
        }
    }
}
```

Конфигурация бэкенд-сервера для ГОСТ-трафика:

```
http {
    default_type text/plain;

    server {
        listen 443 ssl;
        server_name localhost;
        ssl_certificate /etc/angie-lb/crt/server.crt;
        ssl_certificate_key /etc/angie-lb/crt/server.key;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers IANA-GOST2012-GOST8912-GOST8912:GOST2012-MAGMA-MAGMAOMAC:GOST2012-
↪KUZNYECHIK-KUZNYECHIKOMAC;
        ssl_prefer_server_ciphers on;

        location / {
            return 200 "Hello, this is GOST-based server\n";
        }
    }
}
```

#### Случай 1

Трафик балансируется на ГОСТ-бэкенд, если ГОСТ присутствует в списке шифров (может присутствовать наряду с другими шифрами).

Конфигурация:

```
http {

    upstream backend {
        server IP.OF.MAIN.BACKEND:80;
```

```
    }

    server {
        listen       127.0.0.1:8443 ssl;
        ssl_certificate /etc/angie-lb/crt/cp.crt;
        ssl_certificate_key /etc/angie-lb/crt/cp.key;
        server_name  localhost;
        status_zone  default;
        location / {
            proxy_pass http://backend;
        }
    }

}

stream {

    map $ssl_preread_ciphers $handler {
        "~TLS_GOST" 127.0.0.1:9443;
        default     127.0.0.1:8443;
    }

    server {
        listen       443;
        ssl_preread  on;
        pass $handler;
    }

    server {
        listen       127.0.0.1:9443;
        proxy_pass   IP.OF.GOST.BACKEND:443;
    }
}
```

### Случай 2

Трафик балансируется на ГОСТ-бэкенд, если заданы только ГОСТ-шифры. Если помимо ГОСТ-шифров есть другие шифры, трафик отправляется на основной бэкенд.

Конфигурация:

```
http {

    upstream backend {
        server IP.OF.MAIN.BACKEND:80;
    }

    server {
        listen       127.0.0.1:8443 ssl;
        ssl_certificate /etc/angie-lb/crt/cp.crt;
        ssl_certificate_key /etc/angie-lb/crt/cp.key;
        server_name  localhost;
        status_zone  default;
        location / {
            proxy_pass http://backend;
        }
    }
```

```
}

stream {

    map $ssl_preread_ciphers $backend {
        "~^(TLS_GOST[^:]*)(:TLS_GOST[^:]*)*(:TLS_EMPTY_RENEGOTIATION_INFO_SCSV)?$" ␣
↪127.0.0.1:9443;
        default 127.0.0.1:8443;
    }

    server {
        listen       443;
        ssl_preread  on;
        pass $backend;
    }

    server {
        listen       127.0.0.1:9443;
        proxy_pass   IP.OF.GOST.BACKEND:443;
    }
}
```

#### 3.3.4.2 Проверка

##### Случай 1

Трафик без ГОСТ отправляется на основной бэкенд:

```
# curl --ciphers '!IANA-GOST2012-GOST8912-GOST8912:!GOST2012-MAGMA-MAGMAOMAC:!
↪GOST2012-KUZNYECHIK-KUZNYECHIKOMAC:HIGH' --tlsv1.2 --tls-max 1.2 https://example.com
Hello, this is default backend
```

Трафик с ГОСТ отправляется на ГОСТ-бэкенд:

```
# curl --ciphers 'ALL' --tlsv1.2 --tls-max 1.2 https://example.com
Hello, this is GOST-based server
```

##### Случай 2

Трафик без ГОСТ отправляется на основной бэкенд:

```
# curl --ciphers '!IANA-GOST2012-GOST8912-GOST8912:!GOST2012-MAGMA-MAGMAOMAC:!
↪GOST2012-KUZNYECHIK-KUZNYECHIKOMAC:HIGH' --tlsv1.2 --tls-max 1.2 https://example.com
Hello, this is default backend
```

Трафик с ГОСТ отправляется на основной бэкенд:

```
# curl --ciphers 'ALL' --tlsv1.2 --tls-max 1.2 https://example.com
Hello, this is default backend
```

Трафик, где только ГОСТ, отправляется на ГОСТ-бэкенд:

```
# curl --ciphers 'IANA-GOST2012-GOST8912-GOST8912:GOST2012-MAGMA-MAGMAOMAC:GOST2012-
↪KUZNYECHIK-KUZNYECHIKOMAC' --tlsv1.2 --tls-max 1.2 https://example.com
Hello, this is GOST-based server
```

### 3.3.5 Загрузка и раздача статических файлов

Angie ADC поддерживает загрузку и раздачу статических файлов (изображений, статических HTML-страниц и пр.). Загрузка осуществляется по протоколу SCP (Secure Copy Protocol), после чего файл будет доступен по HTTP.

Чтобы добавить статический файл, выполните следующие действия:

1. На машине-источнике запустите загрузку файла (например страницу-заглушку `maintenance.html`):

   ```
   scp -P 2222 maintenance.html admin@<имя_хоста>:/maintenance.html
   ```

2. В веб-интерфейсе Angie ADC перейдите к редактированию конфигурации балансировщика нагрузки (`Управление трафиком` → `Балансировщик нагрузки`) и в блоке `http` настройте доступ к каталогу с этим файлом:

   ```
   http {
       server {
           listen 80;
           root /var/transfer;
       }
   }
   ```

3. Если для расширения файла не определен MIME-тип или требуется указать его явно, добавьте соответствующую строку в блок `types` (например `application/pkix-crl crl;` для CRL).

4. Сохраните изменения, нажав кнопку `Сохранить`.

   Файл `maintenance.html` будет доступен по адресу `http://<имя_хоста>/maintenance.html`.

Чтобы проверить, что файл корректно отдается клиентам, выполните запрос `curl`. Например, для хоста `adc.angie.local`:

```
$ curl adc.angie.local/maintenance.html -v
```

Если получен код `200 OK` и указан правильный Content-Type, то файл был успешно загружен и доступен.

#### 3.3.5.1 Раздача файлов из разных локальных каталогов

В зависимости от типа файлы могут раздаваться из разных каталогов, например HTML-страницы из `/static/www`, а изображения из `/static/images`. Для настройки используется директива location.

Чтобы настроить раздачу файлов из разных локальных каталогов, выполните следующие действия:

1. Создайте локальный каталог со всеми необходимыми подкаталогами.

   Например:

   ```
   mkdir -p static/www static/images
   ```

2. Перенесите в этот каталог статические файлы. Например, поместите файл `index.html` в `/static/www`, а изображение `image1.png` в `/static/images`.

3. Скопируйте весь каталог в файловое хранилище Angie ADC по SCP:

   ```
   scp -P 2222 -r static admin@<имя_хоста>:/
   ```

4. В веб-интерфейсе Angie ADC перейдите к редактированию конфигурации балансировщика нагрузки (`Управление трафиком` → `Балансировщик нагрузки`) и в блоке `http` настройте доступ к каталогам со статическими файлами:

```
server {
    location / {
        root /static/www;
    }

    location /images/ {
        root /static;
    }
}
```

5. Сохраните изменения, нажав кнопку `Сохранить`.

Теперь запросы, URI которых начинаются на `/images/`, будут обрабатываться в `/static/images`. Другие запросы будут обрабатываться в `/static/www`. Если файл не найден, отобразится ошибка `404`.

Например, на запрос `http://<имя_хоста>/images/image1.png` будет отдан файл `/static/images/image1.png`, а на запрос `http://<имя_хоста>/files/index.html` будет отдан файл `/static/www/files/index.html`.

### 3.3.6 Keepalive как аналог OneConnect profile от F5

Для повторного использования уже открытых TCP-соединений с серверами бэкенда используйте механизм keepalive.

Особенности:

- Если в upstream-группе определено несколько серверов, то keepalive-соединения устанавливаются с каждым из них в соответствии с выбранным методом балансировки.
- Мастер-процесс запускает несколько рабочих процессов в зависимости от настройки директивы worker_processes и у каждого рабочего процесса будет собственный пул keepalive-соединений:
  - Если все запросы обрабатывает один рабочий процесс, то они могут проксироваться в одно и то же установленное соединение.
  - Если запросы распределяются между несколькими рабочими процессами, то для каждого процесса формируются свои keepalive-соединения с бэкендом.

#### 3.3.6.1 Настройка

1. Задайте директиву keepalive в блоке `upstream`.

   Пример:

```
upstream http_backend {
    server 192.168.0.1;
    server 192.168.0.2;

    keepalive 32;
}
```

2. Для HTTP установите директиву proxy_http_version в значение `1.1`, а поле заголовка `Connection` в adc__proxy_set_header очистите:

   Пример:

```
server {
    # ...

    location / {
        proxy_pass http://http_backend;
```

```
        proxy_http_version 1.1;
        proxy_set_header Connection "";
        # ...
    }
}
```

3. Если требуется разделять потоки по IP-адресам, используйте привязку по sticky learn в группе upstream в сочетании с директивой geo.

   Пример:

```
geo $idmask {
    ranges;
    default                   ZZ;
    10.21.0.0-10.21.0.255     tau;
    10.21.4.0-10.21.4.255     chi;
    10.21.8.0-10.21.8.255     alpha;
    10.21.20.0-10.21.20.255   nu;
}

upstream http_backend {
    server 10.21.8.6:80;
    server 10.21.8.13:80;

    sticky learn
        create=$idmask
        lookup=$idmask
        zone=client_sessions:1m
        timeout=60s;

    keepalive 32;
}

server {
    listen       10.21.8.8:80;
    server_name  localhost;

    location / {
        proxy_http_version 1.1;
        proxy_set_header Connection "";

        proxy_pass http://http_backend;
    }
}
```

   В этом случае в пределах одного рабочего процесса поведение Angie ADC будет очень похоже на механизм OneConnect в F5 BIG-IP.

### 3.3.7 Делегирование сервиса оператору

Angie ADC поддерживает передачу управления выбранными сервисами оператору. Для этого администратор создает отдельные файлы конфигурации, в которых будет настраиваться балансировка сервиса, и выдает права на доступ к этим файлам оператору. Администратор также подключает эти файлы в основной конфигурационный файл `angie.conf` с помощью директивы `include`. Оператор, занимающийся обслуживанием сервиса, редактирует конфигурацию балансировщика нагрузки в выделенных для него файлах. При этом он не видит другие файлы конфигурации, т.к. не имеет к ним доступа. При сохранении любого файла, включенного по `include`, вся конфигурация балансировщика валидируется и применяется. Также можно настроить доступ оператора к

данным мониторинга сервиса.

**Важно**

Включаемые файлы должны содержать синтаксически верные директивы и блоки, иначе итоговая конфигурация не пройдет валидацию и не будет применена.

#### 3.3.7.1 Настройка со стороны администратора

Ниже приведен пример делегирования оператору управления сервисом `example.com`.

1. В веб-интерфейсе Angie ADC перейдите в раздел `Управление трафиком` → `Балансировщик нагрузки`.
2. Создайте файлы `example_upstream.conf` и `example_server.conf` в папке `http.d`.
3. В конфигурации `angie.conf` создайте блок, управляющий сервисом `example.com`, например:

```
http {

    upstream example_upstream {
        zone example_zone 1m;
        include "http.d/example_upstream.conf";
    }

    server {
        listen       VIP:80;
        server_name  example.com;
        status_zone  example_zone;

        location / {
            proxy_pass "http://example_upstream";
            include "http.d/example_server.conf";
        }
    }

}
```

   Здесь:

   - Создана группа серверов `example_upstream` для балансировки нагрузки.
   - Выделена зона разделяемой памяти `example_zone`. В ней хранится текущее состояние серверов, счетчики запросов и сессии.
   - С помощью директивы adc__include подключен внешний файл `example_upstream.conf`, в котором оператор сможет указать IP-адреса или домены бэкенд-серверов.
   - Описаны правила обработки запросов.
   - Метрики для вывода статистики (adc__status_zone) будут собираться в уже заданную зону разделяемой памяти `example_zone`.
   - Все входящие запросы будут перенаправляться на бэкенд-серверы, описанные в файле `example_upstream.conf`.
   - С помощью директивы adc__include подключен файл `example_server.conf` с дополнительными настройками для проксирования, которые также может настраивать оператор.

4. Выдайте оператору *права* на редактирование файлов `http.d/example_upstream.conf` и `http.d/example_server.conf`.

> **Совет**
>
> Рекомендуется также передать оператору внутренние инструкции и стандарты компании по наполнению этих конфигурационных файлов.

5. Выдайте оператору *права доступа* к зоне разделяемой памяти `example_zone` для просмотра статистики сервиса на странице мониторинга.

Теперь в разделе **Управление трафиком** → **Балансировщик нагрузки** оператору доступны файлы `http.d/example_upstream.conf` и `http.d/example_server.conf`.

Изменения, внесенные оператором в файл, автоматически валидируются системой и применяются после нажатия кнопки **Сохранить** без прерывания трафика. Если конфигурация не проходит валидацию, то она не применяется.

#### 3.3.7.2 Пример работы оператора

1. В веб-интерфейсе Angie ADC перейдите в раздел **Управление трафиком** → **Балансировщик нагрузки**.

   Откроется экран с файлами, доступ к которым у вас есть. Вы можете просматривать и редактировать эти файлы.

   > **Важно**
   >
   > Отредактированные файлы должны содержать синтаксически верные директивы и блоки, иначе итоговая конфигурация балансировщика не пройдет валидацию и не будет применена.

2. В файле `example_upstream.conf` укажите адреса бэкендов и алгоритм балансировки. Например:

   ```
   least_conn;
   server 192.0.2.10:8080;
   server 192.0.2.11:8080;
   ```

3. Нажмите **Сохранить**. Изменения в конфигурации будут применены сразу.

4. Если необходимо, в файле `example_server.conf` добавьте дополнительные настройки для проксирования, например заголовок запроса:

   ```
   proxy_set_header X-Custom-Header "value";
   ```

5. Нажмите **Сохранить**. Изменения в конфигурации будут применены сразу.

### 3.3.8 Настройка ACME

Модуль ACME в Angie обеспечивает автоматическое получение сертификатов с использованием протокола ACME. Протокол ACME предусматривает несколько способов проверки доменов (также используется термин *верификация*); модуль реализует HTTP-проверку, DNS-проверку, ALPN-проверку, а также проверку с помощью хуков через самостоятельно реализуемый внешний сервис.

#### 3.3.8.1 Шаги настройки

Общие шаги для включения запроса сертификатов в конфигурации:

- **Настройте ACME-клиент** в блоке `http` с помощью директивы adc__acme_client, задающей уникальное имя клиента и другие параметры; можно настроить несколько клиентов ACME.

- **Укажите домены, для которых запрашиваются сертификаты**: для доменных имен, перечисленных во всех директивах adc__server_name всех блоков `server` с директивами adc__acme, указывающими на один и тот же ACME-клиент, будет выдан единый сертификат.

  IP-адреса (IPv4 или IPv6), указанные в adc__server_name, также являются допустимыми идентификаторами сертификата, за исключением DNS-проверки.

  Обратите внимание: публичные центры сертификации могут выдавать сертификаты для IP-адресов только в рамках определенных профилей сертификатов; например, Let's Encrypt в настоящее время требует профиль, допускающий IP-идентификаторы (см. документацию по профилям). Чтобы запросить такой профиль, задайте параметр `profile=` директивы adc__acme_client.

- **Настройте обработку запросов и вызовов ACME**: это нужно для проверки владения доменом. Способ настройки зависит от способа проверки доменных имен:

| Способ | Требования к пользователю | Мультидомены | Домены со звездочкой |
|---|---|---|---|
| *HTTP-проверка* | Открыть порт 80 (или указанный в adc__acme_http_port) для входящих соединений на сервере Angie. | ✓ | |
| *DNS-проверка* | Открыть порт 53 (или указанный в adc__acme_dns_port) для входящих соединений на сервере Angie. Настроить NS-запись для поддомена `_acme-challenge.`, направив ее на свой сервер Angie. | ✓ | ✓ |
| *ALPN-проверка* | Открыть порт 443 (или TLS-порт, используемый сервером Angie) для входящих соединений. | ✓ | |
| Проверка хуками | Реализовать внешний сервис (скрипт или приложение), который по команде Angie сможет внести изменения в DNS-зону или разместить специальный ответ на веб-сервере. | ✓ | ✓ |

- **Настройте SSL с использованием полученного сертификата и ключа**: Модуль делает сертификаты и ключи доступными в виде встроенных переменных, которые можно использовать в конфигурации для заполнения adc__ssl_certificate и adc__ssl_certificate_key.

  Инструкции по настройке SSL см. в разделе Настройка SSL.

> **Совет**
>
> Процесс получения и обновления сертификатов зависит от работы многих служб и может занимать какое-то время. Запаситесь терпением, а в случае возникновения проблем или сомнений обратитесь к отладочному логу.

#### 3.3.8.2 Подробности реализации

Здесь ключи и сертификаты клиентов хранятся в кодировке PEM в соответствующих подкаталогах каталога, заданного с помощью параметра сборки `--http-acme-client-path`:

```
$ ls /var/lib/angie/acme/example/

  account.key  certificate.pem  private.key
```

**Примечание**

Эти файлы сохраняются на диске между перезапусками. При запуске клиент повторно использует сохраненный сертификат, если тот еще действителен, вместо запроса нового, что устраняет задержку на выпуск и лишние обращения к серверу CA, на которые могут действовать ограничения частоты запросов. В контейнере размещайте каталог хранения (по умолчанию `/var/lib/angie/acme/`) на постоянном томе, чтобы выданные сертификаты сохранялись при пересоздании контейнера; см. запуск Angie в контейнере.

**Примечание**

Перезагрузка конфигурации (`angie -s reload`) заново сверяет каждый клиент ACME с сертификатами, сохраненными на диске. Клиент, сертификат которого в данный момент недействителен (в том числе если его предыдущий запрос завершился ошибкой), запрашивается заново немедленно, независимо от retry_after_error. Задержка перед повторной попыткой (как и значение `retry_after_error=off`) действует, только пока Angie работает, и не сохраняется после перезагрузки. Поэтому повторные перезагрузки при постоянно неудачных запросах могут приводить к частым обращениям к серверу CA, на которые действуют упомянутые выше ограничения частоты запросов.

Клиенту ACME требуется учетная запись на сервере CA. Для ее создания и управления ею клиент использует закрытый ключ (`account.key`); если ключа у него еще нет, ключ создается при запуске. Затем клиент использует его для регистрации учетной записи на сервере.

**Примечание**

Если у вас уже есть ключ учетной записи, поместите его в подкаталог клиента перед запуском для повторного использования учетной записи. Файл ключа также можно указать с помощью параметра `account_key` в adc__acme_client.

Некоторые CA требуют привязать учетную запись ACME к уже существующей внешней учетной записи с помощью External Account Binding (EAB), прежде чем выпускать сертификаты. Для этого задайте параметр `eab` директивы adc__acme_client, указав идентификатор ключа и MAC-ключ, предоставленные CA; точный синтаксис приведен в описании директивы.

Клиент ACME также использует отдельный ключ (`private.key`) для запросов на подпись сертификата (CSR); если нужно, этот ключ сертификата также создается автоматически при запуске.

При запуске клиент запрашивает сертификат, если его еще нет, подписывая и отправляя CSR для всех доменов, которыми он управляет, серверу CA. Сервер проверяет владение доменом путем HTTP- или DNS-проверки и выдает сертификат, который клиент сохраняет локально (`certificate.pem`).

Как сказано выше, сертификат будет единым для всех доменных имен, для которых используется один и тот же ACME-клиент, то есть потенциально может быть мультидоменным. Список всех имен, для которых выдан сертификат, см. в разделе *Subject Alternative Name* (SAN) полученного сертификата. Проверить его можно в командной строке, например:

```
$ openssl x509 -in certificate.pem -noout -text | grep -A5 "Subject Alternative Name"
```

Когда приближается завершение срока действия сертификата или изменяется список доменов, клиент подписывает и отправляет еще один CSR на сервер CA. Сервер снова проверяет владение и выдает новый сертификат, который клиент устанавливает локально, заменяя предыдущий.

В конфигурации полученный сертификат и соответствующий ключ доступны через префиксные переменные adc__v_acme_cert_name и adc__v_acme_cert_key_name. Их значения — содержи-

мое соответствующих файлов, которое следует использовать с директивами adc__ssl_certificate и adc__ssl_certificate_key, например:

```
server {

    listen 443 ssl;

    server_name example.com www.example.com;
    acme example;

    ssl_certificate $acme_cert_example;
    ssl_certificate_key $acme_cert_key_example;
}
```

**Примечание**

Клиент ACME определяет адрес сервера CA по его имени через adc__resolver того же контекста, который по умолчанию обращается к DNS-серверам из `/etc/resolv.conf`; задавайте директиву явно, только если нужны другие серверы или параметры. На узле без поддержки IPv6 resolver может все равно отправлять AAAA-запросы (IPv6) для сервера CA и не суметь подключиться; отключите их параметром `ipv6=off`, например `resolver conf ipv6=off;`.

#### 3.3.8.3 Сбор доменов и использование сертификата

Директива adc__acme служит только для сбора доменных имен для запросов сертификатов. Она не определяет, где можно использовать сертификат: любой блок `server` может ссылаться на полученный сертификат через переменную $acme_cert_<имя>, независимо от того, содержит ли блок директиву `acme`.

Например, если у вас есть блок `server` с маской, который уже охватывает все поддомены, дополнительные блоки `server` для конкретных поддоменов не нуждаются в директиве `acme`:

```
http {

    acme_client example https://acme-v02.api.letsencrypt.org/directory
        challenge=dns;

    # В этом блоке перечислены домены для запроса сертификата
    server {

        listen 443 ssl;

        server_name example.com *.example.com;
        acme example;

        ssl_certificate $acme_cert_example;
        ssl_certificate_key $acme_cert_key_example;
    }

    # Этот блок использует тот же сертификат, но не добавляет
    # свой server_name в запрос на сертификат
    server {

        listen 443 ssl;

        server_name app.example.com;

        ssl_certificate $acme_cert_example;
```

```
        ssl_certificate_key $acme_cert_key_example;
    }
}
```

### Явный список доменов

Чтобы точно контролировать набор доменных имен в сертификате, не полагаясь на автоматический сбор из всех блоков `server`, создайте отдельный блок `server`, содержащий только директивы adc__server_name и adc__acme. Чтобы этот блок не обрабатывал реальный трафик, привяжите его к Unix-сокету:

```
# Отдельный блок, определяющий список доменов сертификата
server {

    listen unix:/tmp/acme_example.sock;

    server_name example.com www.example.com;
    acme example;
}
```

Другие блоки `server` могут затем использовать сертификат через переменную $acme_cert_<имя>, не влияя на то, какие домены запрашиваются.

### Отдельные сертификаты для разных доменов

Каждый adc__acme_client управляет одним сертификатом. Чтобы получить несколько независимых сертификатов (например, для несвязанных доменов, которым не следует использовать общий сертификат), настройте отдельный adc__acme_client для каждого из них в блоке `http` и укажите в директиве adc__acme каждого блока `server` соответствующий клиент по имени:

```
http {

    # Два независимых клиента, каждый управляет своим сертификатом
    acme_client shop https://acme-v02.api.letsencrypt.org/directory
        challenge=http;

    acme_client blog https://acme-v02.api.letsencrypt.org/directory
        challenge=http;

    server {

        listen 443 ssl;

        server_name shop.example.com www.shop.example.com;
        acme shop;

        ssl_certificate $acme_cert_shop;
        ssl_certificate_key $acme_cert_key_shop;
    }

    server {

        listen 443 ssl;

        server_name blog.example.com;
        acme blog;
```

```
        ssl_certificate $acme_cert_blog;
        ssl_certificate_key $acme_cert_key_blog;
    }
}
```

#### 3.3.8.4 HTTP-проверка

Проверка выполняется автоматически. Когда Angie заказывает сертификат, ACME-сервер запрашивает по HTTP файл-токен по адресу `/.well-known/acme-challenge/<TOKEN>`. Angie отвечает на такие запросы сам, поэтому отдавать этот путь вручную не требуется. Получив файл, ACME-сервер сверяет токен с выданным значением и, если они совпадают, засчитывает проверку домена.

##### Пример конфигурации

Здесь ACME-клиент с именем `example` управляет сертификатами для `example.com` и `www.example.com` (учтите, что wildcard-сертификаты не поддерживаются при HTTP-проверке):

```
http {

    acme_client example https://acme-v02.api.letsencrypt.org/directory;

    server {

        listen 80; # Не обязательно: если порт HTTP-проверки
                   # никто не слушает, модуль откроет его сам
                   # (см. директиву 'acme_http_port')

        listen 443 ssl;

        server_name example.com www.example.com;
        acme example;

        ssl_certificate $acme_cert_example;
        ssl_certificate_key $acme_cert_key_example;
    }
}
```

Как уже отмечалось, порт 80 должен быть открыт для приема вызовов ACME по HTTP. Если ни один сервер не слушает порт HTTP-проверки, модуль создает отдельный слушающий сокет на порту 80 (или указанном в adc__acme_http_port). Отдельный блок adc__server не требуется.

#### 3.3.8.5 DNS-проверка

Проверка выполняется автоматически, но требует настройки DNS с вашей стороны. Когда Angie заказывает сертификат, ACME-сервер отправляет DNS-запрос TXT-записи в поддомене `_acme-challenge.` проверяемого домена. Angie отвечает на такие запросы сам: он выступает авторитетным сервером имен для этого поддомена и возвращает TXT-запись с тем значением, которого ждет ACME-сервер. ACME-сервер сверяет полученную запись с выданным значением и, если они совпадают, засчитывает проверку домена.

Чтобы запрос дошел до Angie, DNS вашего домена должен делегировать поддомен `_acme-challenge.` серверу Angie; нужные записи приведены ниже. По умолчанию Angie отвечает с очень небольшим TTL, а некоторые DNS-провайдеры отфильтровывают ответы с таким низким TTL — если проверка не завершается, увеличьте его директивой adc__acme_dns_ttl.

DNS-проверка подтверждает контроль над доменным именем, поэтому не может проверять идентификаторы в виде IP-адресов: для клиента с `challenge=dns` Angie игнорирует IP-адрес в adc__server_name и выводит предупреждение при запуске. Блок `server`, в котором указан IP-

адрес, должен по-прежнему содержать хотя бы одно доменное имя, которое клиент может проверить.

> **Примечание**
>
> Сервер Angie должен быть доступен из интернета на UDP-порту 53 (или указанном в adc__acme_dns_port). Если сервер находится за межсетевым экраном, убедитесь, что этот порт открыт для входящих соединений.

Например, чтобы подтвердить домен `example.com`, используя сервер Angie с IP-адресом `203.0.113.10`, DNS-конфигурация вашего домена должна включать следующие записи:

```
_acme-challenge.example.com. 60    IN      NS       ns.example.com.
            ns.example.com. 60     IN       A       203.0.113.10
```

Эта конфигурация делегирует разрешение DNS для `_acme-challenge.example.com` на `ns.example.com`, обеспечивая доступность `ns.example.com` путем сопоставления с IP-адресом (`203.0.113.10`).

> **Предупреждение**
>
> Распространение изменений NS-записей может занимать от нескольких минут до 48 часов в зависимости от TTL и DNS-провайдера. Рекомендуется проверить корректность настройки перед запросом сертификата.

Для проверки корректности настройки DNS можно использовать следующие команды:

```
$ dig NS _acme-challenge.example.com +short  # Проверка NS-записи для поддомена _acme-
↪challenge

  ns.example.com.

$ dig A ns.example.com +short  # Проверка A-записи для сервера имен

  203.0.113.10

$ nc -zv 203.0.113.10 53  # Проверка доступности DNS-сервера на порту 53
```

Этот способ позволяет запрашивать wildcard-сертификаты, например сертификат, включающий запись `*.example.com` в разделе *Subject Alternative Name* (SAN). Чтобы в явной форме запросить сертификат для поддомена, например `www.example.com`, следует отдельно подтвердить этот поддомен описанным выше способом.

> **Предупреждение**
>
> Применимость данного сценария во многом зависит от возможностей, предоставляемых вашим DNS-провайдером; некоторые провайдеры не позволяют выполнять такие настройки.

#### Пример конфигурации

В целом, конфигурация схожа с примером из предыдущего раздела. Нет необходимости в настройках, специфичных для HTTP; вместо этого достаточно установить `challenge=dns` для директивы adc__acme_client.

Здесь ACME-клиент с именем `example` управляет сертификатами для `example.com` и `*.example.com`:

```
http {

    acme_client example https://acme-v02.api.letsencrypt.org/directory
        challenge=dns;

    server {

        server_name example.com *.example.com;
        acme example;

        ssl_certificate $acme_cert_example;
        ssl_certificate_key $acme_cert_key_example;
    }
}
```

#### 3.3.8.6 ALPN-проверка

Проверка работает автоматически. ACME-сервер устанавливает TLS-соединение и запрашивает через ALPN протокол `acme-tls/1`. Модуль отдает временный сертификат для валидационного запроса.

Чтобы использовать этот способ, задайте `challenge=alpn` в директиве adc__acme_client и убедитесь, что ваш TLS-слушатель доступен на порту 443 (или на используемом TLS-порту).

##### Пример конфигурации

Конфигурация аналогична предыдущим разделам; достаточно задать `challenge=alpn` для директивы adc__acme_client и убедиться, что TLS-сервер доступен на порту 443.

Здесь ACME-клиент с именем `example` управляет сертификатом для `example.com` и `www.example.com`:

```
http {

    acme_client example https://acme-v02.api.letsencrypt.org/directory
        challenge=alpn;

    server {

        listen 443 ssl;

        server_name example.com www.example.com;
        acme example;

        ssl_certificate $acme_cert_example;
        ssl_certificate_key $acme_cert_key_example;
    }
}
```

#### 3.3.8.7 Проверка с помощью хуков

В отличие от предыдущих способов, эта проверка требует дополнительных усилий. Здесь ACME-сервер производит обычную HTTP-проверку или DNS-проверку, но обращается не к самому серверу Angie, а к внешнему сервису, которым сервер Angie управляет с помощью вызовов-хуков (adc__acme_hook). В свою очередь, этот сервис настраивает отдельный DNS- или HTTP-сервер, куда и направляются запросы ACME-сервера.

Получив ожидаемый ответ от настроенного таким образом DNS- или HTTP-сервера, ACME-сервер подтверждает, что домен принадлежит клиенту.

Когда для выпуска или обновления сертификата требуется проверка домена, Angie формирует внутренний запрос к именованному `location`, содержащему директиву adc__acme_hook. Способ обработки этого запроса полностью зависит от других директив, заданных в том же `location`.

Общая схема такова:

1. Создайте именованный `location` с директивой adc__acme_hook.
2. Настройте обработчик запроса в том же `location` с помощью модуля, подходящего для вашей конфигурации: adc__fastcgi_pass для FastCGI, adc__proxy_pass для HTTP, `cgi_pass` для CGI-скриптов и т. д.
3. Передайте обработчику переменные ACME с помощью поддерживаемого им механизма, например, `fastcgi_param` для FastCGI или `cgi_set_var` для CGI.

Обработчик должен возвращать код состояния `2xx`, который можно передать через заголовок `Status`. Любой другой код от хука `add` прерывает попытку обновления сертификата; от хука `remove` тот же код только записывается в журнал как предупреждение, и обновление продолжается. Вывод обработчика игнорируется.

### Минимальная конфигурация

Независимо от используемого обработчика, `location` для хука имеет следующую структуру:

```
location @acme_hook_location {

    acme_hook example;

    # Директива обработчика (fastcgi_pass, proxy_pass, cgi_pass, ...)
    # Передайте переменные ACME с помощью механизма обработчика:
    #   ACME_HOOK       - $acme_hook_name ("add" или "remove")
    #   ACME_CHALLENGE  - $acme_hook_challenge ("dns" или "http")
    #   ACME_DOMAIN     - $acme_hook_domain
    #   ACME_TOKEN      - $acme_hook_token
    #   ACME_KEYAUTH    - $acme_hook_keyauth
}
```

При DNS-проверке обработчик должен использовать `ACME_HOOK` для определения действия: если значение `add`, создать TXT-запись для `_acme-challenge.`*`ACME_DOMAIN`* со значением из `ACME_KEYAUTH`; если значение `remove`, удалить эту запись.

### Пример с FastCGI

Здесь настраивается ACME-клиент `example` для подтверждения домена при помощи DNS-вызова, на что указывает параметр `challenge=dns` директивы adc__acme_client.

Блок `server` применяется ко всем поддоменам `example.com` (например, `*.example.com`) и использует ACME-клиент `example` для управления сертификатами, что указано в директиве adc__acme.

Именованный блок `location` обрабатывает вызовы хуков. Директива adc__acme_hook связывает его с ACME-клиентом `example`. Запросы хуков отправляются на локальный FastCGI-сервер на порту 9000 с помощью adc__fastcgi_pass. Директивы `fastcgi_param` передают переменные ACME во внешний сервис.

```
acme_client example https://acme-v02.api.letsencrypt.org/directory
    challenge=dns;

server {

    listen 80;

    server_name *.example.com;
```

```
    acme example;

    ssl_certificate $acme_cert_example;
    ssl_certificate_key $acme_cert_key_example;

    location @acme_hook_location {

        acme_hook example;

        fastcgi_pass localhost:9000;

        fastcgi_param ACME_CLIENT $acme_hook_client;
        fastcgi_param ACME_HOOK $acme_hook_name;
        fastcgi_param ACME_CHALLENGE $acme_hook_challenge;
        fastcgi_param ACME_DOMAIN $acme_hook_domain;
        fastcgi_param ACME_TOKEN $acme_hook_token;
        fastcgi_param ACME_KEYAUTH $acme_hook_keyauth;

        include fastcgi.conf;
    }
}
```

Пример соответствующего внешнего FastCGI-сервиса на Perl:

```
#!/usr/bin/perl

use strict; use warnings;

use FCGI;

my $socket = FCGI::OpenSocket(":9000", 5);
my $request = FCGI::Request(\*STDIN, \*STDOUT, \*STDERR, \%ENV, $socket);

while ($request->Accept() >= 0) {
    print "\r\n";

    my $client =    $ENV{ACME_CLIENT};
    my $hook =      $ENV{ACME_HOOK};
    my $challenge = $ENV{ACME_CHALLENGE};
    my $domain =    $ENV{ACME_DOMAIN};
    my $token =     $ENV{ACME_TOKEN};
    my $keyauth =   $ENV{ACME_KEYAUTH};

    if ($hook eq 'add') {

        DNS_set_TXT_record("_acme-challenge.$domain.", $keyauth);

    } elsif ($hook eq 'remove') {

        DNS_clear_TXT_record("_acme-challenge.$domain.");
    }
};

FCGI::CloseSocket($socket);
```

Здесь `DNS_set_TXT_record()` и `DNS_clear_TXT_record()` — функции, предположительно добавляющие и удаляющие TXT-записи в конфигурации некого внешнего DNS-сервера, к которому и

обратится ACME-сервер. В этих записях должны содержаться переданные сервером Angie данные, что позволит внешнему DNS-серверу успешно пройти проверку, аналогичную описанной в разделе *DNS-проверка*. Подробности реализации таких функций выходят за рамки этого руководства; так, например, передавать параметры можно и через URI запроса:

```
# ...

location @acme_hook_location {

    acme_hook example uri=/acme_hook/$acme_hook_name?domain=$acme_hook_domain&key=
↪$acme_hook_keyauth;

    fastcgi_pass localhost:9000;

    fastcgi_param REQUEST_URI $request_uri;
    fastcgi_param ACME_CLIENT $acme_hook_client;
    fastcgi_param ACME_CHALLENGE $acme_hook_challenge;
    fastcgi_param ACME_TOKEN $acme_hook_token;

    include fastcgi.conf;
}
```

### Пример с PHP-FPM

Еще один пример, с использованием PHP-FPM:

```
location @acme_hook_location {

    acme_hook example;
    root /var/www/dns;
    fastcgi_pass unix:/run/php-fpm/php-dns.sock;
    fastcgi_index hook.php;
    fastcgi_param SCRIPT_FILENAME /var/www/dns/hook.php;
    include fastcgi_params;

    fastcgi_param ACME_CLIENT $acme_hook_client;
    fastcgi_param ACME_HOOK $acme_hook_name;
    fastcgi_param ACME_CHALLENGE $acme_hook_challenge;
    fastcgi_param ACME_DOMAIN $acme_hook_domain;
    fastcgi_param ACME_TOKEN $acme_hook_token;
    fastcgi_param ACME_KEYAUTH $acme_hook_keyauth;
}
```

```
[dns]
listen = /run/php-fpm/php-dns.sock
listen.mode = 0666
user = angie
group = angie
chdir = /var/www/dns
# ...
```

Переданные параметры доступны в PHP через `$_SERVER['...']`.

### Пример с CGI и octoDNS

Здесь хук — это скрипт, запускаемый модулем CGI; скрипт добавляет и удаляет TXT-запись `_acme-challenge` через octoDNS, инструмент для описания DNS как кода: зона задается в YAML и синхронизируется с DNS-провайдером. В примере используется Cloudflare, но точно так же подойдет любой другой провайдер octoDNS, если объявить его в `providers` и указать в `targets`. У API-токена должны быть права `Zone:Read`, `DNS:Read` и `DNS:Edit` для этой зоны.

Модуль подключается в контексте `main`. ACME-клиент `example` заказывает один сертификат для `example.com` и `*.example.com`, подтверждая домен при помощи DNS-вызовов, на что указывает параметр `challenge=dns` директивы adc__acme_client. Вызовы хуков обрабатывает именованный блок `location`: директива adc__acme_hook связывает его с клиентом, директивы `cgi_set_var` передают переменные ACME под стандартными именами, а `cgi_pass` запускает скрипт при каждом вызове. Из этих переменных скрипт использует `ACME_CHALLENGE`, `ACME_HOOK`, `ACME_DOMAIN` и `ACME_KEYAUTH`.

```
# в контексте main
load_module modules/ngx_http_cgi_module.so;

acme_client example https://acme-v02.api.letsencrypt.org/directory
    challenge=dns;

server {

    listen 443 ssl;

    server_name example.com *.example.com;

    acme example;

    ssl_certificate $acme_cert_example;
    ssl_certificate_key $acme_cert_key_example;

    location @acme_hook_location {

        acme_hook example;

        cgi_set_var ACME_CLIENT $acme_hook_client;
        cgi_set_var ACME_HOOK $acme_hook_name;
        cgi_set_var ACME_CHALLENGE $acme_hook_challenge;
        cgi_set_var ACME_DOMAIN $acme_hook_domain;
        cgi_set_var ACME_TOKEN $acme_hook_token;
        cgi_set_var ACME_KEYAUTH $acme_hook_keyauth;

        cgi_pass /usr/share/angie/cgi-bin/acme-octodns;
    }
}
```

octoDNS собирает зону из источников по порядку и объединяет их, поэтому хук хранит проверочную запись в отдельном каталоге-источнике `/etc/octodns/acme`, и обновление сертификата никогда не переписывает файл зоны в `/etc/octodns/zones`. Параметр `ignore_missing_zones` (octoDNS 1.16 и новее) разрешает каталогу `/etc/octodns/acme` оставаться пустым между обновлениями. Сам файл зоны не должен содержать записей `_acme-challenge`.

Список 1: /etc/octodns/octodns.yaml

```
providers:
  config:
    class: octodns.provider.yaml.YamlProvider
```

```
    directory: /etc/octodns/zones
  acme:
    class: octodns.provider.yaml.YamlProvider
    directory: /etc/octodns/acme
    ignore_missing_zones: true
  cloudflare:
    class: octodns_cloudflare.CloudflareProvider
    token: YOUR_API_TOKEN

zones:
  example.com.:
    sources:
      - config
      - acme
    targets:
      - cloudflare
```

В файле зоны находятся записи, которые вы ведете сами:

Список 2: /etc/octodns/zones/example.com.yaml

```
'':
  type: A
  value: 203.0.113.10
www:
  type: A
  value: 203.0.113.10
```

Чтобы проверить конфигурацию octoDNS вручную, запустите `octodns-sync` без `--doit`.

Скрипт читает переменные ACME из своего окружения и записывает проверочную запись в источник `acme`:

### Скрипт acme-octodns

Список 3: /usr/share/angie/cgi-bin/acme-octodns

```
#!/opt/octodns/bin/python3
# Хук DNS-01 для ACME-клиента Angie: хранит TXT-запись _acme-challenge
# в отдельном каталоге-источнике octodns и публикует ее через octodns-sync.

import fcntl
import os
import subprocess
import sys
import time

import dns.message
import dns.query
import dns.resolver
import yaml

CONFIG = "/etc/octodns/octodns.yaml"
ACME_DIR = "/etc/octodns/acme"
SYNC = "/opt/octodns/bin/octodns-sync"
TTL = 120
WAIT = int(os.environ.get("ACME_WAIT", "300"))
```

```
def respond(code, message=""):
    if message:
        print(message, file=sys.stderr)
    sys.stdout.write(f"Status: {code}\r\n\r\n")
    sys.exit(0)

def find_zone(domain):
    with open(CONFIG) as f:
        zones = [z.rstrip(".") for z in yaml.safe_load(f)["zones"]]
    match = [z for z in zones if domain == z or domain.endswith("." + z)]
    if not match:
        respond(500, f"no zone for {domain} in {CONFIG}")
    return max(match, key=len)

def update(zone, name, value, add):
    # В одной записи может быть несколько значений: сертификат для example.com
    # и *.example.com проверяется двумя TXT-значениями с одним именем.
    path = f"{ACME_DIR}/{zone}.yaml"
    with open(f"{ACME_DIR}/.lock", "w") as lock:
        fcntl.flock(lock, fcntl.LOCK_EX)
        data = {}
        if os.path.exists(path):
            with open(path) as f:
                data = yaml.safe_load(f) or {}
        values = set(data.get(name, {}).get("values", []))
        (values.add if add else values.discard)(value)
        if values:
            data[name] = {"type": "TXT", "ttl": TTL, "values": sorted(values)}
        else:
            data.pop(name, None)
        if data:
            with open(path, "w") as f:
                yaml.safe_dump(data, f)
        elif os.path.exists(path):
            os.remove(path)
        cmd = [SYNC, f"--config-file={CONFIG}", "--doit", zone + "."]
        run = subprocess.run(cmd, capture_output=True, text=True)
        if run.returncode != 0:
            respond(500, "octodns-sync failed:\n" + run.stdout + run.stderr)

def has_txt(server, fqdn, value):
    try:
        reply = dns.query.udp(dns.message.make_query(fqdn, "TXT"), server, timeout=3)
    except Exception:
        return False
    return any(value == b"".join(getattr(r, "strings", ())).decode()
               for rrset in reply.answer for r in rrset)

def wait_visible(zone, fqdn, value):
    # ACME-сервер обращается к авторитетным серверам зоны, поэтому опрашиваем
    # их напрямую: резолвер мог закэшировать отсутствие записи.
    deadline = time.monotonic() + WAIT
    while time.monotonic() < deadline:
        try:
            servers = [str(a) for ns in dns.resolver.resolve(zone, "NS")
                       for a in dns.resolver.resolve(str(ns.target), "A")]
            if servers and all(has_txt(s, fqdn, value) for s in servers):
```

```
                return True
        except Exception:
            pass
        time.sleep(5)
    return False

def main():
    hook = os.environ.get("ACME_HOOK")
    domain = os.environ.get("ACME_DOMAIN", "")
    value = os.environ.get("ACME_KEYAUTH", "")
    if os.environ.get("ACME_CHALLENGE") != "dns" or hook not in ("add", "remove"):
        respond(500, "the hook supports only DNS validation")
    zone = find_zone(domain)
    rel = domain[: -len(zone) - 1]
    name = "_acme-challenge" + (f".{rel}" if rel else "")
    update(zone, name, value, hook == "add")
    if hook == "add" and WAIT and not wait_visible(zone, f"{name}.{zone}.", value):
        update(zone, name, value, False)
        respond(500, f"{name}.{zone} is not visible on the name servers after {WAIT}
↪ s")
    respond(200)

try:
    main()
except Exception as e:
    respond(500, f"{type(e).__name__}: {e}")
```

Как только хук `add` вернет код `2xx`, Angie попросит ACME-сервер выполнить проверку и будет опрашивать результат в течение 60 секунд. Распространения записи в DNS Angie не дожидается, поэтому ждать приходится самому скрипту. Он опрашивает авторитетные серверы имен зоны напрямую, поскольку резолвер мог закэшировать отсутствие записи, и завершается, как только значение отдадут все серверы; по умолчанию он ждет не дольше 300 секунд.

Если значение не появится вовремя, скрипт отвечает `Status: 500`, Angie прерывает попытку и повторяет ее через интервал, заданный параметром retry_after_error. После неудачного `add` Angie не вызывает хук `remove`, поэтому скрипт сам удаляет свое значение, прежде чем ответить.

Предел ожидания задается переменной окружения `ACME_WAIT`. В показанном выше `location` для хука она не задается; чтобы изменить предел, добавьте ее туда, передав значение через переменную, как и остальные:

```
set $acme_wait 600;
cgi_set_var ACME_WAIT $acme_wait;
```

Значение `0` отключает ожидание — это подходит для собственных авторитетных серверов, на которых изменение через API видно сразу.

Переменная adc__v_acme_hook_domain приходит без префикса `*.`, поэтому сертификат для `example.com` и `*.example.com` порождает два вызова `add` для одного и того же имени `_acme-challenge.example.com`. Скрипт хранит список значений в одной записи, а при `remove` удаляет только значение из этого вызова; когда значений не остается, он удаляет и саму запись, и хранивший ее файл в `/etc/octodns/acme`.

Скрипт выполняется от имени пользователя рабочих процессов Angie — `angie`. Ему нужен доступ на чтение к конфигурации octoDNS, где хранится токен, и на запись в `/etc/octodns/acme`; больше ничего. CGI-процесс не наследует окружение Angie, поэтому токен хранится в файле конфигурации, а скрипт использует абсолютные пути.

Стандартный поток ошибок скрипта попадает в журнал ошибок Angie на уровне `warn` — именно там разбирают сбои хуков; эти сообщения видны только при уровне `warn` или менее серьезном.

Чтобы все это настроить в системе на основе Debian:

1. Установите модуль CGI:

```
$ sudo apt-get install angie-module-cgi
```

2. Установите octoDNS и провайдер Cloudflare в отдельное виртуальное окружение:

```
$ sudo apt-get install python3-venv
$ sudo python3 -m venv /opt/octodns
$ sudo /opt/octodns/bin/pip install 'octodns>=1.16' octodns-cloudflare
```

3. Создайте каталоги; пакет модуля не создает `/usr/share/angie/cgi-bin`:

```
$ sudo install -d /etc/octodns/zones /usr/share/angie/cgi-bin
$ sudo install -d -o angie -g angie -m 750 /etc/octodns/acme
```

4. Создайте `/etc/octodns/octodns.yaml` и `/etc/octodns/zones/example.com.yaml`, как показано выше, и ограничьте доступ к конфигурации группой `angie`, поскольку в ней хранится API-токен:

```
$ sudo chgrp angie /etc/octodns/octodns.yaml
$ sudo chmod 640 /etc/octodns/octodns.yaml
```

5. Сохраните показанный выше скрипт под именем `acme-octodns` и установите его:

```
$ sudo install -m 755 acme-octodns /usr/share/angie/cgi-bin/
```

6. Добавьте показанную выше конфигурацию, проверьте ее и перезагрузите Angie:

```
$ sudo angie -t
$ sudo kill -HUP $(cat /run/angie.pid)
```

   Сразу после перезагрузки клиент запрашивает сертификат; его состояние и статус сертификата видны в разделе API /status/http/acme_clients/, а полученный сертификат сохраняется в каталоге хранения клиента.

#### 3.3.8.8 ACME в потоковом модуле

Потоковый модуль ACME позволяет автоматизировать выпуск и использование сертификатов для TCP-трафика. Для его корректной работы необходимо сначала настроить HTTP-аналог: ACME-клиент должен быть объявлен в контексте `http`, а сам блок `stream` должен располагаться *после* блока `http` в конфигурации.

##### Пример конфигурации

По умолчанию для получения сертификатов используется режим HTTP-проверки. Для него, как упоминалось в разделе *HTTP-проверка*, нужен HTTP-сервер, слушающий на порту 80:

```
# HTTP-часть
http {

    # ACME-клиент для потоковой части
    acme_client example https://acme-v02.api.letsencrypt.org/directory;

    # Сервер для HTTP-проверки
    server {

        listen 80;
        return 444;
```

```
    }
}

# Потоковая часть
stream {

    server {

        listen 12345 ssl;
        proxy_pass backend_upstream;

        ssl_certificate $acme_cert_example;
        ssl_certificate_key $acme_cert_key_example;

        server_name example.com www.example.com;
        acme example; # ссылка на ACME-клиент, определенный в HTTP-части
    }

    upstream backend_upstream {

        server 127.0.0.1:54321;
    }
}
```

Также можно использовать DNS-проверку, настроив `challenge=dns` в директиве adc__acme_client; тогда сервер будет не нужен.

#### 3.3.8.9 Миграция с certbot

Если до перехода с nginx на Angie вы использовали certbot для получения и продления SSL-сертификатов от центра сертификации Let's Encrypt, выполните следующие шаги, чтобы перейти к использованию нашего модуля ACME.

Предположим, вы настроили сертификаты следующим образом:

```
$ sudo certbot --nginx -d example.com -d www.example.com
```

Автоматически созданная при этом конфигурация обычно находится в файле `/etc/nginx/sites-available/example.conf` и выглядит приблизительно так:

```
server {

    listen 80;
    server_name example.com www.example.com;
    return 301 https://$host$request_uri;
}

server {

    listen 443 ssl;
    server_name example.com www.example.com;

    root /var/www/example;
    index index.html;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
```

```
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}
```

В примере выше выделены строки, которые потребуется изменить. В зависимости от ваших обстоятельств и предпочтений настройте HTTP-проверку или DNS-проверку с помощью модуля ACME.

Итоговая конфигурация Angie может выглядеть приблизительно так:

```
http {

    acme_client example https://acme-v02.api.letsencrypt.org/directory;

    server {

        listen 80;
        server_name example.com www.example.com;
        return 301 https://$host$request_uri;
    }

    server {
        listen 443 ssl;
        server_name example.com www.example.com;

        root /var/www/example;
        index index.html;

        acme                 example;

        ssl_certificate      $acme_cert_example;
        ssl_certificate_key  $acme_cert_key_example;
    }
}
```

Не забудьте перезагрузить конфигурацию после изменения:

```
$ sudo kill -HUP $(cat /run/angie.pid)
```

Убедившись, что эта конфигурация работает, вы можете удалить сертификаты `certbot`, а также отключить или целиком удалить его с сервера, если он больше нигде не используется, например:

```
$ sudo rm -rf /etc/letsencrypt

$ sudo systemctl stop certbot.timer
$ sudo systemctl disable certbot.timer
$ # -- или --
$ sudo rm /etc/cron.d/certbot

$ sudo apt remove certbot
$ # -- или --
$ sudo dnf remove certbot
```

### 3.3.9 Настройка пользовательских метрик

Angie может собирать пользовательские числовые метрики в разделяемой памяти и выводить их через API статистики по адресу `/status/http/metric_zones/`. Это обеспечивает модуль Metric.

**Примечание**

Это руководство описывает метрики для HTTP-трафика. Для TCP- и UDP-трафика см. потоковый модуль Metric, который выводит свои зоны по адресу `/status/stream/metric_zones/`.

#### 3.3.9.1 Шаги настройки

1. Создайте зону метрик в блоке `http`:
   - adc__metric_zone создает зону с одной метрикой.
   - adc__metric_complex_zone создает зону с несколькими именованными метриками.
2. Обновляйте метрики при обработке запросов директивой adc__metric. Используйте пару `key=value` (оба — комплексные значения), и выберите этап обновления параметром `on=` (`request`, `response` или `end`).
3. Откройте API через `location`:

```
location /status/ {
    api /status/http/metric_zones/;
}
```

#### 3.3.9.2 Пример

Подсчет запросов по хостам и вывод метрик через API:

```
http {
    metric_zone requests:128k count;

    server {
        listen 80;

        location / {
            metric requests $host=1;
        }

        location /status/ {
            api /status/http/metric_zones/;
        }
    }
}
```

#### 3.3.9.3 Примечания

- При `expire=on` и переполнении памяти истекают самые давно неиспользуемые записи. При `expire=off` новые обновления отбрасываются, а счетчик `discarded` увеличивается.
- Если задан `discard_key`, метрики истекших записей агрегируются под этим ключом в API.
- Длина ключей и значений ограничена 255 байт; длинные ключи усекутся в API.
- Пустое значение трактуется как `0`, а непустая строка без числа в начале — как `1`.

### 3.3.10 Глобальная балансировка с учетом мониторов доступности

Рассмотрим схему, где:

- Для домена `www.example.com` используется два дата-центра, ЦОД-1 и ЦОД-2.
- В дата-центре ЦОД-1 есть два бэкенда (приложения), app1 и app2, и база данных db1.

- В дата-центре ЦОД-2 есть два бэкенда (приложения), app3 и app4, и две базы данных, db2 и db3.
- В каждом дата-центре установлен Angie ADC, dc1 и dc2, соответственно.
- Трафик распределяется между дата-центрами по методу балансировки `round-robin`. При этом учитывается доступность дата-центров. Дата-центр считается доступным, если одновременно доступны хотя бы один бэкенд (приложение) и все имеющиеся базы данных этого дата-центра.

| Дата-центр | Бэкенд | База данных | Узел Angie ADC |
|---|---|---|---|
| ЦОД-1 | `app1`, `app2` | `db1` | `dc1` |
| ЦОД-2 | `app3`, `app4` | `db2`, `db3` | `dc2` |

#### 3.3.10.1 Настройка

1. На любом узле кластера Angie ADC перейдите в раздел `Система` → `Доступность` → вкладка `Датчики` и создайте семь датчиков (см. *Настройка датчиков и мониторов*).

| Датчик (имя датчика автоматически формируется из имени проверяемого ресурса и имени узла) | Узел кластера | Проверка | Ресурс |
|---|---|---|---|
| `dc1-app1` | `dc1` | `HTTP` | `app1` |
| `dc1-app2` | `dc1` | `HTTP` | `app2` |
| `dc1-db1` | `dc1` | `TCP` | `db1` |
| `dc2-app3` | `dc1` | `HTTP` | `app3` |
| `dc2-app4` | `dc2` | `HTTP` | `app4` |
| `dc2-db2` | `dc2` | `TCP` | `db2` |
| `dc2-db3` | `dc2` | `TCP` | `db3` |

2. Перейдите на вкладку `Мониторы` и создайте четыре монитора.

| Монитор | Датчики | Правило |
|---|---|---|
| `mon-dc1-app` | `dc1-app1`, `dc1-app2` | `Любой датчик` |
| `mon-dc1-db` | `dc1-db1` | `Любой датчик` |
| `mon-dc2-app` | `dc2-app3`, `dc2-app4` | `Любой датчик` |
| `mon-dc2-db` | `dc2-db2`, `dc2-db3` | `Все датчики` |

   Здесь:

   - Мониторы приложения используют правило `Любой датчик`, т.е. дата-центр продолжает обслуживать трафик, пока отвечает хотя бы один бэкенд.
   - Мониторы БД используют правило `Все датчики`, т.е. дата-центр продолжает обслуживать трафик, если отвечают все его БД.

3. Перейдите в раздел `Управление трафиком` → `Глобальная балансировка` и настройте конфигурацию GSLB. Для домена `example.com` добавьте динамическую запись типа `A` с именем `www`, балансировкой `round_robin` и двумя бэкендами — по одному на дата-центр (см. *Настройка конфигурации*).

| Дата-центр | Бэкенд (IP-адрес узла кластера Angie ADC) | Мониторы |
|---|---|---|
| ЦОД-1 | `VIP dc1` | `mon-dc1-app`, `mon-dc1-db` |
| ЦОД-2 | `VIP dc2` | `mon-dc2-app`, `mon-dc2-db` |

Оба монитора на бэкенде объединяются по логике **И**: дата-центр доступен, только когда исправны и монитор приложения, и монитор БД. Запросы по методу `round-robin` распределяются по дата-центрам последовательно с учетом их доступности.

4. Настройте делегирование зоны `example.com` на GSLB-узлы Angie ADC в родительской зоне `com`:

```
example.com.      IN NS dc1.example.com.
example.com.      IN NS dc2.example.com.

; Glue-записи - IP-адреса GSLB-узлов Angie ADC
dc1.example.com.  IN A  <IP-адрес dc1>
dc2.example.com.  IN A  <IP-адрес dc2>
```

ГЛАВА 4

# Отказоустойчивость и маршрутизация

Здесь описывается обеспечение отказоустойчивости с помощью пары высокой доступности, а также настройка динамической и статической маршрутизации.

В этом разделе:



## 4.1 Пара высокой доступности

В этом разделе:



Высокая доступность (High Availability, HA) – это режим, при котором два устройства Angie ADC объединяются в пару высокой доступности, обеспечивающую автоматическую синхронизацию конфигураций двух узлов и сохранение работоспособности системы при сбое одного из узлов пары.

Пример типовой схемы работы пары высокой доступности:

**Примечание**

Рекомендуется резервировать все компоненты инфраструктуры. Отказ сопутствующей инфраструктуры (например выход из строя вышестоящего маршрутизатора или сбой серверов) выходит за рамки решения высокой доступности Angie ADC.

Пример отказоустойчивой инфраструктуры:

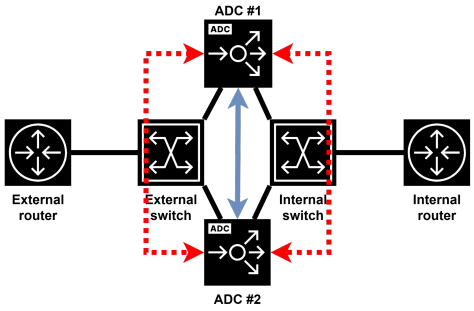



Рис. 1: Рис. 1

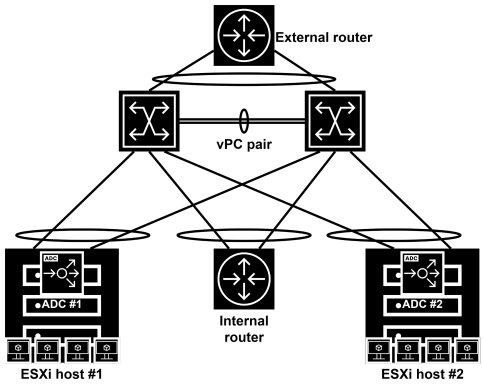



Рис. 2: Рис. 1

### 4.1.1 Обзор работы

Вы можете объединить два устройства Angie ADC в пару высокой доступности в режиме Active / Standby. В дальнейшем будут добавлены другие режимы.

**Примечание**

Перед настройкой пары высокой доступности необходимо создать кластер Angie ADC. Подробнее см. в статье *Создание и масштабирование кластера управления*.

#### 4.1.1.1 Режим Active / Standby

При создании пары одно устройство назначается основным, а второе — резервным. Конфигурации узлов синхронизированы. Основной узел принимает соединения и управляет серверами. Резервный узел находится в режиме ожидания. При выходе из строя основного узла происходит *смена ролей*: резервный узел становится основным, а основной после восстановления переходит в режим ожидания.

Высокая доступность достигается за счет двунаправленных проверок доступности узлов. Узлы отслеживают состояние друг друга, отправляя запросы (UDP-сообщения) на все интерфейсы соседнего узла, участвующие в паре. По умолчанию запросы отправляются раз в 250 мс. Если с момента получения последнего успешного ответа от основного узла прошло больше времени, чем указано в таймауте ожидания, интерфейс будет считаться недоступным. По умолчанию таймаут равен 3 секундам. Если проверка доступности всех интерфейсов основного узла пары завершается неудачно, происходит смена ролей. Параметры опроса можно *настраивать*.

Поддерживается *привязка клиентских сессий* (sticky sessions), позволяющая сохранять клиентские сессии при смене основного узла, принимающего соединения.

Информация о состоянии пары, ролях узлов и событиях в работе пары отображается в веб-интерфейсе Angie ADC. Подробнее см. *Просмотр информации о паре*.

#### 4.1.1.2 Развертывание

Развертывание пары высокой доступности включает следующие шаги:

1. *Предварительные действия*.
2. *Создание пары высокой доступности в режиме Active / Standby*.
3. *Настройка высокой доступности в паре*.

#### 4.1.1.3 Обновление конфигурации и синхронизация

Конфигурации узлов автоматически синхронизируются при создании пары.

> **Примечание**
>
> После изменения конфигурации основного узла необходимо вручную запустить синхронизацию узлов пары.

При синхронизации основной узел передает свою конфигурацию на резервный узел. На резервном узле перезаписываются следующие файлы и папки:

- конфигурация балансировщика нагрузки;
- сертификаты и ключи балансировщика нагрузки;
- конфигурация маршрутизации (VRRP, BGP, OSPF);
- конфигурация GSLB;
- все файлы в папках `/etc/angie/crt` и `/etc/angie-lb/crt`;
- все файлы в папке `/var/transfer`.

Остальные файлы резервного узла, если они находятся вне указанных папок (например файл лицензии), удалены не будут.

Подробнее см. *Обновление конфигурации и синхронизация*.

### 4.1.2 Создание пары высокой доступности

Вы можете создать пару высокой доступности в веб-интерфейсе Angie ADC.

#### 4.1.2.1 Требования

Для создания пары высокой доступности должны выполняться следующие условия:

- Оба устройства Angie ADC должны иметь одинаковую версию ПО и лицензию.

#### 4.1.2.2 Предварительные действия

1. Разверните два устройства Angie ADC из будущей пары.

   При этом должны выполняться следующие условия:

   - В пару можно поставить устройства Angie ADC, у которых хотя бы одна общая подсеть.
   - Интерфейсы самого устройства Angie ADC могут находиться в разных подсетях, но при формировании пар интерфейсов все IP-адреса попарно должны быть из одинаковых подсетей.
   - Каждый узел должен иметь собственный набор уникальных IP-адресов. На данный момент поддерживаются только IPv4-адреса. Необходимо прописать IP-адреса для всех физических интерфейсов, которые будут использоваться для обеспечения высокой доступности.

**Предупреждение**

В дальнейшем не рекомендуется изменять IP-адреса узлов, объединенных в пару высокой доступности, иначе пара потеряет работоспособность.

Мы рекомендуем иметь одинаковое количество интерфейсов на обоих устройствах Angie ADC, чтобы каждый интерфейс получил пару на другом устройстве Angie ADC. Интерфейсы без пары не будут участвовать в обеспечении высокой доступности.

2. Объедините устройства Angie ADC в кластер. Подробнее см. в статье *Создание и масштабирование кластера управления*.
3. Настройте авторизацию на резервном узле Angie ADC с использованием локальной учетной записи `admin` (эти учетные данные необходимы для создания пары).
4. Настройте минимальную конфигурацию узла, который будет иметь роль основного. При создании пары запустится синхронизация конфигураций (конфигурация основного узла будет перенесена на резервный узел). Настройку также можно выполнить позже, после создания пары, при этом необходимо будет вручную запустить синхронизацию конфигураций после настройки.

#### 4.1.2.3 Создание пары высокой доступности в режиме Active / Standby

При создании пары высокой доступности роль основного узла (Active) автоматически назначается тому узлу, с которого создается пара, роль резервного (Standby) – добавляемому узлу.

**Важно**

Для успешной сборки и работы пары высокой доступности необходимо, чтобы время на узлах пары было синхронизировано, например, по *NTP*.

Чтобы создать пару высокой доступности, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Высокая доступность и кластеризация` → `Высокая доступность`.

   Откроется экран подключения резервного узла к текущему узлу.
2. На экране `Создание пары высокой доступности` заполните следующие поля:

   - `IP-адрес` – IP-адрес второго узла пары (IPv4) для удаленного администрирования и мониторинга этого узла. Это может быть выделенный интерфейс только для управления или один из интерфейсов, передающих продуктовый трафик. Этот адрес не распространяется и не синхронизируется.
   - `Логин` и `Пароль` – данные для авторизации на резервном узле.

   **Примечание**

   Используйте логин локальной учетной записи `admin` (см. "Предварительные действия" выше).
3. Нажмите `Подключить`.

   Между двумя узлами будет создано соединение. Откроется окно настройки соответствия интерфейсов.
4. В окне `Настройка соответствия интерфейсов` в раскрывающихся списках основного и резервного узла выберите интерфейсы, которые вы хотите поставить в пару. Опрос доступности узлов будет проводиться в рамках каждой добавленной пары.

Чтобы добавить еще одну пару, нажмите значок `Добавить` и повторите действия.

В списках отображаются только интерфейсы в состоянии `up`. Интерфейсы loopback, в состоянии `down` и без IP-адреса не отображаются и не доступны для выбора.

> **Примечание**
>
> Адреса двух сопоставляемых интерфейсов должны быть из одной подсети. Минимально достаточно одной пары интерфейсов, но мы рекомендуем задать пары для всех интерфейсов обоих узлов.

5. Нажмите `Сохранить`.

   Будет сформирована пара высокой доступности в режиме Active / Standby. Сразу после создания у пары будет отображаться статус `starting` (происходит синхронизация конфигураций). После успешной синхронизации статус поменяется на `on` (пара в рабочем состоянии).

6. После объединения двух устройств Angie ADC в пару необходимо *настроить механизмы высокой доступности* на узлах этой пары.

### 4.1.3 Настройка высокой доступности в паре

После объединения двух устройств Angie ADC в пару необходимо настроить механизмы высокой доступности на узлах этой пары.

Для этого необходимо выбрать механизм обеспечения высокой доступности. Поддерживаются следующие механизмы:

- VRRP;
- OSPF;
- BGP.

Можно одновременно использовать несколько механизмов на разных интерфейсах устройства.

Настройку основного узла можно проводить до или после создания пары высокой доступности. При создании пары уже имеющиеся конфигурации синхронизируются, а после создания пары можно запустить ручную синхронизацию конфигураций.

#### 4.1.3.1 VRRP

Для обеспечения высокой доступности по VRRP необходимо настроить общий VRRP-интерфейс на двух узлах пары. Для этого на каждом узле пары перейдите в CLI (порт 2022) и задайте параметры интерфейса:

```
vrrp <номер_группы> int <имя_интерфейса> ip <IP_адрес>
```

Также настройки, относящиеся к VRRP, можно продублировать, задав их на одном узле и затем запустив ручную *синхронизацию пары*.

На активном узле виртуальный VRRP-интерфейс будет находиться состоянии `UP`, а на резервном узле соответствующий виртуальный интерфейс будет выключен. При смене ролей меняется состояние виртуальных VRRP-интерфейсов на узлах (виртуальный VRRP-интерфейс узла, принявшего на себя роль Active, перейдет в состояние `UP`). Таким образом можно дополнительно организовать резервирование пары высокой доступности другим устройством или группой.

Пример вывода команды `ip a`, выполненной на активном узле (интерфейс `vrrp4-ens33-1@ens33` находится в состоянии `UP`):

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
```

```
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 brd 203.0.113.1 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:96:0c:dc brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 10.65.1.7/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fe96:cdc/64 scope link
valid_lft forever preferred_lft forever
3: vrrp4-ens33-1@ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue␣
↪state UP group default qlen 1000
link/ether 00:00:5e:00:01:01 brd ff:ff:ff:ff:ff:ff
inet 10.65.1.5/24 metric 1000 scope global vrrp4-ens33-1
valid_lft forever preferred_lft forever
inet6 fe80::200:5eff:fe00:101/64 scope link
valid_lft forever preferred_lft forever
```

Пример вывода команды `ip a`, выполненной на резервном узле (интерфейс `vrrp4-ens33-1@ens33` находится в состоянии `DOWN`):

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 brd 203.0.113.1 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:c1:36:b4 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 10.65.1.8/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fec1:36b4/64 scope link
valid_lft forever preferred_lft forever
5: vrrp4-ens33-1@ens33: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue␣
↪state DOWN group default qlen 1000
link/ether 00:00:5e:00:01:01 brd ff:ff:ff:ff:ff:ff protodown on protodown_reason <7>
inet 10.65.1.5/24 metric 1000 scope global vrrp4-ens33-1
valid_lft forever preferred_lft forever
inet6 fe80::200:5eff:fe00:101/64 scope link
valid_lft forever preferred_lft forever
```

Далее вы можете настроить высокую доступность двумя способами:

- Назначение адреса сервиса на виртуальный VRRP-интерфейс.
- Назначение адреса сервиса на интерфейс `lo`.

### Назначение адреса сервиса на виртуальный VRRP-интерфейс

При смене роли узла сервис будет перенесен на тот узел, который взял на себя роль Active. Допускается назначение нескольких IP-адресов одновременно на один виртуальный интерфейс. Для маршрутизатора адрес сервиса находится в подключенной сети, то есть доступен напрямую:

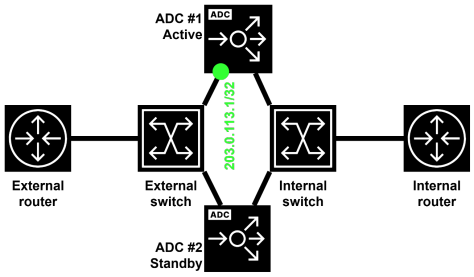



Рис. 3: Рис. 1

Если клиенты находятся в одной IP-сети с адресом сервиса, этот способ будет наиболее удобным:

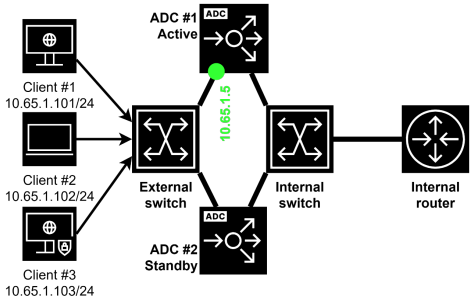



Рис. 4: Рис. 2

### Назначение адреса сервиса на интерфейс lo

При настройке статического маршрута на маршрутизаторе необходимо указать виртуальный VRRP-адрес как адрес следующего перехода (next-hop) для префикса, содержащего адрес сервиса. На интерфейс `lo` может быть назначено несколько адресов одновременно. Мы рекомендуем использовать этот способ как более гибкий:

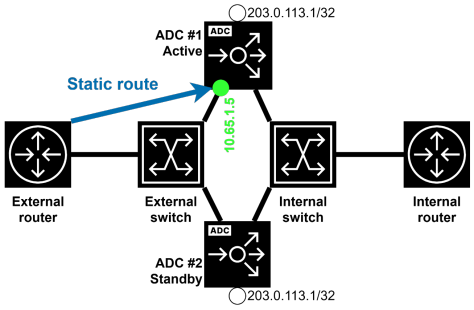



Рис. 5: Рис. 3

#### 4.1.3.2 OSPF

Если в сети используются протоколы динамической маршрутизации, то вы можете добавить пару высокой доступности Angie ADC в OSPF-домен. При этом только основной узел пары будет анонсировать маршрут на адрес сервиса.

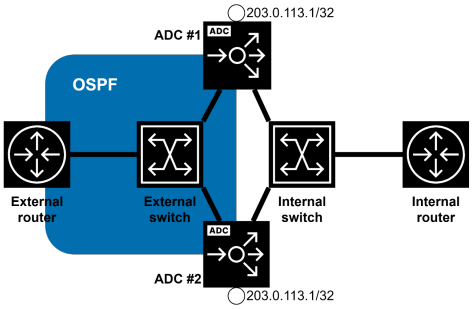



Рис. 6: Рис. 4

### Пример

Сервис доступен по адресу `203.0.113.1`. Необходимо, чтобы основной узел пары анонсировал маршрут на префикс `203.0.113.1/32`.

Сначала нужно вручную назначить адрес `203.0.113.1/32` на интерфейсы loopback обоих узлов пары высокой доступности.

Интерфейсы основного узла:

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:96:0c:dc brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 10.65.1.7/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fe96:cdc/64 scope link
valid_lft forever preferred_lft forever
```

Интерфейсы резервного узла:

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:c1:36:b4 brd ff:ff:ff:ff:ff:ff
```

```
altname enp2s1
inet 10.65.1.8/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fec1:36b4/64 scope link
valid_lft forever preferred_lft forever
```

Далее необходимо добавить поддержку протокола OSPF на интерфейсе `ens33` и включить процесс маршрутизации OSPF, в котором анонсируются все подключенные сети.

```
interface ens33
ip ospf 1 area 0
router ospf 1
redistribute connected
distribute-list %ha%pair% out connected
```

После внесения изменений *синхронизируйте конфигурации в паре*.

**Примечание**

Последняя строка добавляется в момент выполнения синхронизации. Не нужно вносить эту строку вручную, а также удалять или изменять ее.

### Проверка

Проверка на маршрутизаторе или L3-коммутаторе (список OSPF-соседей):

```
R1#sho ip ospf nei

Neighbor ID     Pri   State           Dead Time   Address         Interface
1.1.1.1           1   FULL/BDR        00:00:39    10.65.1.7       GigabitEthernet0/0
2.2.2.2           1   FULL/DR         00:00:38    10.65.1.8       GigabitEthernet0/0
```

Соседство по протоколу OSPF устанавливают оба узла пары одновременно:

```
R1#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set
203.0.113.0/32 is subnetted, 1 subnets
O E2     203.0.113.1 [110/20] via 10.65.1.7, 00:52:39, GigabitEthernet0/0
```

Маршрут анонсируется только одним узлом. В этом можно убедиться, если просмотреть список LSA5 в LSDB маршрутизатора:

```
R1#sho ip os database external
    OSPF Router with ID (10.65.1.140) (Process ID 1)

        Type-5 AS External Link States
Routing Bit Set on this LSA in topology Base with MTID 0
LS age: 1341
```

```
Options: (No TOS-capability, No DC, Upward)
LS Type: AS External Link
Link State ID: 203.0.113.1 (External Network Number )
Advertising Router: 1.1.1.1
LS Seq Number: 80000005
Checksum: 0x575B
Length: 36
Network Mask: /32
Metric Type: 2 (Larger than any link state path)
MTID: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

Теперь выключите основной узел и проверьте изменения:

```
R1#sho ip os ne

Neighbor ID     Pri   State           Dead Time   Address         Interface
2.2.2.2           1   FULL/DR         00:00:33    10.65.1.8       GigabitEthernet0/0
```

```
R1#sho ip os da ex
    OSPF Router with ID (10.65.1.140) (Process ID 1)

        Type-5 AS External Link States
Routing Bit Set on this LSA in topology Base with MTID 0
LS age: 114
Options: (No TOS-capability, No DC, Upward)
LS Type: AS External Link
Link State ID: 203.0.113.1 (External Network Number )
Advertising Router: 2.2.2.2
LS Seq Number: 80000001
Checksum: 0x139B
Length: 36
Network Mask: /32
Metric Type: 2 (Larger than any link state path)
MTID: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

```
R1#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set
203.0.113.0/32 is subnetted, 1 subnets
O E2     203.0.113.1 [110/20] via 10.65.1.8, 00:01:57, GigabitEthernet0/0
R1#
```

Остался только один OSPF-сосед, который стал основным узлом пары. Теперь он анонсирует марш-

рут до префикса `203.0.113.1/32`.

#### 4.1.3.3 BGP

Маршрутизатор устанавливает BGP-сессии с обоими узлами пары. Обе BGP-сессии активны одновременно, но анонс префикса производится только с основного узла.

Пример типовой схемы:

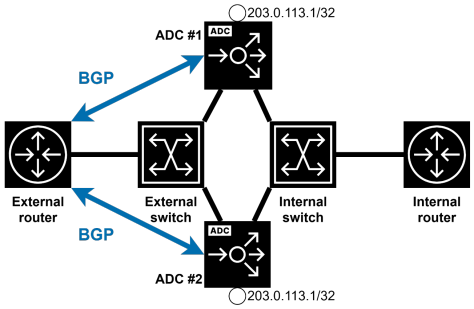



Рис. 7: Рис. 5

#### Пример

Сервис доступен по адресу `203.0.113.1`. Основной узел пары будет анонсировать маршрут на префикс `203.0.113.1/32`.

Сначала необходимо вручную назначить адрес `203.0.113.1/32` на интерфейсы loopback обоих узлов пары высокой доступности.

Интерфейсы основного узла:

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:96:0c:dc brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 10.65.1.7/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fe96:cdc/64 scope link
valid_lft forever preferred_lft forever
```

Интерфейсы резервного узла:

```
ip a

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
```

```
valid_lft forever preferred_lft forever
inet 203.0.113.1/32 scope global lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group␣
↪default qlen 1000
link/ether 00:0c:29:c1:36:b4 brd ff:ff:ff:ff:ff:ff
altname enp2s1
inet 10.65.1.8/24 brd 10.65.1.255 scope global ens33
valid_lft forever preferred_lft forever
inet6 fe80::20c:29ff:fec1:36b4/64 scope link
valid_lft forever preferred_lft forever
```

На основном узле настройте процесс BGP так, чтобы он устанавливал соединение с маршрутизатором (`10.65.1.140`). Также перераспределите подключенные сети в BGP:

```
router bgp 65001
neighbor 10.65.1.140 remote-as 65001
!
address-family ipv4 unicast
redistribute connected
exit-address-family
exit
```

Запустите *синхронизацию конфигураций* между узлами пары высокой доступности, чтобы настроить процесс маршрутизации BGP на резервном узле.

**Примечание**

Ручная настройка конфигурации на резервном узле не должна использоваться.

### Проверка

Пример конфигурации со стороны маршрутизатора:

```
router bgp 65001
bgp log-neighbor-changes
neighbor 10.65.1.7 remote-as 65001
neighbor 10.65.1.8 remote-as 65001
```

Убедитесь, что маршрутную информацию отправляет только основной узел (`10.65.1.7`):

```
R1#sho ip bg su
BGP router identifier 10.65.1.140, local AS number 65001
BGP table version is 17, main routing table version 17
2 network entries using 296 bytes of memory
2 path entries using 128 bytes of memory
1/1 BGP path/bestpath attribute entries using 136 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 560 total bytes of memory
BGP activity 4/2 prefixes, 13/11 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/
↪PfxRcd
10.65.1.7       4        65001    3283    3206       17    0    0 02:43:57        2
```

```
10.65.1.8       4        65001      28      29       17    0    0 00:01:17        0
R1#sho ip bgp
BGP table version is 17, local router ID is 10.65.1.140
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network          Next Hop            Metric LocPrf Weight Path
r>i 10.65.1.0/24     10.65.1.7                0    100      0 ?
*>i 203.0.113.1/32   10.65.1.7                0    100      0 ?
R1#sho ip ro bgp
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set
203.0.113.0/32 is subnetted, 1 subnets
B        203.0.113.1 [200/0] via 10.65.1.7, 02:45:34
R1#
```

Теперь выключите основной узел и проверьте изменения:

```
R1#sho ip bg su
BGP router identifier 10.65.1.140, local AS number 65001
BGP table version is 21, main routing table version 21
2 network entries using 296 bytes of memory
2 path entries using 128 bytes of memory
1/1 BGP path/bestpath attribute entries using 136 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 560 total bytes of memory
BGP activity 6/4 prefixes, 15/13 paths, scan interval 60 secs

Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/
↪PfxRcd
10.65.1.7       4        65001       0       0        1    0    0 00:07:13 Active
10.65.1.8       4        65001     228     222       21    0    0 00:11:04        2
```

Остался только один BGP-сосед (`10.65.1.8`), который стал основным узлом пары. Теперь он анонсирует маршрут до префикса `203.0.113.1/32`:

```
R1#sho ip bg
BGP table version is 21, local router ID is 10.65.1.140
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network          Next Hop            Metric LocPrf Weight Path
r>i 10.65.1.0/24     10.65.1.8                0    100      0 ?
*>i 203.0.113.1/32   10.65.1.8                0    100      0 ?
R1#sho ip ro bg
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override

Gateway of last resort is not set
203.0.113.0/32 is subnetted, 1 subnets
B        203.0.113.1 [200/0] via 10.65.1.8, 00:05:50
```

За анонс отвечает distribute-list с именем `%ha%pair%`, который автоматически прикрепляется ко всем BGP-соседям, описанным в конфигурации. Соответствующая конфигурационная строка добавляется в момент синхронизации конфигурации между узлами пары:

```
router bgp 65001
neighbor 10.65.1.140 remote-as 65001
!
address-family ipv4 unicast
redistribute connected
neighbor 10.65.1.140 distribute-list %ha%pair% out
exit-address-family
exit
```

Если узел является основным, то distribute-list ссылается на разрешающий список доступа:

```
access-list %ha%pair% seq 5 permit any
```

Если узел является резервным, то distribute-list ссылается на запрещающий список доступа:

```
access-list %ha%pair% seq 5 deny any
```

### 4.1.4 Управление парой

В этой статье:

- *Просмотр информации о паре*
- *Обновление конфигурации и синхронизация*
- *Настройка протокола опроса доступности узлов*
- *Удаление пары*

См. также *Привязка клиентских сессий* (sticky sessions).

#### 4.1.4.1 Просмотр информации о паре

1. Откройте веб-интерфейс Angie ADC на одном из узлов пары.
2. Перейдите `Высокая доступность и кластеризация` → `Высокая доступность`.

   Откроется экран просмотра свойств пары.

Отображаются следующие параметры:

| | |
|---|---|
| `Состояние пары` | Состояние пары высокой доступности. Возможны следующие варианты:<br>• `on` – статус отображается при нормальной работе пары. Высокая доступность обеспечивается.<br>• `failed` – статус отображается после попытки запуска пары, если пару сформировать не удалось. Синхронизация и обмен данными между узлами в паре невозможны. Высокая доступность не обеспечивается.<br>• `starting` – пара запускается. Статус отображается сразу после создания пары в процессе ее конфигурации и синхронизации.<br>• `not_reachable` – статус отображается, когда узел не получает ответ от соседнего узла по всем интерфейсам, участвующим в паре, в течение заданного таймаута (например из-за сетевой проблемы или недоступности соседнего узла). Если резервный узел фиксирует недоступность основного узла, выполняется переключение ролей: резервный узел становится основным. При этом для соседнего узла будет отображаться его последняя известная роль.<br>**Примечание**<br>Рекомендуется проверить соседний узел и, если требуется, восстановить его связность и работоспособность.<br>**Совет**<br>Чтобы убрать сообщение о смене статуса в паре, запустите ручную синхронизацию конфигураций по кнопке `Синхронизировать`. |
| `Последнее изменение` | Дата и время последнего изменения в паре, например смены ролей узлов. |
| `Узел` | Идентификация узла пары, на котором открыт веб-интерфейс Angie ADC:<br>• `Текущий` – узел, на котором в данный момент открыт веб-интерфейс Angie ADC со свойствами пары;<br>• `Соседний` – второй узел пары. |
| `IP-адрес узла` | IP-адрес узла, используемый для взаимодействия в паре.<br>**Предупреждение**<br>Не рекомендуется изменять настройки интерфейсов узлов, объединенных в пару высокой доступности, иначе пара может потерять связность и работоспособность. |
| `Роль узла` | Роль узла пары:<br>• `Основной` – обрабатывает трафик (находится в режиме Active).<br>• `Резервный` – находится в режиме ожидания (Standby). |

#### 4.1.4.2 Обновление конфигурации и синхронизация

После внесения изменений в конфигурацию основного узла необходимо вручную запустить синхронизацию пары, чтобы перенести внесенные изменения на резервный узел.

Чтобы обновить конфигурацию на узлах пары, выполните следующие действия:

1. Внесите необходимые изменения в конфигурацию основного узла.
2. Откройте веб-интерфейс Angie ADC на любом узле пары.

3. Перейдите `Высокая доступность и кластеризация` → `Высокая доступность`.

   Откроется экран просмотра свойств пары.

4. Нажмите `Синхронизировать`.

Конфигурация основного узла будет перенесена на резервный узел, на экране отобразится сообщение `Пара синхронизирована`. Если синхронизация завершится с ошибкой, на экране отобразится сообщение `Ошибка синхронизации`.

На резервном узле перезаписываются следующие файлы и папки:

- конфигурация балансировщика нагрузки;
- сертификаты и ключи балансировщика нагрузки;
- конфигурация маршрутизации (VRRP, BGP, OSPF);
- конфигурация GSLB;
- все файлы в папках `/etc/angie/crt` и `/etc/angie-lb/crt`;
- все файлы в папке `/var/transfer`.

#### 4.1.4.3 Настройка протокола опроса доступности узлов

Чтобы изменить настройки протокола опроса доступности узлов, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC на одном из узлов пары.
2. Перейдите `Высокая доступность и кластеризация` → `Высокая доступность`.

   Откроется экран просмотра узлов, входящих в пару.

3. В блоке `Проверка доступности узлов` укажите следующие данные:

   - `Интервал отправки запросов (мс)`

     Интервал отправки запросов резервным узлом для проверки доступности основного узла. По умолчанию запросы отправляются каждые 250 мс. Допустимый диапазон значений – от 200 до 1000 мс.

   - `Таймаут ожидания ответа (с)`

     Таймаут ожидания ответа от интерфейса основного узла. Если с момента получения последнего успешного ответа прошло больше заданного времени интерфейс будет считаться недоступным. При недоступности всех интерфейсов основного узла произойдет переключение ролей. По умолчанию – `3` секунды. Допустимый диапазон значений – от 1 до 3 секунд.

4. Нажмите `Сохранить`.

Новые настройки будут применены сразу.

#### 4.1.4.4 Удаление пары

> **Примечание**
>
> Удаление пары возможно только с основного узла.

Чтобы удалить пару, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC на узле, который является основным в паре на данный момент.
2. Перейдите `Высокая доступность и кластеризация` → `Высокая доступность`.

   Откроется экран просмотра свойств пары.

3. Нажмите `Расформировать` и подтвердите удаление.

Конфигурация пары высокой доступности будет удалена. Конфигурации узлов сохранятся, узлы продолжат функционировать независимо друг от друга.

### 4.1.5 Привязка клиентских сессий

В паре высокой доступности поддерживается привязка клиентских сессий (sticky sessions). При переключении узлов все запросы клиента в рамках одной сессии будут по-прежнему перенаправляться на один и тот же бэкенд-сервер, т.е. клиентские сессии будут сохраняться.

Информация о клиентских сессиях хранится в хранилище сессий Angie ADC. Привязка сессий реализуется с помощью директивы sticky (режим `learn` с `remote_action`). Для включения функции необходимо указать адрес хранилища сессий Angie ADC (`http://127.0.0.1:3380/sticky_session`) в конфигурации балансировщика нагрузки.

Пример конфигурации:

```
http {

    upstream sticky_test {

        server 127.0.0.1:8091 sid=web-server-01;
        server 127.0.0.1:8092 sid=web-server-02;

        sticky learn
            lookup=$http_x_userid
            remote_action=/create_session
            remote_result=$upstream_http_x_sid;
    }

    server {

        listen *:80;

        location / {

            proxy_pass http://sticky_test/;

        }

        location /create_session {

            internal;
            proxy_set_header X-Sticky-Sessid $sticky_sessid;
            proxy_set_header X-Sticky-Sid    $sticky_sid;
            proxy_set_header X-Sticky-Last   $msec;
            proxy_method POST;
            proxy_pass http://127.0.0.1:3380/sticky_session;
            proxy_connect_timeout 1s;
            proxy_read_timeout    1s;
        }
    }
}
```

## 4.2 IP-маршрутизация

Angie ADC поддерживает ручную настройку режимов высокой доступности на устройстве Angie ADC по протоколам BGP и OSPF (динамическая маршрутизация) и с VRRP в вариантах Active-Standby и Active-Active. Также поддерживается статическая маршрутизация и использование протокола BFD для уменьшения времени реакции на события и механизма RHI для оценки состояния апстримов.

В этом разделе:



**Примечание**

Рекомендуется резервировать все компоненты инфраструктуры. Отказ сопутствующей инфраструктуры (например выход из строя вышестоящего маршрутизатора или сбой серверов) выходит за рамки решения высокой доступности Angie ADC.

Пример отказоустойчивой инфраструктуры:

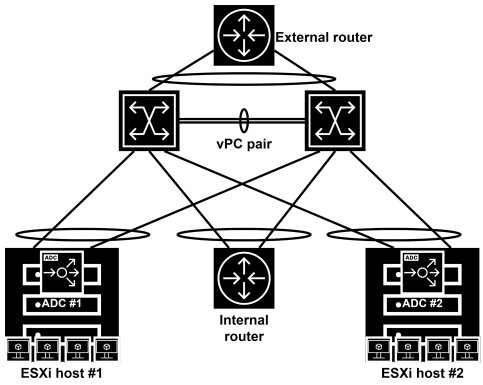



Рис. 8: Рис. 1

### 4.2.1 VRRP-маршрутизация

VRRP (Virtual Router Redundancy Protocol) — это протокол для виртуального резервирования устройств. VRRP позволяет переключиться на резервный Angie ADC, который автоматически принимает на себя IP-адрес и роль мастера, если основное устройство Angie ADC отказало. Таким образом обеспечивается отказоустойчивость для подключенных узлов в сети при статической маршрутизации. Поддерживаемые версии: VRRPv2 (для IPv4) и VRRPv3 (для IPv4 и IPv6).

**Примечание**

Настройка VRRP осуществляется через командную строку (CLI) на порту 2022.

**Важно**

Настройки VRRP не отображаются в веб-интерфейсе. Не рекомендуется одновременно использовать ручную настройку высокой доступности по VRRP и настройку VRRP *в рамках пары высокой доступности*.

В этом разделе:

- *Предварительная настройка для VMware ESXi*
- *Режим резервирования (Active-Standby)*
- *Режим распределения нагрузки (Active-Active)*
- *Команды VRRP*
- *Создание и удаление VRRP-групп*

#### 4.2.1.1 Предварительная настройка для VMware ESXi

Если Angie ADC развернут на VMware ESXi, то для работы VRRP необходимо дополнительно настроить сетевые политики безопасности ESXi и разрешить передачу трафика с портов, участвующих в VRRP.

##### Настройка VMware ESXi до версии 6.7

Для VMware ESXi версии до версии 6.7 необходимо создать отдельную порт-группу (Port Group) для портов, участвующих в VRRP, и включить для нее настройки **Promiscuous mode** и **Forged Transmits**.

1. Создайте отдельную порт-группу (Port Group) для портов, участвующих в VRRP.

   Пример пути для создания порт-группы: `Host` → `Configure` → `Networking` → `Port Groups` → `Add Port Group`.

2. Задайте следующие значения для передачи трафика при изменении MAC-адреса для новой порт-группы:

   - `Promiscuous Mode` – `Accept`.
   - `Forged transmits` – `Accept`.

   Пример пути для настройки порт-группы: `Host` → `Configure` → `Networking` → `Port Groups` → `<порт_группа>` → `Edit Settings` → `Security`.

3. Обеспечьте размещение виртуальных машин Angie ADC на разных ESXi-хостах кластера. Для этого запретите DRS-перемещение виртуальных машин Angie ADC на один хост с помощью запрещающего правила Anti-Affinity.

   Пример пути для настройки правила: `Host` → `Configure` → `Configuration` → `VM/Host Rules` → `Add`.

   В открывшемся окне заполните необходимые поля и укажите виртуальные машины Angie ADC.

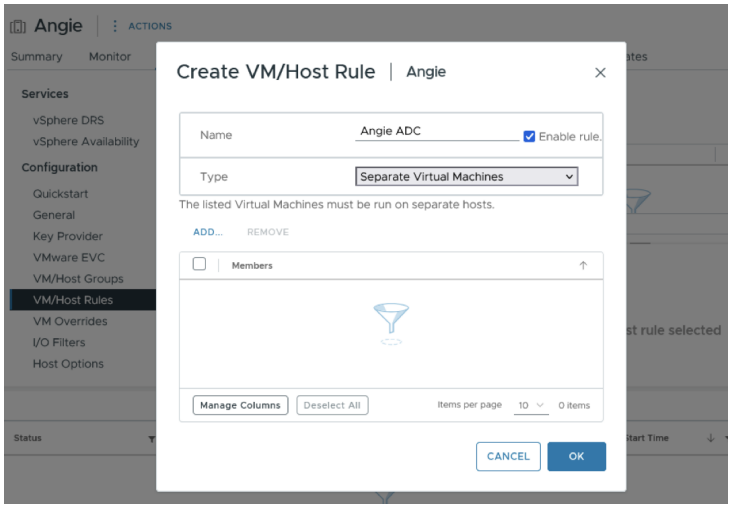



Рис. 1: создание запрещающего правила для ESXi до версии 6.7.

Общая схема будет такой: устройства Angie ADC подключены через отдельную порт-группу, на которой включен режим Promiscuous:

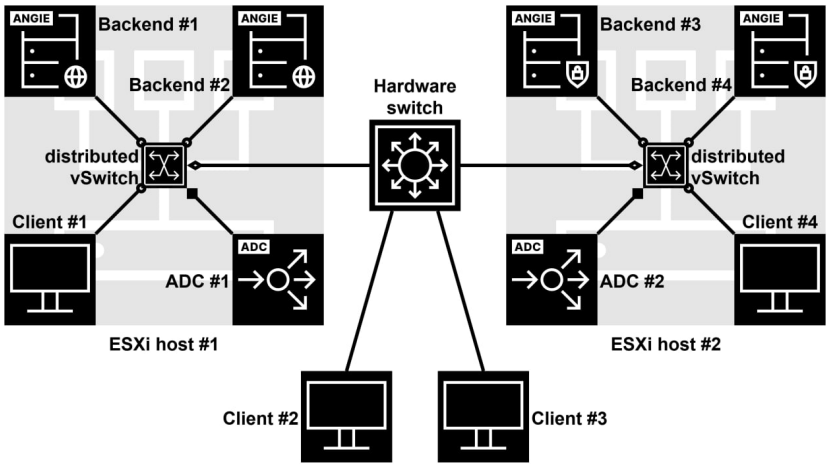



Рис. 2: подключение Angie ADC для ESXi до версии 6.7.

### Настройка VMware ESXi версии 6.7+

Для VMware ESXi версии 6.7+ необходимо настроить обычную порт-группу (Port Group) для портов, участвующих в VRRP, и включить для нее настройки **Forged Transmits** и **MAC Learning**.

1. Подключитесь к vCenter Server и перейдите `Inventory` → `Networks`.
2. Выберите DSwitch и щелкните правой кнопкой мыши по нужной порт-группе. В контекстном меню выберите пункт `Edit Settings` → `Security`.
3. Задайте следующие значения для передачи трафика при изменении MAC-адреса:
   - `Promiscuous Mode` – `Reject`.
   - `MAC address changes` – `Reject`.
   - `Forged transmits` – `Accept`.
4. В блоке `MAC Learning` установите `Status` в значение `Enabled`.

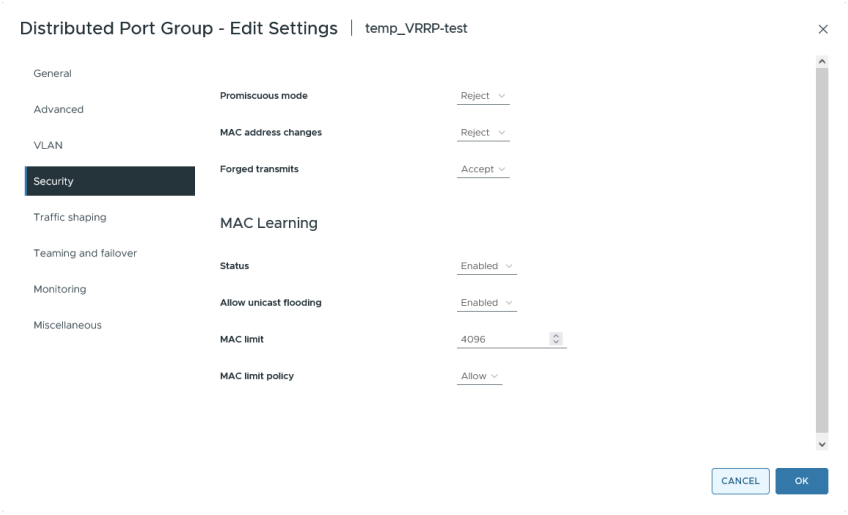



Рис. 3: настройка порт-группы на ESXi 6.7+.

Общая схема будет такой: устройства Angie ADC подключены через обычные порт-группы, при этом включена политика, разрешающая подмену MAC-адресов в Distributed Virtual Port Group (DVPG.macManagementPolicy=yes):

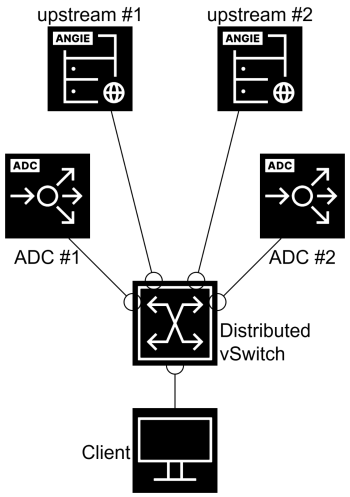



Рис. 4: подключение Angie ADC для ESXi версии 6.7+.

#### 4.2.1.2 Режим резервирования (Active-Standby)

В этом разделе описана настройка высокой доступности с использованием протокола VRRP в режиме резервирования (Active-Standby). Для настройки используется интерфейс командной строки (CLI) Angie ADC.

Также можно посмотреть следующие статьи:

- *Настройка режима распределения нагрузки (Active-Active) с помощью протокола VRRP*;
- *Особенности создания и удаления VRRP-групп*;
- *Команды VRRP*.

### Введение

Для обеспечения отказоустойчивости нужны две системы балансировки Angie ADC — основная и резервная. Резервная система Angie ADC будет принимать нагрузку при отказе основной. Автоматическое переключение между двумя системами балансировки Angie ADC можно реализовать с

помощью протокола VRRP. Для настройки VRRP используется командная строка Angie ADC CLI. Пример настройки приведен ниже.

> **Примечание**
>
> Резервирование других сетевых компонентов (маршрутизаторы, коммутаторы, межсетевые экраны) выходит за рамки настоящей статьи.

Пример типового участка сети с системами балансировки Angie ADC (ADC #1 и ADC #2):

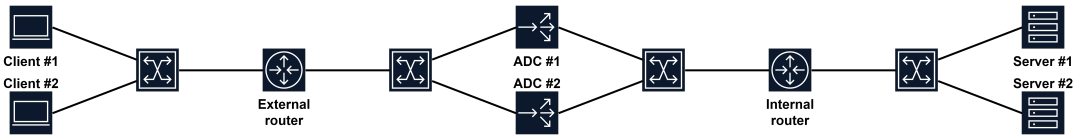



Рис. 1

Добавим *IP-адресацию*:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

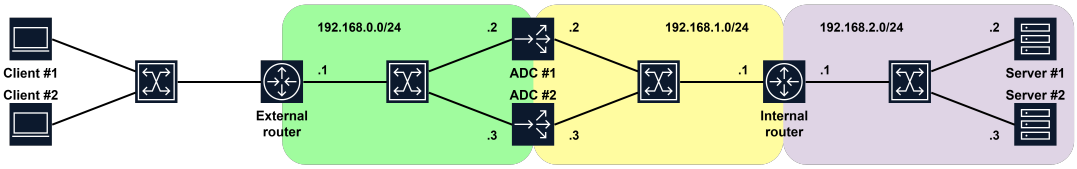



Рис. 2

*Адресация клиентов* не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть, они могут подключаться из любой сети, включая сеть интернет. Для этого на ADC #1 и ADC #2 необходимо указать адрес правого интерфейса маршрутизатора External router (`192.168.0.1`) в качестве шлюза по умолчанию.

*Виртуальный адрес*, который соответствует сервису, может быть размещен на loopback-интерфейсе каждой системы балансировки, либо это может быть виртуальный адрес, используемый протоколом VRRP. Первый способ будет более гибким.

При размещении адреса сервиса на loopback-интерфейсе (на схеме ниже это адрес `2.2.2.2`) необходимо добавить статический маршрут на этот адрес на маршрутизаторе External router. Причем в качестве next-hop необходимо использовать виртуальный адрес VRRP, за счет чего достигается отказоустойчивость решения.

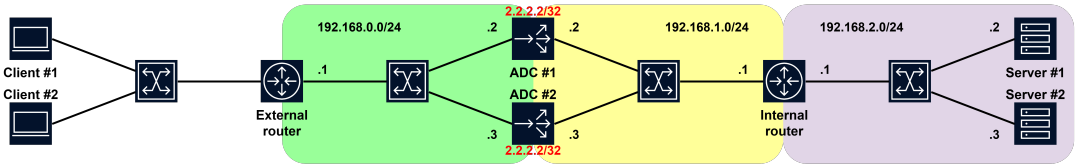



Рис. 3

При использовании виртуального адреса VRRP в качестве адреса сервиса (на схеме ниже это `192.168.0.10`) настройка соответствующих статических маршрутов на маршрутизаторе External router не требуется. Виртуальный адрес VRRP находится в одной подсети с интерфейсами маршрутизатора с точки зрения маршрутизатора External router.

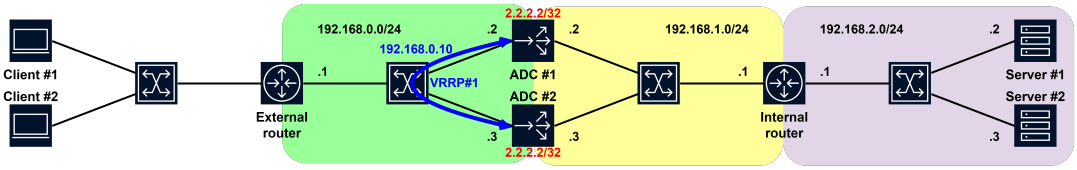



Рис. 4

Также допускается использование VRRP и при работе системы балансировки Angie ADC в «одноруком» режиме.

### Обработка типовых сбоев

Ниже описаны сценарии работы Angie ADC при сбоях.

### Полный отказ ADC #1

При полном отказе или отключении от сети ADC #1 происходят перевыборы VRRP-мастера. Переключение на резервную систему ADC #2 производится автоматически. Никакие дополнительные настройки не требуются.

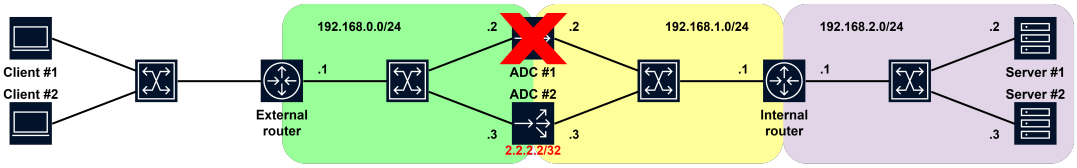



Рис. 5

### Отказ внешнего интерфейса ADC #1

Предлагаемая схема не предполагает возможности отслеживания частичных отказов и реагирования на них (пример ниже).

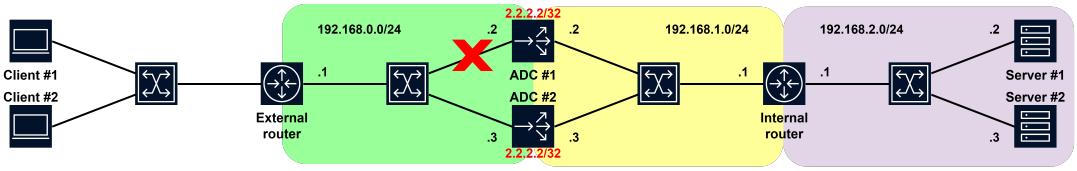



Рис. 6

В случае частичного отказа, как на схеме выше, ADC #1 перестает рассылать сообщения VRRP. ADC #2 начинает считать себя мастером, что заставляет виртуальную машину отвечать на ARP-запросы об адресе `192.168.0.10`, а также обрабатывать пользовательский трафик. Обратный трафик (от серверов в сторону клиентов) будет проходить через ADC #2, так как ADC #2 устанавливает сессии до серверов с адреса своего внутреннего интерфейса.

**Примечание**

IP transparency на данном этапе обеспечить невозможно. Для работы IP transparency требуется наличие маршрута по умолчанию на маршрутизаторе Internal router. Если прописать маршрут по умолчанию через ADC #1, то отказ внешнего интерфейса ADC #1 приведет к невозможности доставки трафика и его отбрасыванию (black hole). Решить эту проблему можно за счет использования протокола VRRP как на внешнем, так и на внутреннем интерфейсах. Причем состояние групп должно быть согласованным, то есть если для внешней сети мастером является система балансировки ADC #2, то и для внутренней сети ADC #2 также должна являться мастером. Добиться этого можно с помощью опции VRRP tracking, которая будет выпущена в ближайших релизах Angie ADC.

### Отказ линка между ADC #1 и Internal router

Отказ линка между ADC #1 и маршрутизатором Internal router более сложная ситуация: если ADC #1 сохраняет за собой роль мастера, то трафик от клиентов до серверов будет все еще передаваться через ADC #1, хотя реальные сервера для системы балансировки ADC #1 уже недостижимы. Таким образом, мы получаем классический пример black hole:

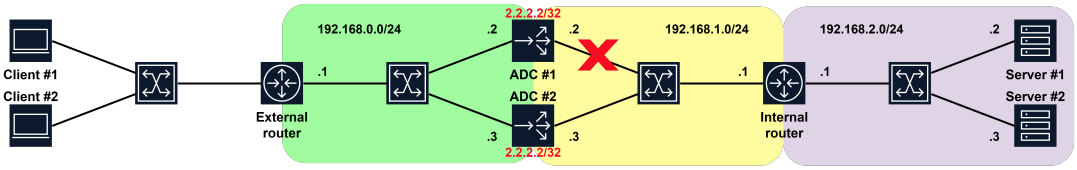



Рис. 7

Чтобы избежать этой проблемы, необходимо, чтобы ситуация отказа только одного из линков была невозможной, т.е. нужно обеспечить отказоустойчивое подключение к сети для хоста виртуализации, на котором работает ADC.

Рекомендованная схема подключения:

- Объединить в одну LAG-группу все физические интерфейсы хоста виртуализации и использовать этот виртуальный LAG-интерфейс в качестве транка, то есть передавать по нему тегированными несколько виртуальных сетей (один VLAN для external сети, второй VLAN для internal сети).
- Для резервирования коммутаторов можно использовать технологию vPC или стекирование, пример представлен на схеме ниже.

Из соображений простоты на схеме не отображено резервирование внешнего и внутреннего маршрутизаторов.

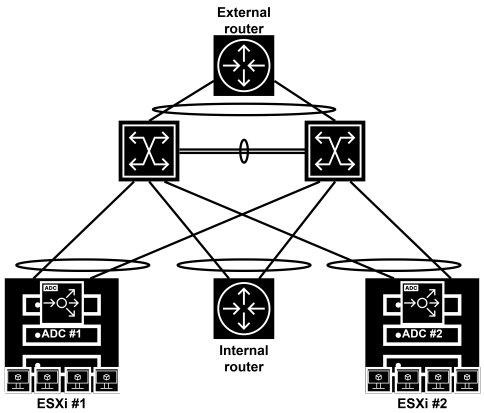



Рис. 8

**Примечание**

В ближайших релизах Angie ADC появится опция VRRP tracking, которая позволит автоматически переводить систему балансировки ADC #2 в режим VRRP backup при возникновении обсуждаемого отказа.

### Настройка VRRP

#### Базовая настройка

Для настройки VRRP необходимо указать виртуальный IP-адрес, все остальные настройки опциональны:

```
#! Задать виртуальный IP-адрес 192.168.0.10/24 для первой VRRP группы внешнего␣
↪интерфейса (ens33).
#! Задается для ADC #1 и для ADC #2.
# vrrp 1 int ens33 ip 192.168.0.10/24
```

Указанная выше настройка одинакова для ADC #1 и ADC #2.

**Примечание**

Полный список поддерживаемых команд VRRP можно посмотреть в *справочнике по командам VRRP*.

#### Изменение приоритета

Если требуется, чтобы система балансировки ADC #1 выигрывала выборы и принимала на себя роль мастера, то ее приоритет необходимо сделать выше, чем приоритет у ADC #2. Значение приоритета по умолчанию – 100. Пример настройки ADC #1 представлен ниже:

```
#! Задается для ADC #1
#vrrp 1 int ens33 priority 110
```

Аналогичная настройка делается для ADC #2, но с меньшим значением приоритета:

```
#! Задается для ADC #2
#vrrp 1 int ens33 priority 90
```

#### Использование preempt

Опция VRRP `preempt` включена по умолчанию. Эта опция заставляет включившуюся систему балансировки инициировать перевыборы мастера, если ее собственный приоритет выше приоритета текущего мастера.

Включить `preempt` можно с помощью следующей команды:

```
vrrp 1 int ens33 preempt
```

Выключение `preempt` реализовано через `no`:

```
no vrrp 1 int ens33 preempt
```

### Прописывание маршрута

Для указания статического маршрута, например на внешнем маршрутизаторе с указанием виртуального адреса VRRP в качестве next-hop, используется стандартная команда:

```
External(config)#ip route 2.2.2.2 255.255.255 192.168.0.10
```

### Проверка работы

Убедимся, что ADC #1 является мастером:

```
# show vrrp int ens33 1 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 110 1 0 Master Backup
```

Убедимся, что ADC #2 не обладает ролью мастера:

```
# show vrrp int ens33 1 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 90 1 0 Backup Backup
```

Отключим теперь систему балансировки ADC #1 и убедимся, что ADC #2 приняла на себя роль мастера:

```
# show vrrp int ens33 1 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 90 1 0 Master Backup
```

**Примечание**

В момент переезда роли Master на вторую систему балансировки ADC возможны кратковременные перерывы обработки трафика, вызванные задержками обнаружения падения активной ноды. Характерное время переключения ~3 секунды. Ускорить процедуру переключения можно за счет редактирования параметра `advertisement-interval` (см. *справочник по командам VRRP*). В будущем планируется использование протокола BFD для ускорения процедуры обнаружения отказов.

Заново включим систему балансировки ADC #1. Поведение устройства зависит от состояния опции VRRP `preempt`. Если опция `preempt` активирована (значение по умолчанию), то ADC #1 инициирует перевыборы и заберет на себя роль мастера. Если опция `preempt` отключена, то ADC #1 не будет пытаться отобрать роль мастера у системы балансировки ADC #2.

Проверить текущее состояние опции `preempt` можно с помощью следующей команды:

```
# show vrrp int ens33 1
Virtual Router ID 1
Protocol Version 3
Autoconfigured No
Shutdown No
Interface ens33
VRRP interface (v4) vrrp4-ens33-1
VRRP interface (v6) None
Primary IP (v4) 192.168.0.3
Primary IP (v6) ::
Virtual MAC (v4) 00:00:5e:00:01:01
```

```
Virtual MAC (v6) 00:00:5e:00:02:01
Status (v4) Master
Status (v6) Initialize
Priority 110
Effective Priority (v4) 110
Effective Priority (v6) 110
Preempt Mode No
Accept Mode Yes
Checksum with IPv4 Pseudoheader Yes
Advertisement Interval 1000 ms
Master Advertisement Interval (v4) Rx 1000 ms (stale)
Master Advertisement Interval (v6) Rx 0 ms (stale)
Advertisements Tx (v4) 1163
Advertisements Tx (v6) 0
Advertisements Rx (v4) 0
Advertisements Rx (v6) 0
Gratuitous ARP Tx (v4) 1
Neigh. Adverts Tx (v6) 0
State transitions (v4) 2
State transitions (v6) 0
Skew Time (v4) 600 ms
Skew Time (v6) 0 ms
Master Down Interval (v4) 3600 ms
Master Down Interval (v6) 0 ms
IPv4 Addresses 1
.............................. 192.168.0.10
IPv6 Addresses 0
```

Вне зависимости от того, какая система балансировки активна в данный момент, на маршрутизаторах в ARP-записи об адресе `192.168.0.10` будет один и тот же MAC-адрес: `00-00-5e-00-01-01` (при использовании первой VRRP-группы).

```
External#sho ip arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.0.10 - 0000.5e00.0101 ARPA GigabitEthernet0/0
```

Это означает, что при отказе одной из систем балансировки и аварийном переключении на вторую, единственное изменение, которое должно произойти в сети – переобучение коммутаторов, то есть потребуется актуализировать таблицы коммутации для адреса `00-00-5e-00-01-01`.

#### Заключение

С помощью протокола VRRP можно построить отказоустойчивое решение по балансировке нагрузки на сервера там, где использование протоколов динамической маршрутизации невозможно по той или иной причине.

Ряд дополнительных опций, расширяющих возможности протокола VRRP, находятся в данный момент в стадии разработки и будут доступны в ближайших релизах Angie ADC.

### 4.2.1.3 Режим распределения нагрузки (Active-Active)

В этом разделе описана настройка высокой доступности с использованием протокола VRRP в режиме распределения нагрузки (Active-Active). Для настройки используется интерфейс командной строки (CLI) Angie ADC.

Также можно посмотреть следующие статьи:

- *Настройка режима резервирования (Active-Standby) с помощью протокола VRRP*;
- *Особенности создания и удаления VRRP-групп*;

- *Команды VRRP*.

### Введение

Реализация полноценной схемы распределения нагрузки (Active-Active) не предусмотрена протоколом VRRP, но это ограничение можно обойти. Для этого рассмотрим два сценария подключения Angie ADC:

- между клиентами и системами балансировки установлен хотя бы один маршрутизатор;
- между клиентами и системами балансировки нет маршрутизатора.

### Подключение Angie ADC с маршрутизатором

Пример сети с маршрутизатором между клиентами и системами балансировки Angie ADC (ADC #1 и ADC #2):

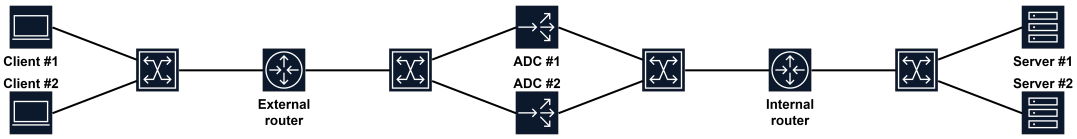



Рис. 1

Добавим *IP-адресацию*:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

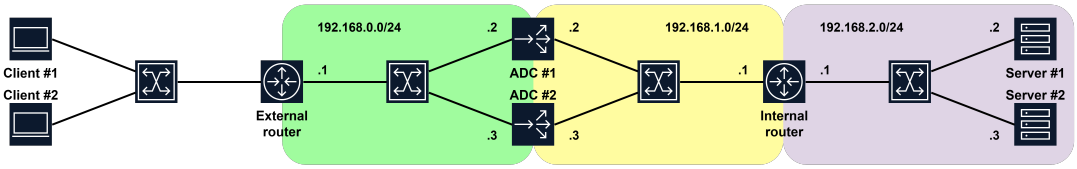



Рис. 2

*Адресация клиентов* не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть, они могут подключаться из любой сети, включая сеть интернет. Для этого на ADC #1 и ADC #2 необходимо указать адрес правого интерфейса маршрутизатора External router (`192.168.0.1`) в качестве шлюза по умолчанию.

*Виртуальный адрес*, который соответствует сервису, может быть размещен на loopback-интерфейсе каждой системы балансировки, либо это может быть виртуальный адрес, используемый протоколом VRRP.

- Первый способ (рис. 3 – сервис размещен на loopback-интерфейсах и IP-адреса этих интерфейсов совпадают) будет более гибким, так как позволяет использовать только один IP-адрес для сервиса.
- Во втором случае (используется виртуальный адрес протокола VRRP) необходимо сообщить клиенту, что сервис работает на двух адресах (например, с помощью DNS).

- В третьем случае, если сервис размещен на loopback-интерфейсах, но IP-адреса этих интерфейсов не совпадают, также потребуется сообщить клиенту, что сервис работает на двух адресах.

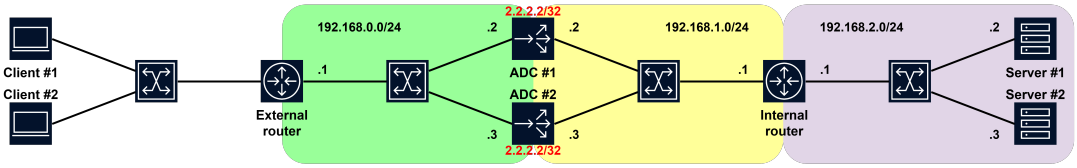



Рис. 3

При размещении адреса сервиса на loopback-интерфейсе (на схеме ниже это адрес `2.2.2.2`) необходимо добавить статический маршрут на этот адрес на маршрутизаторе External router. Причем в качестве next-hop необходимо использовать виртуальный адрес VRRP – `192.168.0.10`, за счет чего достигается отказоустойчивость решения.

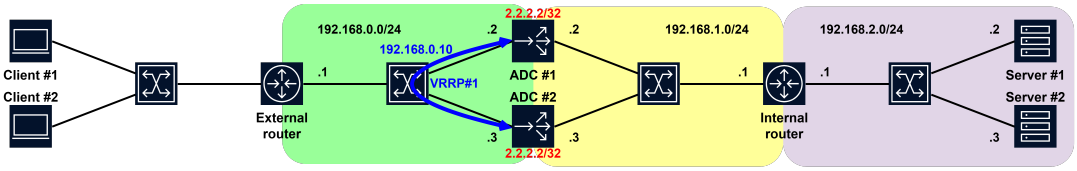



Рис. 4

### Настройка режима Active-Active

Чтобы перейти от режима работы Active-Standby к режиму Active-Active, необходимо на том же интерфейсе систем балансировки настроить вторую VRRP группу таким образом, чтобы мастером в ней являлась другая система Angie ADC (достигается за счет опции `priority`). Для второй группы должен быть настроен виртуальный IP-адрес, отличающийся от адреса первой группы.

На схеме ниже вторая группа имеет виртуальный IP-адрес `192.168.0.11`:

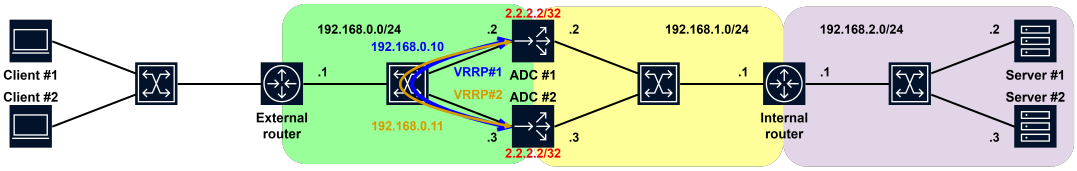



Рис. 5

На стороне маршрутизатора External router должно быть прописано два статических маршрута до префикса `2.2.2.2/32`:

- один с next-hop `192.168.0.10`;
- второй – `192.168.0.11`.

Маршрутизатор External router должен выполнять ECMP-балансировку трафика (часть сессий отправляется в сторону ADC #1, а оставшиеся – в сторону ADC #2).

Ниже приведен пример настройки маршрутизатора Cisco:

```
External(config)#ip route 2.2.2.2 255.255.255.255 192.168.0.10
External(config)#ip route 2.2.2.2 255.255.255.255 192.168.0.11
External(config)#exi
External#sho ip ro 2.2.2.2
Routing entry for 2.2.2.2/32
```

```
Known via "static", distance 1, metric 0
Routing Descriptor Blocks:
192.168.0.11
Route metric is 0, traffic share count is 1
* 192.168.0.10
Route metric is 0, traffic share count is 1
External#
```

У VRRP-группы есть виртуальный MAC-адрес (`00:00:5e:00:01:n`, где n – номер группы VRRP, номер виртуального маршрутизатора, в шестнадцатеричном виде). По MAC-адресу коммутатор определяет, для какой из систем балансировок предназначен трафик. Первая группа будет иметь виртуальный MAC-адрес `00:00:5e:00:01:01`, вторая – `00:00:5e:00:01:02`.

Убедиться в этом можно, просмотрев ARP-таблицу на маршрутизаторе External router:

```
External#sho ip arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 192.168.0.1 - ca01.0950.0008 ARPA GigabitEthernet0/0
Internet 192.168.0.10 0 0000.5e00.0101 ARPA GigabitEthernet0/0
Internet 192.168.0.11 0 0000.5e00.0102 ARPA GigabitEthernet0/0
External#
```

ARP-таблица маршрутизатора External router не меняется при отказе одной из систем балансировки нагрузки Angie ADC. Изменения будут на стороне коммутатора. Ниже приводится часть таблицы коммутации, когда обе системы балансировки «живы» и выполняют свои функции:

```
switch#sho mac address-table dynamic vlan 51
Mac Address Table
-------------------------------------------
Vlan Mac Address Type Ports
----------------------------
51  0000.5e00.0101 DYNAMIC Gi0/7
51  0000.5e00.0102 DYNAMIC Gi0/8
```

Система балансировки ADC #1 подключена к интерфейсу Gi0/7 коммутатора, а система балансировки ADC #2 – к интерфейсу Gi0/8.

При отказе одной из систем балансировки оставшаяся будет иметь обе VRRP-группы в статусе Master. С точки зрения коммутатора это будет выглядеть так, словно оба виртуальных MAC-адреса доступны через один интерфейс.

```
switch#sho mac address-table dynamic vlan 51
Mac Address Table
-------------------------------------------
Vlan Mac Address Type Ports
----------------------------
51 0000.5e00.0101 DYNAMIC Gi0/7
51 0000.5e00.0102 DYNAMIC Gi0/7
```

### Подключение Angie ADC без маршрутизатора

Пример схемы сети представлен на рисунке ниже.

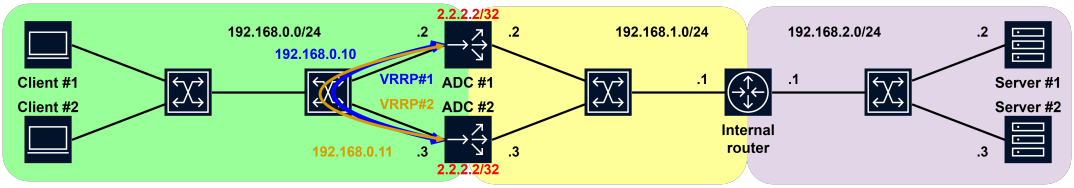


Рис. 6

В данном случае клиенты должны находиться в той же подсети, что и внешние интерфейсы систем балансировки Angie ADC (`192.168.0.0/24`).

Если клиенты Client #1 и Client #2 не имеют выхода в интернет, то тогда на них будет достаточно настроить один из адресов `192.168.0.10` и `192.168.0.11` в качестве шлюза по умолчанию. Тогда половина клиентов будет использовать адрес `192.168.0.10` в качестве шлюза по умолчанию, а вторая половина – `192.168.0.11`.

Если для выхода в интернет используется какое-то отдельное устройство, то на клиентах придется прописать статический маршрут до префикса `2.2.2.2/32`. На половине клиентов в качестве адреса следующего перехода (next-hop) необходимо настроить виртуальный адрес первой группы – `192.168.0.10`, а на оставшихся клиентах – адрес второй группы, `192.168.0.11`.

На схеме ниже желтой и красной стрелками указаны направления на шлюзы по умолчанию.

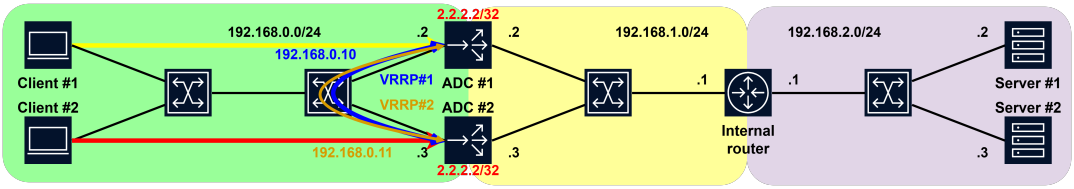



Рис. 7

Использовать ECMP-подход здесь нецелесообразно, так как некоторые клиентские операционные системы могут не поддерживать такую балансировку.

### Настройка VRRP

Настройка VRRP на системах балансировки не зависит от того, какая схема подключения используется, с маршрутизатором или без.

Для настройки VRRP необходимо:

- на каждой системе балансировки создать две группы;
- для каждой из групп указать виртуальный IP-адрес и приоритет.

Все остальные настройки опциональны.

**Примечание**

Полный список поддерживаемых команд VRRP можно посмотреть в *справочнике по командам VRRP*.

Пример настройки VRRP для ADC #1:

```
#! Задать виртуальный IP-адрес 192.168.0.10/24 для первой VRRP группы внешнего␣
↪интерфейса (ens33).
# vrrp 1 int ens33 ip 192.168.0.10/24
#! Задать приоритет 110 для первой VRRP группы внешнего интерфейса (ens33).
# vrrp 1 int ens33 priority 110
#! Задать виртуальный IP-адрес 192.168.0.11/24 для второй VRRP группы внешнего␣
↪интерфейса (ens33).
# vrrp 2 int ens33 ip 192.168.0.11/24
#! Задать приоритет 90 для второй VRRP группы внешнего интерфейса (ens33).
# vrrp 2 int ens33 priority 90
```

Настройка VRRP для ADC #2 симметричная:

```
#! Задать виртуальный IP-адрес 192.168.0.10/24 для первой VRRP группы внешнего␣
↪интерфейса (ens33).
# vrrp 1 int ens33 ip 192.168.0.10/24
#! Задать приоритет 90 для первой VRRP группы внешнего интерфейса (ens33).
# vrrp 1 int ens33 priority 90
#! Задать виртуальный IP-адрес 192.168.0.11/24 для второй VRRP группы внешнего␣
↪интерфейса (ens33).
# vrrp 2 int ens33 ip 192.168.0.11/24
#! Задать приоритет 110 для второй VRRP группы внешнего интерфейса (ens33).
# vrrp 2 int ens33 priority 110
```

### Проверка работы

Убедимся, что в стабильном и полностью рабочем состоянии ADC #1 является мастером для первой группы VRRP, а для второй группы мастером является система ADC #2.

Вывод с системы балансировки ADC #1:

```
# show vrrp int ens33 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 110 1 0 Master Backup
ens33 2 90 1 0 Backup Backup
```

Вывод с системы балансировки ADC #2:

```
# show vrrp int ens33 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 90   1 0 Backup Backup
ens33 2 110   1 0 Master Backup
```

При отказе, например, системы балансировки ADC #2, вывод информации о группах VRRP с системы балансировки ADC #1 будет выглядеть так:

```
# show vrrp int ens33 sum
Interface VRID Priority IPv4 IPv6 State (v4) State (v6)
------------------------------------------------------------------------
ens33 1 110 1 0 Master Backup
ens33 2 90 1 0 Master Backup
##
```

### Заключение

С помощью протокола VRRP можно построить отказоустойчивое решение по балансировке нагрузки на сервера там, где использование протоколов динамической маршрутизации невозможно по той или иной причине. Стоит отдельно отметить, что поддерживается работа систем балансировки как в режиме Active-Standby, так и в режиме Active-Active.

Ряд дополнительных опций, расширяющих возможности протокола VRRP, находятся сейчас в разработке и будут доступны в ближайших релизах Angie ADC.

#### 4.2.1.4 Команды VRRP

Здесь описываются команды VRRP, поддерживаемые в Angie ADC, и особенности их работы. Настройка VRRP осуществляется через командную строку.

Также вы можете посмотреть следующие статьи:

- *Настройка режима резервирования (Active-Standby) с помощью протокола VRRP*;
- *Настройка режима распределения нагрузки (Active-Active) с помощью протокола VRRP*;
- *Особенности создания и удаления VRRP-групп*.

Обобщенный пример команды VRRP:

```
vrrp n int ens33 command
```

Параметры:

- `n` - номер VRRP-группы (так же VRID — Virtual Router ID). Поддерживаются номера от 1 до 255.
- `ens33` - имя физического интерфейса. Если требуется создать VRRP-группы для другого физического интерфейса, то в команде необходимо указать имя этого интерфейса. VRRP можно независимо использовать для нескольких интерфейсов одновременно.
- `command` - команда VRRP (список поддерживаемых команд см. ниже).

### Команды настройки

| Команда | Описание | Значение по умолчанию |
|---|---|---|
| `vrrp n int <interface> ip <ip-address>` | Команда добавляет IPv4-адрес для Angie ADC. Добавление IP-адреса автоматически активирует его. Задается в формате `<ip_address>/<subnet_mask>`.<br>**Примечание**<br>Маска подсети IP-адреса должна соответствовать маске физического интерфейса. | n/a |
| `vrrp n int <interface> version v` | Команда задает версию протокола VRRP (`v` равно 2 или 3). Если версия протокола не задана, по умолчанию используется VRRPv3.<br>**Примечание**<br>Версии протокола VRRP на двух Angie ADC, работающих вместе, должны совпадать, иначе оба Angie ADC VA будут считать себя мастером. | 3 |
| `vrrp n int <interface> advertisement-interval i` | Команда устанавливает значение интервала отправки анонсов для VRRP-группы `n`. Значение задается в миллисекундах, но должно быть кратно 10, так как VRRP использует единицы измерения в сотых долях секунды (`i` от 10 до 40950 в мс). | 1000 мс (1 с) |
| `vrrp n int <interface> preempt` | Команда включает режим `preempt`, разрешающий запуск выборов нового мастера. При включенном режиме `preempt` резервный Angie ADC может автоматически стать мастером, если его приоритет выше, чем у текущего Angie ADC с ролью `master`. | Enabled (режим `preempt` включен) |
| `vrrp n int <interface> priority p` | Команда устанавливает приоритет `p` для устройств в VRRP-группе `n` на интерфейсе `<interface>` (`p` от 1 до 254). Значение приоритета `255` зарезервировано для текущего устройства в роли `master`, чей адрес интерфейса совпадает с виртуальным адресом группы. Значение `255` нельзя задать вручную. Устройство с наивысшим приоритетом становится `master`. Если у нескольких устройств в VRRP-группе одинаковый приоритет и режим preempt включен, мастером становится устройство с наибольшим основным IP-адресом. | 100 |
| `vrrp n int <interface> shutdown` | Команда выключает VRRP-группу `n` на интерфейсе `<interface>`. Используйте эту команду, если нужно завершить работу VRRP-группы без ее удаления. Чтобы обратно включить VRRP для этой группы, необходимо отменить текущую команду с помощью `no` (см. следующую таблицу). | Disabled (режим `shutdown` не активирован, группа включена) |
| `vrrp n int <interface> checksum-with-ipv4` | Команда включает использование псевдо-заголовка IPv4 при расчете контрольной суммы VRRPv3 для VRRP-группы `n`. Эта команда не влияет на VRRPv2 и IPv6. | Enabled (использование псевдо-заголовков IPv4 включено) |

### Команды удаления и отмены

| Команда | Описание |
|---|---|
| `no vrrp n int <interface> ip <ip-address>` | Команда удаляет IP-адрес, настроенный для VRRP-группы `n`. |
| `no vrrp n int <interface>` | Команда удаляет все пользовательские настройки, относящиеся к VRRP-группе `n` для интерфейса `<interface>`. Выставляются значения по умолчанию.<br>**Примечание**<br>Сама группа следом удаляется. |
| `no vrrp n int <interface> priority p` | Команда удаляет пользовательскую настройку приоритета для VRRP-группы `n`. |
| `no vrrp n int <interface> version v` | Команда удаляет версию протокола VRRP, настроенную для VRRP-группы `n`. Выставляется значение по умолчанию. |
| `no vrrp n int <interface> advertisement-interval i` | Команда удаляет значение интервала отправки анонсов, настроенное для VRRP-группы `n`. Выставляется значение по умолчанию. |
| `no vrrp n int <interface> shutdown` | Команда включает ранее выключенную VRRP-группу `n` на интерфейсе `<interface>`.<br>**Примечание**<br>После перезагрузки Angie ADC при включенном режиме `vrrp default shutdown` **все** VRRP-группы переводятся в выключенное состояние, в том числе те, которые были включены вручную. Чтобы заново включить конкретную группу после перезагрузки, введите последовательно следующие две команды:<br>`vrrp n int <interface> shutdown`<br>`no vrrp n int <interface> shutdown` |
| `no vrrp n int <interface> preempt` | Команда выключает режим `preempt` для группы `n`, выборы не будут запускаться даже при более высоком приоритете группы `n`. |
| `no vrrp n int <interface> checksum-with-ipv4-pseudo` | Команда выключает использование псевдо-заголовка IPv4 при расчете контрольной суммы VRRPv3 для VRRP-группы `n`. |

### Дефолтные команды для настройки и отмены

| Команда | Описание | Значение по умолчанию |
|---|---|---|
| `vrrp default advertisement-inte` | Команда устанавливает значение интервала отправки анонсов для всех новых VRRP-групп при их создании. Значение задается в миллисекундах, но должно быть кратно 10, так как VRRP использует единицы измерения в сотых долях секунды (`i` от 10 до 40950 в мс). | 1000 мс (1 с) |
| `vrrp default shutdown` | Команда устанавливает значение "выключено" для всех новых VRRP-групп при их создании, то есть переводит их в выключенное состояние сразу после загрузки или применения конфигурации. Режим shutdown можно выключить глобально (`no vrrp default shutdown`).<br>**Примечание**<br>После перезагрузки Angie ADC при включенном режиме `vrrp default shutdown` **все** VRRP-группы переводятся в выключенное состояние, в том числе те, которые были включены вручную. | Enabled (режим `default shutdown` включен, новые группы выключены) |
| `vrrp default checksum-with-ipv4` | Команда включает использование псевдо-заголовка IPv4 при расчете контрольной суммы VRRPv3 для всех новых VRRP-групп при их создании. Эта команда не влияет на VRRPv2 и IPv6. | Enabled (дефолтное использование включено) |
| `no vrrp default advertisement-inte` | Команда удаляет пользовательское значение интервала отправки анонсов, настроенное для всех новых групп VRRP. Выставляется значение по умолчанию 1000 мс (1 сек.). | n/a |
| `no vrrp default shutdown` | Команда отключает настройку, по которой все новые VRRP-группы создаются выключенными. Новые VRRP-группы будут создаваться сразу включенными. | n/a |
| `no vrrp default checksum-with-ipv4` | Команда отключает настройку, по которой для всех новых VRRP-групп при расчете контрольной суммы используется псевдо-заголовок IPv4. | n/a |

### Команды для просмотра информации

| Команда | Описание |
|---|---|
| `show vrrp` | Команда выводит информацию о состоянии всех VRRP-групп для всех интерфейсов. |
| `show vrrp int <interface>` | Команда выводит информацию о всех VRRP-группах для интерфейса `<interface>`. |
| `show vrrp int <interface> n`<br>или<br>`show vrrp n int <interface>` | Команда выводит информацию о VRRP-группе `n` для интерфейса `<interface>` (`n` от 1 до 255). |
| `show vrrp int <interface> n sum` | Команда выводит краткую информацию о VRRP-группе `n` для интерфейса `<interface>` (`n` от 1 до 255). |

#### 4.2.1.5 Создание и удаление VRRP-групп

Здесь описываются особенности создания и удаления VRRP-групп. Настройка VRRP осуществляется через интерфейс командной строки (CLI) Angie ADC.

Также можно посмотреть следующие статьи:

- *Настройка режима резервирования (Active-Standby) с помощью протокола VRRP*;
- *Настройка режима распределения нагрузки (Active-Active) с помощью протокола VRRP*;
- *Справочник по VRRP-командам*.

VRRP-группа — это логическая сущность, привязанная к физическому интерфейсу или его подынтерфейсу. Создание и удаление VRRP-групп вручную в Angie ADC не требуется. VRRP-группы создаются автоматически при введении команд, изменяющих конфигурацию по умолчанию, и удаляются также автоматически, если для группы не остается параметров, значения которых отличаются от значений по умолчанию. Это позволяет сохранять минимальную конфигурацию VRRP по умолчанию.

Вы можете просмотреть список всех VRRP-групп с помощью команды `show vrrp`.

Ниже приведены примеры ситуаций, в которых VRRP-группа будет создана или удалена.

##### Создание VRRP-группы

Новая VRRP-группа создается автоматически, если изменяется значение по умолчанию любого из ее параметров.

Например, если для VRRP-группы `10`, привязанной к интерфейсу `ens33`, указать IP-адрес (по умолчанию не указан), то указанная группа будет создана автоматически:

```
vrrp 10 int ens33 ip 192.168.0.1/24
```

Если для VRRP-группы `11`, привязанной к интерфейсу `ens33`, указать значение параметра `preempt` равным `disabled` (по умолчанию `enabled`), то указанная группа также будет создана автоматически:

```
no vrrp 11 int ens33 preempt
```

##### Удаление VRRP-группы

Рассмотрим пример. Для интерфейса `ens33` создано две VRRP-группы, с параметром `preempt` со значением `disabled` (выключено):

```
no vrrp 10 int ens33 preempt
```

```
no vrrp 11 int ens33 preempt
```

Группа удаляется в следующих случаях:

- Администратор ввел команду, возвращающую все параметры группы к значениям по умолчанию, в результате чего группа удаляется. Пример удаления группы `10`:

  ```
  no vrrp 10 int ens33
  ```

- Администратор вернул параметр группы к значению по умолчанию, при этом другие параметры группы уже имеют значения по умолчанию. Группа будет автоматически удалена, так как ее настройка больше не содержит ни одного параметра, значение которого отличается от значения по умолчанию.

Пример удаления группы `11` (параметру `preempt` присваивается значение по умолчанию `enabled`):

```
vrrp 11 int ens33 preempt
```

### 4.2.2 BGP-маршрутизация

В этом разделе описана настройка BGP-маршрутизации для обеспечения высокой доступности в режиме резервирования (Active-Standby) и в режиме распределения нагрузки (Active-Active). Для настройки используется интерфейс командной строки (CLI) Angie ADC.

В этом разделе:



#### 4.2.2.1 Резервирование с помощью протокола BGP (Active-Standby)

##### Введение

В этом разделе описана настройка высокой доступности с использованием протокола BGP в режиме резервирования (Active-Standby). Для настройки используется интерфейс командной строки (CLI) Angie ADC.

В качестве протокола динамической маршрутизации используется BGPv4 для адресного семейства IPv4 unicast. Отказоустойчивость в рамках IPv6 unicast настраивается аналогично.

В рассматриваемом примере Angie ADC #1 выполняет функции активной системы балансировки трафика, а Angie ADC #2 остается в резерве.

Пример типового участка сети с системами балансировки Angie ADC (ADC #1 и ADC #2) представлен на схеме:

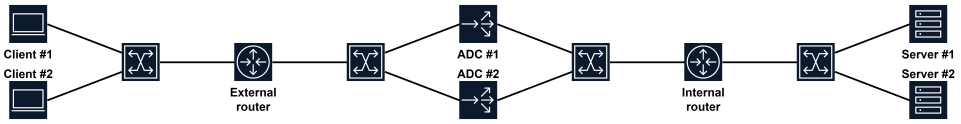



Рис. 1

Добавим IP-адресацию:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

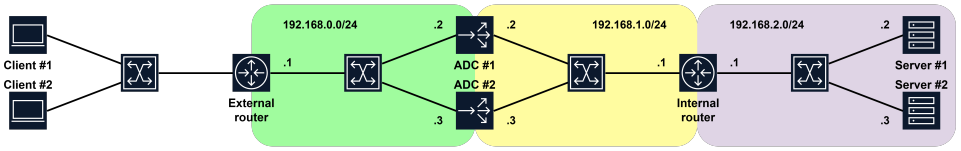



Рис. 2

Адресация клиентов не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть они могут подключаться из любой сети, включая сеть интернет.

Для этих целей External router будет анонсировать маршрут по умолчанию по BGP для ADC #1 и ADC #2. В данном документе будем использовать только одну автономную систему BGP №65001.

> **Примечание**
>
> Автономная система BGP — логическое объединение устройств в сети для маршрутизации через протокол BGP.

### BGP-сессии

Устанавливаемые BGP-сессии представлены на схеме:

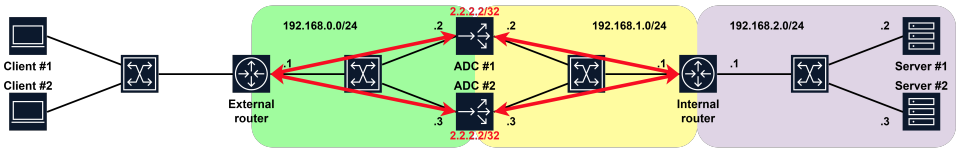



Рис. 3

Красным цветом на схеме отмечены IP-адреса виртуальных серверов. Это те адреса, к которым подключаются клиенты снаружи. Системы балансировки должны анонсировать маршруты до этих виртуальных серверов в сторону External router, причем маршрут через ADC #1 должен быть более предпочтительным. Добиться этого можно разными способами, манипулируя атрибутами пути в BGP.

Так как все BGP-взаимодействие производится строго в рамках одной автономной системы, мы будем использовать атрибут Local preference (чем больше, тем лучше) для выбора наиболее оптимального пути. Трафик от серверов в сторону клиента должен проходить через ту же систему балансировки, что и трафик от клиентов к серверам. Добиться этого можно также путем манипулирования значением атрибута Local preference для префикса `0.0.0.0/0`.

### Обработка типовых сбоев

Ниже описаны сценарии работы Angie ADC при сбоях.

### Полный отказ ADC #1

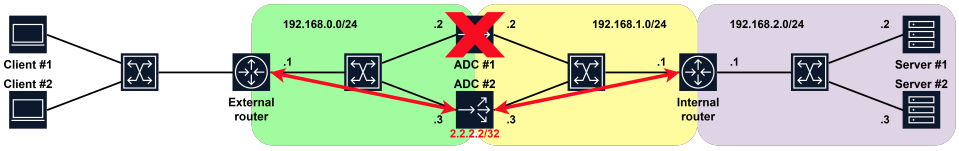



Рис. 4

При полном отказе или отключении от сети ADC #1 разрываются обе BGP-сессии с ADC #1. Через ADC #1 не проходят никакие объявления маршрутов. Остается связь по BGP только с ADC #2. Переключение на резервную систему ADC #2 производится автоматически. Никакие дополнительные настройки не требуются.

### Отказ внешнего интерфейса ADC #1

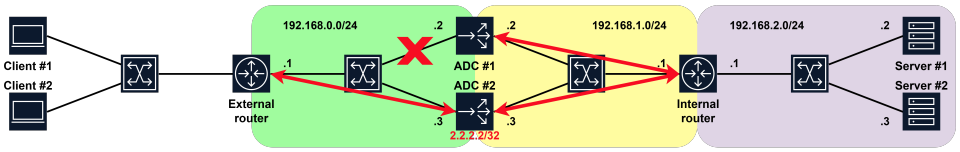



Рис. 5

В этом случае разрывается BGP-сессия между ADC #1 и External router. ADC #1 теперь не может объявить маршрут до адреса виртуального сервера в сторону External router. Несмотря на наличие BGP-сессии между ADC #1 и Internal router, ADC #1 не отправляет маршрут по умолчанию в сторону Internal router, так как сама не получает его от External router из-за сбоя линка. Таким образом, Internal router видит маршрут по умолчанию только через ADC #2, а External router получает маршрут до адреса виртуального сервера только через полностью работающий ADC #2.

### Отказ линка между ADC #1 и Internal router

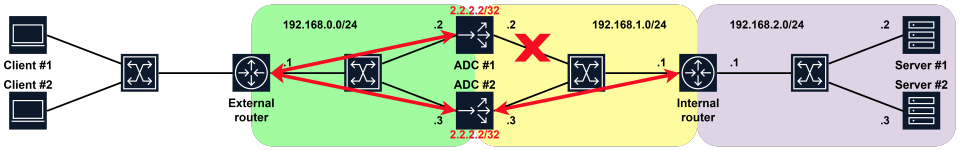



Рис. 6

Если BGP-сессия между External router и ADC #1 сохраняется, то ADC #1 будет продолжать анонсировать маршрут до адреса виртуального сервера, хотя реальные серверы для ADC #1 уже недостижимы. Таким образом, мы получаем классический black hole для трафика. Чтобы избежать этого, анонсирование маршрутов до виртуальных серверов должно быть условным, то есть ADC #1 анонсирует маршрут до адреса виртуального сервера, только если в BGP-таблице присутствуют префиксы, соответствующие подсетям реальных серверов.

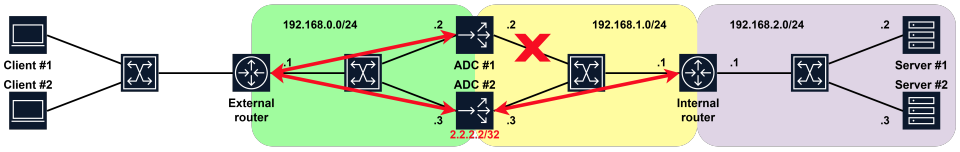



Рис. 7

### Настройка маршрутизации

Приведем ключевые настройки маршрутизации для External router, Internal router, ADC #1 и ADC #2.

### Ключевые настройки маршрутизации для External router

```
hostname external
interface GigabitEthernet0/0
 ip address 192.168.0.1 255.255.255.0
!Интерфейс в сторону систем балансировки
router bgp 65001
 neighbor 192.168.0.2 remote-as 65001
!Сессия с ADC#1
```

```
 neighbor 192.168.0.2 default-originate
!Анонсирование маршрута по умолчанию
 neighbor 192.168.0.3 remote-as 65001
!Сессия с ADC#2
 neighbor 192.168.0.3 default-originate
!Анонсирование маршрута по умолчанию
end
```

### Ключевые настройки маршрутизации для Internal router

```
hostname internal
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
!Интерфейс в сторону систем балансировки
interface GigabitEthernet1/0
ip address 192.168.2.1 255.255.255.0
!Интерфейс в сторону подсети с реальными серверами
router bgp 65001
 network 192.168.2.0 mask 255.255.255.0
!Добавление информации о подсети с реальными серверами в BGP
 neighbor 192.168.1.2 remote-as 65001
!Сессия с ADC#1
 neighbor 192.168.1.3 remote-as 65001
!Сессия с ADC#2
end
```

### Ключевые настройки маршрутизации Angie ADC #1

```
hostname adc1
!Имя устройства
ip prefix-list vip seq 5 permit 2.2.2.2/32
!Префикс-лист, описывающий адреса виртуальных серверов
ip prefix-list servers seq 5 permit 192.168.2.0/24
!Префикс-лист, описывающий адреса реальных серверов.
interface ens37
 ip address 192.168.0.2/24
!Интерфейс в сторону External router
exit
interface ens38
 ip address 192.168.1.2/24
!Интерфейс в сторону Internal router
exit
interface lo
 ip address 2.2.2.2/32
!Loopback-интерфейс для адресов виртуальных серверов.
exit
router bgp 65001
 neighbor 192.168.0.1 remote-as 65001
!Сессия с External router
 neighbor 192.168.1.1 remote-as 65001
!Сессия с Internal router
address-family ipv4 unicast
  network 2.2.2.2/32
!Добавление префикса для адреса виртуального сервера в BGP
  neighbor 192.168.0.1 route-reflector-client
!Разрешаем передачу префиксов между двумя iBGP-клиентами
```

```
  neighbor 192.168.0.1 route-map fromclients in
!Манипуляции над маршрутами, полученными от External router, меняем local preference
  neighbor 192.168.0.1 route-map 2clients out
!Манипуляции над маршрутами, отправляемыми в сторону External router, меняем local␣
↪preference
  neighbor 192.168.0.1 advertise-map 2clients exist-map servers
!Условная отправка префиксов: если префикс есть в route-map servers, то будет отправка
  neighbor 192.168.1.1 route-reflector-client
!Разрешаем передачу префиксов между двумя iBGP-клиентами
  neighbor 192.168.1.1 next-hop-self force
!Подменяем адрес атрибута next-hop на собственный даже для отраженных префиксов
 exit-address-family
exit
route-map 2clients permit 10
 match ip address prefix-list vip
 set local-preference 150
!Для префиксов, соответствующих адресам виртуальных серверов, увеличиваем Local␣
↪preference
exit
route-map 2clients deny 20
!Блокируем отправку всех остальных префиксов
exit
route-map fromclients permit 10
 set local-preference 150
!Увеличиваем Local preference для маршрутов, получаемых от External router
exit
route-map servers permit 10
 match ip address prefix-list servers
!Условие для условной отправки: если в BGP-таблице есть маршрут,
!соответствующий префикс-листу servers, то условие считается выполненным
exit
end
```

### Ключевые настройки маршрутизации Angie ADC #2

```
hostname adc2
!Имя устройства
ip prefix-list vip seq 5 permit 2.2.2.2/32
!Префикс-лист, описывающий адреса виртуальных серверов
ip prefix-list servers seq 5 permit 192.168.2.0/24
!Префикс-лист, описывающий адреса реальных серверов.
interface ens37
 ip address 192.168.0.3/24
!Интерфейс в сторону External router
exit
interface ens38
 ip address 192.168.1.3/24
!Интерфейс в сторону Internal router
exit
interface lo
 ip address 2.2.2.2/32
!Loopback-интерфейс для адресов виртуальных серверов.
exit
router bgp 65001
 neighbor 192.168.0.1 remote-as 65001
!Сессия с External router
 neighbor 192.168.1.1 remote-as 65001
```

```
!Сессия с Internal router
address-family ipv4 unicast
  network 2.2.2.2/32
!Добавление префикса для адреса виртуального сервера в BGP
  neighbor 192.168.0.1 route-reflector-client
!Разрешаем передачу префиксов между двумя iBGP-клиентами
  neighbor 192.168.0.1 route-map fromclients in
!Манипуляции над маршрутами, полученными от External router, меняем local preference
  neighbor 192.168.0.1 route-map 2clients out
!Манипуляции над маршрутами, отправляемыми в сторону External router, меняем local␣
↪preference
  neighbor 192.168.0.1 advertise-map 2clients exist-map servers
!Условная отправка префиксов: если префикс есть в route-map servers, то будет отправка
  neighbor 192.168.1.1 route-reflector-client
!Разрешаем передачу префиксов между двумя iBGP-клиентами
  neighbor 192.168.1.1 next-hop-self force
!Подменяем адрес атрибута next-hop на собственный даже для отраженных префиксов
 exit-address-family
exit
route-map 2clients permit 10
 match ip address prefix-list vip
 set local-preference 50
!Для префиксов, соответствующих адресам виртуальных серверов, уменьшаем Local␣
↪preference
exit
route-map 2clients deny 20
!Блокируем отправку всех остальных префиксов
exit
route-map fromclients permit 10
 set local-preference 50
!Уменьшаем Local preference для маршрутов, получаемых от External router
exit
route-map servers permit 10
 match ip address prefix-list servers
!Условие для условной отправки: если в BGP-таблице есть маршрут,
!соответствующий префикс-листу servers, то условие считается выполненным
exit
end
```

### Проверка работы

Выполним проверку работы системы маршрутизации, когда ADC #1 и ADC #2 находятся в полностью работоспособном состоянии.

> **Примечание**
>
> В листингах ниже отсутствуют префиксы, не относящиеся к обсуждаемым вопросам отказоустойчивости.

Маршрутизатор External router видит маршрут к адресу виртуального сервера через обе системы балансировки, однако маршрут через ADC #1 является более предпочтительным:

```
external#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```

```
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

      2.0.0.0/32 is subnetted, 1 subnets
B        2.2.2.2 [200/0] via 192.168.0.2, 02:06:11
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet0/0
L        192.168.0.1/32 is directly connected, GigabitEthernet0/0
external#sho ip bg
BGP table version is 9, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
     0.0.0.0          0.0.0.0                                0 i
 *>i 2.2.2.2/32       192.168.0.2              0    150      0 i
 * i                  192.168.0.3              0     50      0 i
```

Маршрутизатор Internal router видит маршрут по умолчанию через обе системы балансировки, однако маршрут через ADC #1 является более предпочтительным:

```
internal#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

B*    0.0.0.0/0 [200/0] via 192.168.1.2, 02:12:37
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.2.0/24 is directly connected, GigabitEthernet1/0
L        192.168.2.1/32 is directly connected, GigabitEthernet1/0
internal#sho ip bgp
BGP table version is 12, local router ID is 33.33.33.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          192.168.1.2              0    150      0 i
```

```
 * i                  192.168.1.3              0     50      0 i
 *>  192.168.2.0/24   0.0.0.0                  0         32768 i
```

Отключим теперь на ADC #1 интерфейс в сторону Internal router и проверим маршрутизацию:

```
adc1# sho int bri
Interface       Status  VRF             Addresses
---------       ------  ---             ---------
ens37           up      default         192.168.0.2/24
ens38           down    default         192.168.1.2/24
lo              up      default         2.2.2.2/32
adc1# conf t
adc1(config)# int ens38
adc1(config-if)# shut
adc1# sho int bri
Interface       Status  VRF             Addresses
---------       ------  ---             ---------
ens37           up      default         192.168.0.2/24
ens38           down    default         192.168.1.2/24
lo              up      default         2.2.2.2/32
```

Изучим теперь таблицу маршрутизации и BGP-таблицу на маршрутизаторе External router. Несмотря на наличие BGP-сессии до каждой из систем балансировки, только по одной сессии прилетает маршрут на адрес виртуального сервера – от ADC #2:

```
external#sho ip bg su
BGP router identifier 11.11.11.11, local AS number 65001
BGP table version is 8, main routing table version 8
2 network entries using 296 bytes of memory
2 path entries using 128 bytes of memory
2/1 BGP path/bestpath attribute entries using 272 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 696 total bytes of memory
BGP activity 3/1 prefixes, 4/2 paths, scan interval 60 secs
Neighbor        V           AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/
↪PfxRcd
192.168.0.2     4            1     156     150        8    0    0 00:07:29        0
192.168.0.3     4            1     157     150        8    0    0 00:07:29        1
external#sho ip bg
BGP table version is 8, local router ID is 11.11.11.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
     0.0.0.0          0.0.0.0                              0 i
 *>i 2.2.2.2/32       192.168.0.3              0     50      0 i
external#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
```

```
       + - replicated route, % - next hop override
Gateway of last resort is not set
      2.0.0.0/32 is subnetted, 1 subnets
B        2.2.2.2 [200/0] via 192.168.0.3, 00:04:36
      192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.0.0/24 is directly connected, GigabitEthernet0/0
L        192.168.0.1/32 is directly connected, GigabitEthernet0/0
```

И теперь маршрутная информация со стороны Internal router:

```
internal#sho ip bgp
BGP table version is 8, local router ID is 33.33.33.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
              x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

     Network          Next Hop            Metric LocPrf Weight Path
 *>i 0.0.0.0          192.168.1.3              0     50      0 i
 *>  192.168.2.0/24   0.0.0.0                  0         32768 i
internal#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is 192.168.1.3 to network 0.0.0.0
B*    0.0.0.0/0 [200/0] via 192.168.1.3, 00:00:49
C        192.168.1.0/24 is directly connected, GigabitEthernet0/0
L        192.168.1.1/32 is directly connected, GigabitEthernet0/0
C        192.168.2.0/24 is directly connected, GigabitEthernet1/0
L        192.168.2.1/32 is directly connected, GigabitEthernet1/0
```

#### Применение с IP transparency

Все описанное выше относится к режиму работы Angie ADC с включенной опцией IP transparency. При включенной опции IP Transparency Angie ADC сохраняет исходный адрес отправителя в IP-пакетах, позволяя серверам на сетевом уровне различать клиентские подключения.

Очень часто администраторы систем балансировки отключают эту опцию, в этом случае Angie ADC будет выполнять не только dNAT, но sNAT над трафиком, приходящим от клиентов. Если система балансировки Angie ADC выполняет sNAT, то в этом случае маршрутизация несколько упрощается, так как в этом случае нет необходимости анонсировать на Internal router маршрут по умолчанию. Если ADC #1 и ADC #2 выполняют sNAT в разные адреса, то в этом случае нет необходимости заботиться о том, с какими значениями Local preference (и иными атрибутами пути) маршруты до соответствующих адресов попадают на Internal router.

### 4.2.2.2 Распределение нагрузки с помощью протокола BGP (Active-Active)

#### Введение

В этом разделе описана настройка высокой доступности с использованием протокола BGP в режиме распределения нагрузки (Active-Active). Для настройки используется командная строка Angie ADC CLI.

В качестве протокола динамической маршрутизации используется BGPv4 для адресного семейства IPv4 unicast. Отказоустойчивость в рамках IPv6 unicast настраивается аналогично.

В рассматриваемом примере устройства Angie ADC #1 и Angie ADC #2 выполняют функции активной системы балансировки трафика одновременно. Для динамического исключения одного из Angie ADC из работы в случае сбоя (технические проблемы, потери связности и так далее) можно использовать *механизм RHI (Route Health Injection)*.

Пример типового участка сети с системами балансировки Angie ADC (ADC #1 и ADC #2) представлен на схеме:

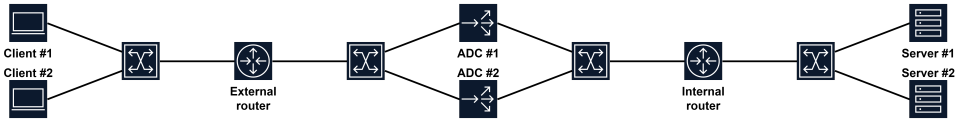



Рис. 1

Добавим IP-адресацию:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

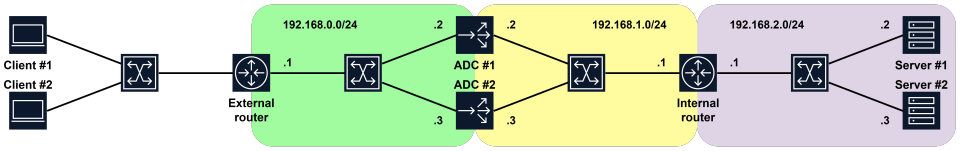



Рис. 2

Адресация клиентов не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть они могут подключаться из любой сети, включая сеть интернет. Для этих целей External router будет анонсировать маршрут по умолчанию по BGP для ADC #1 и ADC #2.

### BGP-сессии

Устанавливаемые BGP-сессии представлены на схеме:

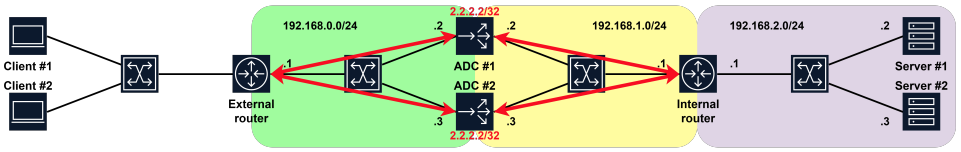



Рис. 3

Красным цветом на схеме отмечены IP-адреса виртуальных серверов. Это те адреса, к которым подключаются клиенты снаружи. ADC #1 и ADC #2 должны анонсировать маршруты до этих виртуальных серверов в сторону External router, причем маршруты через ADC #1 и ADC #2 должны иметь схожие атрибуты пути, чтобы маршрутизатор External router мог выполнить распределение трафика между ними.

Трафик внутри пользовательской сессии от серверов должен возвращаться строго через ту же систему балансировки, через которую он проходил до этого. Добиться этого можно с помощью механизма *SNAT*, когда адрес клиента подменяется на адрес интерфейса системы балансировки.

### Обработка типовых сбоев

Ниже описаны сценарии работы Angie ADC при сбоях.

### Полный отказ ADC #1

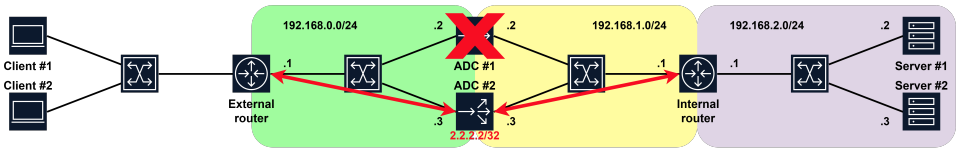



Рис. 4

При полном отказе или отключении от сети ADC #1 разрываются обе BGP-сессии с ADC #1. Через ADC #1 не проходят никакие объявления маршрутов. Остается связь по BGP только с ADC #2. Отключение ADC #1 происходит автоматически. Никакие дополнительные действия не требуются.

### Отказ внешнего интерфейса ADC #1

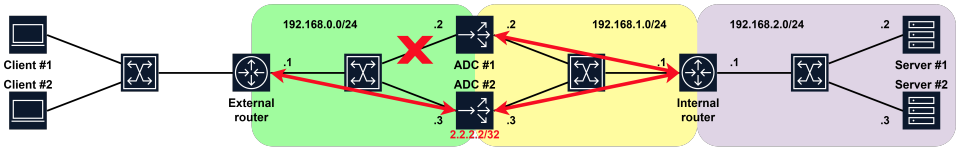



Рис. 5

В этом случае разрывается BGP-сессия между ADC #1 и External router. ADC #1 теперь не может объявить маршрут до адреса виртуального сервера в сторону External router, а пользовательский трафик будет распределяться только на систему балансировки ADC #2. Так как клиентские сессии устанавливаются снаружи, трафик всех вновь устанавливаемых сессий будет проходить только через ADC #2.

### Отказ линка между ADC #1 и Internal router

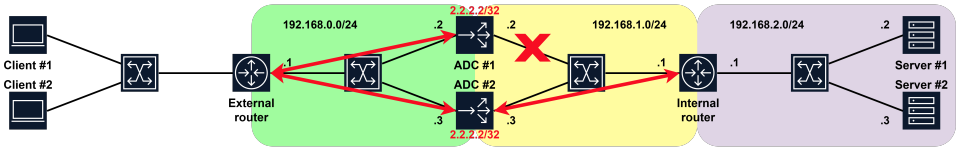



Рис. 6

Если BGP-сессия между External router и ADC #1 сохраняется, то ADC #1 будет продолжать анонсировать маршрут до адреса виртуального сервера, хотя реальные серверы для ADC #1 уже недостижимы. Таким образом, мы получаем классический black hole для трафика. Чтобы избежать этого, анонсирование маршрутов до виртуальных серверов должно быть условным. То есть ADC #1 должен отозвать маршрут до адреса виртуального сервера при падении тестов RHI:

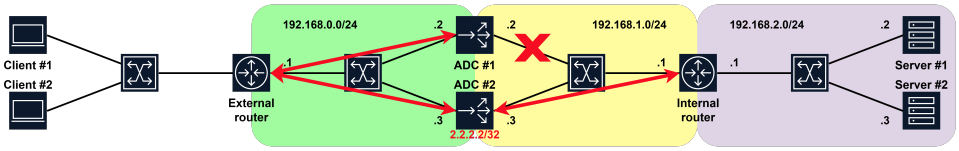



Рис. 7

### Настройка маршрутизации

При настройке BGP-маршрутизации необходимо указывать собственный номер автономной системы.

**Примечание**

Автономная система BGP — логическое объединение устройств в сети для маршрутизации через протокол BGP.

Ниже приведены примеры трех распространенных топологий настройки маршрутизации во внешнем сегменте сети. Настройка маршрутизации во внутреннем сегменте выполняется аналогично.

**Примечание**

В список обязательных команд для некоторых платформ необходимо включить команду задания ограничения на максимальное количество путей, между которыми производится ECMP-балансировка.

Пример такой команды для маршрутизаторов Cisco на базе IOS и IOS-XE:

```
platform loadbalance max-paths 2
```

### Топология #1

В топологии #1 все три устройства (External router, ADC #1 и ADC #2) принадлежат одной автономной системе:

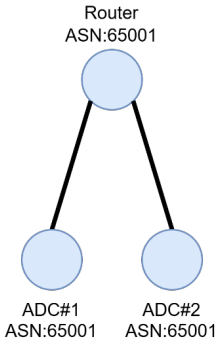



Рис. 8

Обычно на ADC #1 не нужны маршруты, объявляемые со стороны ADC #2. Если для конкретной сети вам нужно, чтобы ADC#1 знал маршруты, которые объявляет ADC #2 (или наоборот), то тогда на маршрутизаторе необходимо создать RR-кластер (Route Reflector cluster). Это позволит обойти ограничение на передачу префиксов, накладываемое правилом BGP split-horizon.

**Примечание**

Правило BGP split-horizon запрещает отправлять полученные от iBGP-соседа маршруты в сторону другого iBGP-соседа, защищая таким образом от зацикливания маршрутов.

Создание RR-кластера поможет также в ситуации, когда перед маршрутизатором External router расположен другой маршрутизатор, находящийся в той же самой автономной системе.

### Ключевые настройки маршрутизации для External router

На примере маршрутизатора Cisco Systems на базе IOS или IOS-XE:

```
!Имя устройства
hostname external
interface GigabitEthernet0/0
!Интерфейс в сторону систем балансировки
ip address 192.168.0.1 255.255.255.0
router bgp 65001
!Сессия с ADC#1
neighbor 192.168.0.2 remote-as 65001
!Сессия с ADC#2
neighbor 192.168.0.3 remote-as 65001
address-family ipv4
!Включение соседа
neighbor 192.168.0.2 activate
!Включение соседа
neighbor 192.168.0.3 activate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.2 default-originate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.3 default-originate
!Добавление соседа в RR-кластер
neighbor 192.168.0.2 route-reflector-client
!Добавление соседа в RR-кластер
neighbor 192.168.0.3 route-reflector-client
!Указание максимального количества iBGP путей, между которыми будет выполняться␣
↪балансировка
maximum-paths ibgp 2
end
```

### Ключевые настройки маршрутизации Angie ADC #1

```
!Имя устройства
hostname adc1
!Интерфейс в сторону External router
interface ens33
ip address 192.168.0.2/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
ip address 3.3.3.3/32
exit
router bgp 65001
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
```

```
network 2.2.2.2/32
network 3.3.3.3/32
exit-address-family
exit
end
```

### Ключевые настройки маршрутизации Angie ADC #2

```
!Имя устройства
hostname adc2
!Интерфейс в сторону External router
interface ens33
ip address 192.168.0.3/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
ip address 4.4.4.4/32
exit
router bgp 65001
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
network 2.2.2.2/32
network 4.4.4.4/32
exit-address-family
exit
end
```

### Проверка работы

```
external#sho ip bgp
BGP table version is 4, local router ID is 192.168.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
0.0.0.0 0.0.0.0 0 i
*mi 2.2.2.2/32 192.168.0.2 0 100 0 i
*>i 192.168.0.3 0 100 0 i
external#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
2.0.0.0/32 is subnetted, 1 subnets
B 2.2.2.2 [200/0] via 192.168.0.2, 00:12:23
[200/0] via 192.168.0.3, 00:12:23
192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C 192.168.0.0/24 is directly connected, GigabitEthernet0/0
L 192.168.0.1/32 is directly connected, GigabitEthernet0/0
```

Выводы просмотровых команд сокращены для краткости. Для тестирования работы RR-кластера мы завели на ADC #1 адрес `3.3.3.3/32`, а на ADC #2 — `4.4.4.4/32`.

Ниже представлены BGP-таблица и таблица маршрутизации от ADC #2:

```
adc2# sho ip bgp
BGP table version is 8, local router ID is 4.4.4.4, vrf id 0
Default local pref 100, local AS 65001
Status codes: s suppressed, d damped, h history, u unsorted, * valid, > best, =␣
↪multipath,
i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
*>i 0.0.0.0/0 192.168.0.1  0 100 0 i
*> 2.2.2.2/32 0.0.0.0(adc2) 0 32768 i
* i 192.168.0.2  0 100 0 i
*>i 3.3.3.3/32 192.168.0.2  0 100 0 i
*> 4.4.4.4/32 0.0.0.0(adc2)   0 32768 i
Displayed 4 routes and 5 total paths
adc2# sho ip bgp 3.3.3.3
BGP routing table entry for 3.3.3.3/32, version 7
Paths: (1 available, best #1, table default)
Not advertised to any peer
Local
192.168.0.2 from 192.168.0.2 (3.3.3.3)
Origin IGP, metric 0, localpref 100, valid, internal, bestpath-from-AS Local, best␣
↪(First path received)
Originator: 3.3.3.3, Cluster list: 192.168.0.1
Last update: Mon Apr 21 19:47:21 2025
adc2# sho ip ro
Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, F - PBR,
f - OpenFabric, t - Table-Direct,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure
B 0.0.0.0/0 [200/0] via 192.168.0.1, ens33, weight 1, 00:03:35
L * 2.2.2.2/32 is directly connected, lo, 00:40:12
C>* 2.2.2.2/32 is directly connected, lo, 00:40:12
B>* 3.3.3.3/32 [200/0] via 192.168.0.2, ens33, weight 1, 00:03:35
L * 4.4.4.4/32 is directly connected, lo, 00:40:12
C>* 4.4.4.4/32 is directly connected, lo, 00:40:12
C>* 192.168.0.0/24 is directly connected, ens33, 00:40:13
L>* 192.168.0.3/32 is directly connected, ens33, 00:40:13
```

### Топология #2

В топологии #2 все системы балансировки (Angie ADC #1 и Angie ADC #2) принадлежат одной автономной системе, а маршрутизатор External router – другой:

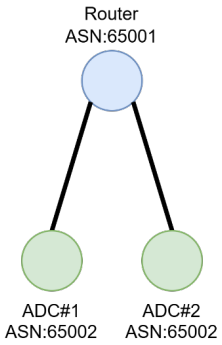



Рис. 9

Обычно на ADC #1 не нужны маршруты, объявляемые со стороны ADC #2. Если для конкретной сети вам нужно, чтобы ADC #1 знал маршруты, которые объявляет ADC #2 (или наоборот), то тогда на системе балансировки нагрузки надо разрешить прием маршрутов, содержащих собственный номер автономной системы в атрибуте `AS-PATH`.

### Ключевые настройки маршрутизации для External router

На примере маршрутизатора Cisco Systems на базе IOS или IOS-XE:

```
!Имя устройства
hostname external
!Интерфейс в сторону систем балансировки
interface GigabitEthernet0/0
ip address 192.168.0.1 255.255.255.0
router bgp 65001
!Сессия с ADC#1
neighbor 192.168.0.2 remote-as 65002
!Сессия с ADC#2
neighbor 192.168.0.3 remote-as 65002
address-family ipv4
!Включение соседа
neighbor 192.168.0.2 activate
!Включение соседа
neighbor 192.168.0.3 activate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.2 default-originate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.3 default-originate
!Указание максимального количества eBGP путей, между которыми будет выполняться␣
↪балансировка
maximum-paths 2
end
```

### Ключевые настройки маршрутизации Angie ADC #1

```
!Имя устройства
hostname adc1
!Интерфейс в сторону External router
interface ens33
```

```
ip address 192.168.0.2/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
ip address 3.3.3.3/32
exit
router bgp 65002
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
network 2.2.2.2/32
!Прием маршрутов, AS-PATH которых содержит собственный ASN
!Опция origin указывает на то, что приниматься будут только те маршруты,
!которые порождены в автономной системе с собственным ASN
neighbor 192.168.0.1 allowas-in origin
exit-address-family
exit
end
```

#### Ключевые настройки маршрутизации Angie ADC #2

```
!Имя устройства
hostname adc2
!Интерфейс в сторону External router
interface ens33
ip address 192.168.0.3/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
ip address 4.4.4.4/32
exit
router bgp 65002
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
network 2.2.2.2/32
!Прием маршрутов, AS-PATH которых содержит собственный ASN.
!Опция origin указывает на то, что приниматься будут только те маршруты,
!которые порождены в автономной системе с собственным ASN
neighbor 192.168.0.1 allowas-in origin
exit-address-family
exit
end
```

#### Проверка работы

```
external#sho ip bgp
BGP table version is 11, local router ID is 192.168.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
0.0.0.0 0.0.0.0 0 i
*> 2.2.2.2/32 192.168.0.2   0 0 65002 i
*m 192.168.0.3   0 0 65002 i
external#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
Gateway of last resort is not set
2.0.0.0/32 is subnetted, 1 subnets
B 2.2.2.2 [20/0] via 192.168.0.2, 00:13:18
[20/0] via 192.168.0.3, 00:13:18
192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly connected, GigabitEthernet0/0
L 192.168.0.1/32 is directly connected, GigabitEthernet0/0
```

Выводы просмотровых команд сокращены для краткости.

Для тестирования команды `allowas-in` мы завели на ADC #1 адрес `3.3.3.3/32`, а на ADC #2 — `4.4.4.4/32`.

Ниже представлены BGP-таблица и таблица маршрутизации от ADC #2:

```
adc2# sho ip bgp
BGP table version is 4, local router ID is 2.2.2.2, vrf id 0
Default local pref 100, local AS 65002
Status codes: s suppressed, d damped, h history, u unsorted, * valid, > best, =␣
↪multipath,
i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
*> 0.0.0.0/0 192.168.0.1  0 65001 i
*> 2.2.2.2/32 0.0.0.0(adc2)  0 32768 i
* 192.168.0.2 0 65001 65002 i
*> 3.3.3.3/32 192.168.0.2 0 65001 65002 i
*> 4.4.4.4/32 0.0.0.0(adc2) 0 32768 i
* 192.168.0.1 0 65001 65002 i
Displayed 4 routes and 6 total paths
adc2# sho ip ro
Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, F - PBR,
f - OpenFabric, t - Table-Direct,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure
B 0.0.0.0/0 [20/0] via 192.168.0.1, ens33, weight 1, 00:07:26
L * 2.2.2.2/32 is directly connected, lo, 00:12:14
C>* 2.2.2.2/32 is directly connected, lo, 00:12:14
B>* 3.3.3.3/32 [20/0] via 192.168.0.2, ens33, weight 1, 00:01:02
L * 4.4.4.4/32 is directly connected, lo, 00:06:43
C>* 4.4.4.4/32 is directly connected, lo, 00:06:43
```

```
C>* 192.168.0.0/24 is directly connected, ens33, 00:12:15
L>* 192.168.0.3/32 is directly connected, ens33, 00:12:15
```

### Топология #3

Все три устройства (External router, ADC #1 и ADC #2) принадлежат разным автономным системам:

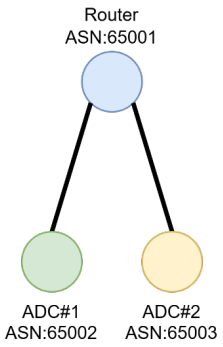



Рис. 10

Обычно маршрутизаторы Cisco не балансируют трафик до подсети, известной через соседей из разных автономных систем. Чтобы обойти данное ограничение, необходимо изменить поведение по умолчанию маршрутизатора BGP с помощью следующей команды:

```
bgp bestpath as-path multipath-relax
```

### Ключевые настройки маршрутизации для External router

На примере маршрутизатора Cisco Systems на базе IOS или IOS-XE:

```
!Имя устройства
hostname external
!Интерфейс в сторону систем балансировки
interface GigabitEthernet0/0
ip address 192.168.0.1 255.255.255.0
router bgp 65001
!Команда, разрешающая выполнять балансировку для префиксов, полученных из разных ASN
bgp bestpath as-path multipath-relax
!Сессия с ADC#1
neighbor 192.168.0.2 remote-as 65002
!Сессия с ADC#2
neighbor 192.168.0.3 remote-as 65003
address-family ipv4
!Включение соседа
neighbor 192.168.0.2 activate
!Включение соседа
neighbor 192.168.0.3 activate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.2 default-originate
!Анонсирование маршрута по умолчанию
neighbor 192.168.0.3 default-originate
!Указание максимального количества eBGP путей, между которыми будет выполняться␣
↪балансировка
maximum-paths 2
end
```

### Ключевые настройки маршрутизации Angie ADC #1

```
!Имя устройства
hostname adc1
!Интерфейс в сторону External router
interface ens33
ip address 192.168.0.2/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
exit
router bgp 65002
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
network 2.2.2.2/32
exit-address-family
exit
end
```

### Ключевые настройки маршрутизации Angie ADC #2

```
!Имя устройства
hostname adc2
!Интерфейс в сторону External router
interface ens33
ip address 192.168.0.3/24
exit
!Loopback-интерфейс для адресов виртуальных серверов
interface lo
ip address 2.2.2.2/32
exit
router bgp 65003
!Сессия с External router
neighbor 192.168.0.1 remote-as 65001
address-family ipv4 unicast
!Добавление префикса для адреса виртуального сервера в BGP
network 2.2.2.2/32
exit-address-family
exit
end
```

### Проверка работы

```
external#sho ip bgp
BGP table version is 11, local router ID is 192.168.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
0.0.0.0 0.0.0.0 0 i
*> 2.2.2.2/32 192.168.0.2 0 0 65002 i
```

```
*m 192.168.0.3 0 0 65003 i
external#sho ip ro
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
Gateway of last resort is not set
2.0.0.0/32 is subnetted, 1 subnets
B 2.2.2.2 [20/0] via 192.168.0.2, 00:10:22
[20/0] via 192.168.0.3, 00:16:07
192.168.0.0/24 is variably subnetted, 2 subnets, 2 masks
C 192.168.0.0/24 is directly connected, GigabitEthernet0/0
L 192.168.0.1/32 is directly connected, GigabitEthernet0/0
```

Выводы просмотровых команд сокращены для краткости.

#### Применение с IP transparency

Все описанное выше относится к режиму работы Angie ADC с выключенной опцией IP transparency. В этом случае Angie ADC будет выполнять проксирование клиентских запросов с подменой адресов источника и получателя. Если Angie ADC подменяет адрес клиента, то при маршрутизации нет необходимости анонсировать на Internal router маршрут по умолчанию. Если ADC #1 и ADC #2 используют разные адреса для подмены адреса клиента, то нет необходимости заботиться о том, с какими значениями Local preference (и иными атрибутами пути) маршруты до соответствующих адресов попадают на Internal router.

При включении опции IP Transparency Angie ADC будет сохранять исходный адрес отправителя в IP-пакетах, позволяя серверам на сетевом уровне различать клиентские подключения.

#### Заключение

Система балансировки нагрузки Angie ADC может работать в любой из перечисленных топологий. Каждый вариант имеет свои особенности настройки и работы, о чем следует помнить при проектировании и настройке информационной системы.

### 4.2.3 OSPF-маршрутизация

В этом разделе описана настройка OSPF-маршрутизации для обеспечения высокой доступности в режиме резервирования (Active-Standby) и в режиме распределения нагрузки (Active-Active). Для настройки используется интерфейс командной строки (CLI) Angie ADC.

В этом разделе:

- *Режим резервирования (Active-Standby)*
- *Режим распределения нагрузки (Active-Active)*

#### 4.2.3.1 Резервирование с помощью протокола OSPF (Active-Standby)

#### Введение

В этом разделе описана настройка высокой доступности с использованием протокола OSPF в режиме резервирования (Active-Standby). Для настройки OSPF используется интерфейс командной строки (CLI) Angie ADC.

В качестве протокола динамической маршрутизации используется OSPFv2 для адресного семейства IPv4 unicast. Отказоустойчивость в рамках IPv6 unicast настраивается аналогично.

Для обеспечения отказоустойчивости нужны две системы балансировки Angie ADC — основная и резервная. Резервная система Angie ADC будет принимать нагрузку при отказе основной.

Пример типового участка сети с системами балансировки Angie ADC (ADC #1 и ADC #2):

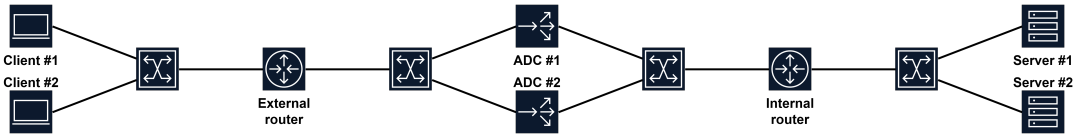



Рис. 1

Добавим *IP-адресацию*:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

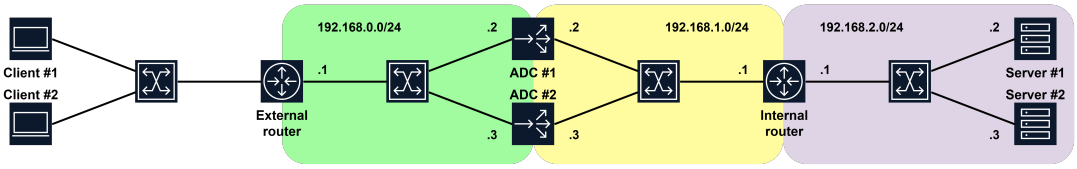



Рис. 2

*Адресация клиентов* не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть, они могут подключаться из любой сети, включая сеть интернет. Для этих целей External router будет анонсировать маршрут по умолчанию по OSPF для ADC #1 и ADC #2. Мы будем использовать OSPF-домен, включающий в себя четыре устройства: External router, Internal router, ADC #1 и ADC #2.

Красным текстом на схеме ниже отмечены *IP-адреса виртуальных серверов*. Это те адреса, к которым подключаются клиенты снаружи. Системы балансировки Angie ADC должны анонсировать маршруты до этих виртуальных серверов в сторону External router, причем маршрут через ADC #1 должен быть более предпочтительным. Добиться этого можно разными способами, например, манипулируя значением метрики внешнего маршрута в OSPF. Чем больше значение метрики, тем менее предпочтительный маршрут. В данном документе мы будем вручную задавать косты для интерфейсов системы балансировки.

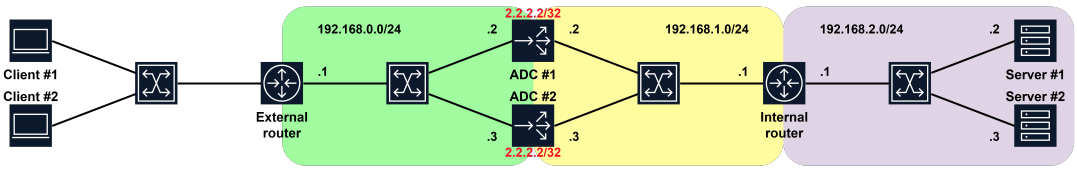



Рис. 3

Трафик от серверов в сторону клиента должен проходить через ту же систему балансировки, что и трафик от клиентов к серверам. Добиться этого можно также путем манипулирования значением метрики для префикса `0.0.0.0/0`. Манипуляции с маршрутом по умолчанию нужны при включении опции IP transparency. В противном случае трафик и так будет возвращаться через ту же систему балансировки, так как поведение ADC напоминает операцию sNAT.

### Обработка типовых сбоев

Ниже описаны сценарии работы Angie ADC при сбоях.

### Полный отказ ADC #1

В этом случае оба OSPF-соседства с ADC #1 разрываются, никакие объявления маршрутов через ADC #1 не пролетают. Остается связь по OSPF только с ADC #2. То есть, переключение на резервную систему производится полностью автоматически, никакие дополнительные настройки не требуются.

Красными стрелками обозначены рабочие OSPF-соседства:

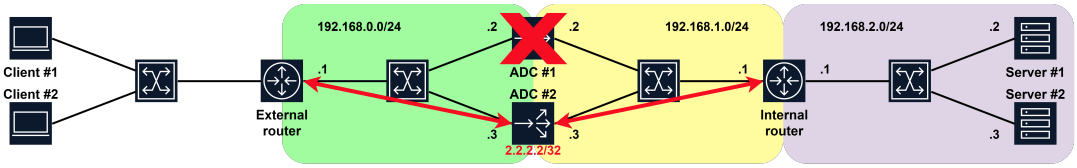



Рис. 4

### Отказ внешнего интерфейса ADC #1

Предлагаемая схема не предполагает возможности отслеживания частичных отказов и реагирования на них. Пример такого отказа представлен на схеме ниже.

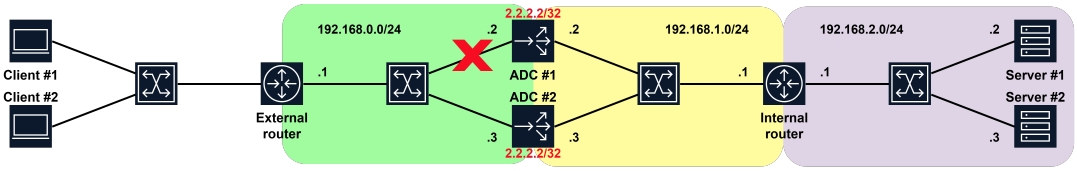



Рис. 5

В случае обозначенного на схеме отказа линка разрывается OSPF-сессия между ADC #1 и External router. ADC #1 теперь не может объявить маршрут до адреса виртуального сервера в сторону External router напрямую. Internal router рассчитывает маршрут по умолчанию только через ADC #1, а External router считает маршрут до адреса виртуального сервера только через полностью работающий ADC #2.

### Отказ линка между ADC #1 и Internal router

Отказ линка между ADC #1 и Internal router более сложная ситуация: если OSPF-соседство между External router и ADC #1 сохраняется, то ADC #1 будет продолжать анонсировать маршрут до адреса виртуального сервера, хотя реальные сервера для системы балансировки ADC #1 уже недостижимы.

Таким образом, мы получаем классический пример black hole для трафика:

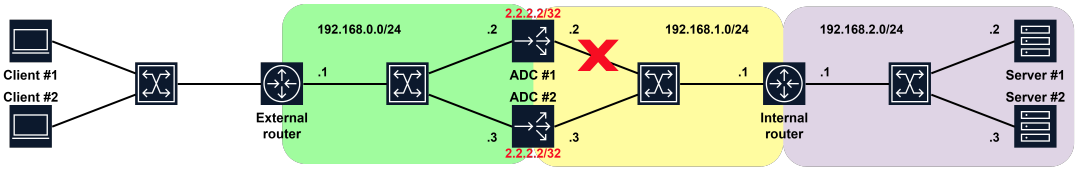



Рис. 6

Чтобы избежать этой проблемы, необходимо, чтобы ситуация отказа только одного из линков была невозможной, т.е. нужно обеспечить отказоустойчивое подключение к сети для хоста виртуализации, на котором работает ADC.

Рекомендованная схема подключения:

- Объединить в одну LAG-группу все физические интерфейсы хоста виртуализации и использовать этот виртуальный LAG-интерфейс в качестве транка, то есть передавать по нему тегированными несколько виртуальных сетей (один VLAN для external сети, второй VLAN для internal сети).
- Для резервирования коммутаторов можно использовать технологию vPC или стекирование, пример представлен на схеме ниже.

Из соображений простоты на схеме не отображено резервирование внешнего и внутреннего маршрутизаторов.

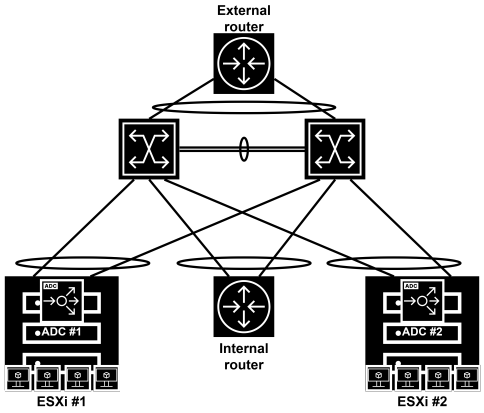



Рис. 7

### Настройки маршрутизации

Приведем ключевые настройки маршрутизации для External router, Internal router, ADC #1 и ADC #2.

### Ключевые настройки маршрутизации для External router

```
hostname external
!Имя устройства
interface GigabitEthernet0/0
 ip address 192.168.0.1 255.255.255.0
!Интерфейс в сторону систем балансировки
router ospf 1
 network 192.168.0.0 0.0.0.255 area 0
!Включение OSPF на интерфейсе в сторону систем балансировки
 default-information originate always
!Анонсирование маршрута по умолчанию
end
```

### Ключевые настройки маршрутизации для Internal router

```
hostname internal
!Имя устройства
interface GigabitEthernet0/0
 ip address 192.168.1.1 255.255.255.0
```

```
!Интерфейс в сторону систем балансировки
interface GigabitEthernet1/0
ip address 192.168.2.1 255.255.255.0
!Интерфейс в сторону подсети с реальными серверами
router ospf 1
 redistribute connected subnets
!добавление в OSPF информации о подсети с серверами
 network 192.168.1.0 0.0.0.255 area 0
!Включение OSPF на интерфейсе в сторону систем балансировки
end
```

### Ключевые настройки маршрутизации системы балансировки ADC #1

```
hostname adc1
!Имя устройства
interface ens33
!Интерфейс в сторону маршрутизатора external router
 description external
 ip address 192.168.0.2/24
 ip ospf 1 area 0
!Включение OSPF на интерфейсе
 ip ospf cost 10
!Назначение стоимости интерфейса в OSPF вручную
exit
interface ens37
!Интерфейс в сторону маршрутизатора internal router
 description internal
 ip address 192.168.1.2/24
 ip ospf 1 area 0
!Включение OSPF на интерфейсе
exit
interface lo
!Loopback-интерфейс для назначения адресов виртуальных серверов
 ip address 2.2.2.2/32
!Адрес тестового виртуального сервера
exit
ip prefix-list c2o seq 5 permit 2.2.2.2/32
!Префикс-лист, который используется для отбора префиксов для добавления в OSPF
route-map c2o permit 10
 match ip address prefix-list c2o
!Route-map с помощью которой производится отбор префиксов для добавления в OSPF
exit
router ospf 1
!Настройки процесса маршрутизации OSPF
 ospf router-id 2.2.2.2
!Необходимо вручную задать идентификатор маршрутизатора для OSPF
!Если этого не сделать, то автоматика может выбрать одинаковые для ADC#1 и ADC#2
 redistribute connected metric 10 route-map c2o
!Добавление информации о подключенных сетях в OSPF
!Метрика 10, чтобы стать более предпочтительным next-hop по сравнению с ADC#2
exit
end
```

### Ключевые настройки маршрутизации системы балансировки ADC #2

```
hostname adc2
!Имя устройства
interface ens33
!Интерфейс в сторону маршрутизатора external router
 description external
 ip address 192.168.0.3/24
 ip ospf 1 area 0
!Включение OSPF на интерфейсе
 ip ospf cost 100
!Назначение стоимости интерфейса в OSPF вручную
exit
interface ens37
!Интерфейс в сторону маршрутизатора internal router
 description internal
 ip address 192.168.1.3/24
 ip ospf 1 area 0
!Включение OSPF на интерфейсе
exit
interface lo
!Loopback-интерфейс для назначения адресов виртуальных серверов
 ip address 2.2.2.2/32
!Адрес тестового виртуального сервера
exit
ip prefix-list c2o seq 5 permit 2.2.2.2/32
!Префикс-лист, который используется для отбора префиксов для добавления в OSPF
route-map c2o permit 10
 match ip address prefix-list c2o
!Route-map с помощью которой производится отбор префиксов для добавления в OSPF
exit
router ospf 1
!Настройки процесса маршрутизации OSPF
 ospf router-id 22.22.22.22
!Необходимо вручную задать идентификатор маршрутизатора для OSPF
!Если этого не сделать, то автоматика может выбрать одинаковые для ADC#1 и ADC#2
 redistribute connected metric 20 route-map c2o
!Добавление информации о подключенных сетях в OSPF
!Метрика 20, чтобы стать менее предпочтительным next-hop по сравнению с ADC#1
exit
end
```

### Проверка работы

Выполним проверку работы системы маршрутизации, когда ADC #1 и ADC #2 находятся в полностью работоспособном состоянии.

**Примечание**

В листингах ниже отсутствуют префиксы, не относящиеся к обсуждаемым вопросам отказоустойчивости.

Маршрутизатор External router видит маршрут к адресу виртуального сервера через обе системы балансировки, однако маршрут через ADC #1 является более предпочтительным.

```
external#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
```

```
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is not set
      2.0.0.0/32 is subnetted, 1 subnets
O E2     2.2.2.2 [110/10] via 192.168.0.2, 00:41:28, GigabitEthernet0/0
external#sho ip os da ex 2.2.2.2
            OSPF Router with ID (192.168.0.1) (Process ID 1)
                Type-5 AS External Link States
  Routing Bit Set on this LSA in topology Base with MTID 0

  LS age: 898
  Options: (No TOS-capability, No DC, Upward)
  LS Type: AS External Link
  Link State ID: 2.2.2.2 (External Network Number )
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000003
  Checksum: 0xAA0C
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 10
        Forward Address: 0.0.0.0
        External Route Tag: 0

  LS age: 885
  Options: (No TOS-capability, No DC, Upward)
  LS Type: AS External Link
  Link State ID: 2.2.2.2 (External Network Number )
  Advertising Router: 22.22.22.22
  LS Seq Number: 80000003
  Checksum: 0xB4A7
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

Маршрутизатор Internal router видит маршрут по умолчанию (путь через ADC #1 является более предпочтительным).

```
internal#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is 192.168.1.2 to network 0.0.0.0
```

```
O*E2  0.0.0.0/0 [110/1] via 192.168.1.2, 00:43:21, GigabitEthernet0/0
internal#sho ip os da ext 0.0.0.0
            OSPF Router with ID (192.168.2.1) (Process ID 1)
                Type-5 AS External Link States
  Routing Bit Set on this LSA in topology Base with MTID 0
  LS age: 992
  Options: (No TOS-capability, DC, Upward)
  LS Type: AS External Link
  Link State ID: 0.0.0.0 (External Network Number )
  Advertising Router: 192.168.0.1
  LS Seq Number: 80000003
  Checksum: 0x2521
  Length: 36
  Network Mask: /0
        Metric Type: 2 (Larger than any link state path)
        MTID: 0
        Metric: 1
        Forward Address: 0.0.0.0
        External Route Tag: 1
```

Отключим теперь ADC #1 полностью и проверим маршрутизацию.

Маршрутизатор Internal router:

```
internal#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is 192.168.1.3 to network 0.0.0.0
O*E2  0.0.0.0/0 [110/1] via 192.168.1.3, 00:01:08, GigabitEthernet0/0
```

Маршрутизатор External router:

```
external#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override
Gateway of last resort is not set
      2.0.0.0/32 is subnetted, 1 subnets
O E2     2.2.2.2 [110/10] via 192.168.0.3, 00:00:10, GigabitEthernet0/0
```

После возвращения в строй системы балансировки ADC #1 маршрутизация через это устройство восстанавливается.

#### Заключение

Все описанное выше относится к режиму работы системы балансировки трафика с включенной опцией IP transparency. При опции IP Transparency Angie ADC сохраняет исходный адрес отправителя в IP-пакетах, позволяя серверам на сетевом уровне различать клиентские подключения. Очень часто администраторы систем балансировки отключают эту опцию, в этом случае Angie ADC будет выполнять не только dNAT, но sNAT над трафиком, приходящим от клиентов. Если система балансировки выполняет sNAT, то в этом случае маршрутизация несколько упрощается, так как в этом случае нет необходимости анонсировать в сторону Internal router маршрут по умолчанию. Если ADC #1 и ADC #2 выполняют sNAT в разные адреса, то в этом случае нет необходимости заботиться о том, с какими значениями метрики маршруты до соответствующих адресов попадают на маршрутизатор Internal router, так как анонсы будут уникальными.

### 4.2.3.2 Распределение нагрузки с помощью протокола OSPF (Active-Active)

#### Введение

В этом разделе описана настройка высокой доступности с использованием протокола OSPF в режиме распределения нагрузки (Active-Active). Для настройки OSPF используется интерфейс командной строки (CLI) Angie ADC.

В качестве протокола динамической маршрутизации используется OSPFv2 для адресного семейства IPv4 unicast. Отказоустойчивость в рамках IPv6 unicast настраивается аналогично.

Для обеспечения распределения нагрузки между ADC нужны две системы балансировки Angie, работающие параллельно и одновременно. Пример типового участка сети с системами балансировки Angie ADC (ADC #1 и ADC #2):

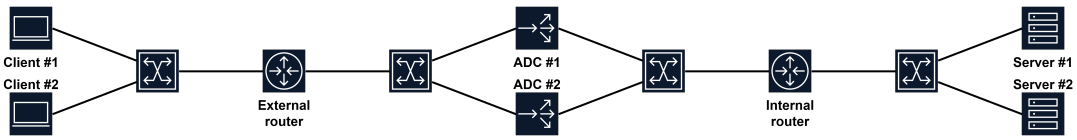



Рис. 1

Добавим *IP-адресацию*:

- В сегменте сети, куда подключаются External router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.0.0/24`.
- В сегменте сети, куда подключаются Internal router, ADC #1 и ADC #2, будет использоваться подсеть `192.168.1.0/24`.
- Четвертый октет адреса для каждого устройства (в рамках соответствующей подсети) показан на схеме ниже.

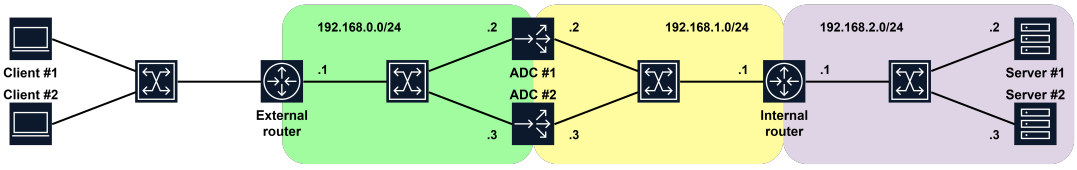



Рис. 2

*Адресация клиентов* не приводится, так как в данном случае мы будем считать, что их адреса формально не определены, то есть, они могут подключаться из любой сети, включая сеть интернет. Для этих целей External router будет анонсировать маршрут по умолчанию по OSPF для ADC #1 и ADC #2. Мы будем использовать OSPF-домен, включающий в себя четыре устройства: External router, Internal router, ADC #1 и ADC #2.

Красным текстом на схеме ниже отмечены *IP-адреса виртуальных серверов*. Это те адреса, к которым подключаются клиенты снаружи. Чтобы маршрутизатор External router мог выполнить распределение трафика между системами балансировки Angie ADC (см. ECMP – Equal-Cost Multi-Path), они должны одинаково анонсировать маршруты до виртуальных серверов в сторону External router (должны совпадать типы маршрутов и косты).

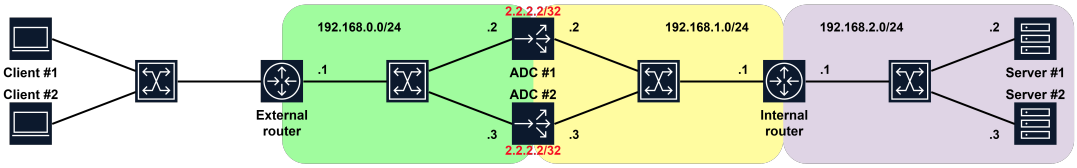



Рис. 3

Трафик внутри пользовательской сессии от серверов должен возвращаться строго через ту же систему балансировки, через которую он проходил до этого. Добиться этого можно с помощью механизма *SNAT*, когда адрес клиента подменяется на адрес интерфейса системы балансировки.

### Обработка типовых сбоев

Ниже описаны сценарии работы Angie ADC при сбоях.

### Полный отказ ADC #1

В этом случае оба OSPF-соседства с ADC #1 разрываются, никакие объявления маршрутов через ADC #1 не пролетают. Остается связь по OSPF только с ADC #2. То есть, переключение на резервную систему производится полностью автоматически, никакие дополнительные настройки не требуются.

Красными стрелками обозначены рабочие OSPF-соседства:

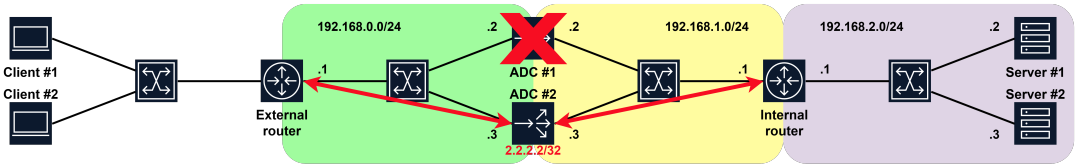



Рис. 4

### Отказ внешнего интерфейса ADC #1

Пример такого отказа представлен на схеме ниже.

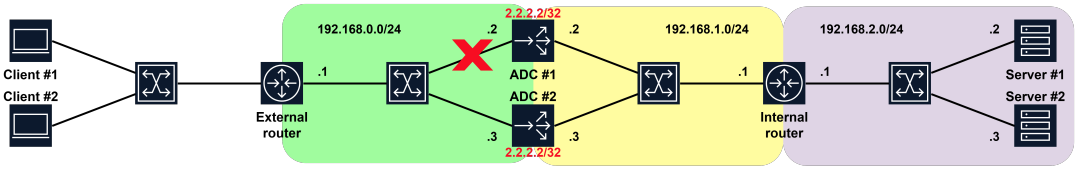



Рис. 5

В случае обозначенного на схеме отказа линка разрывается OSPF-сессия между ADC #1 и External router. ADC #1 теперь не может объявить маршрут до адреса виртуального сервера в сторону External router напрямую. Internal router получает маршруты до адресов, которые используются в SNAT от обеих систем балансировки, однако трафик от клиентов, прошедший через ADC #2,

будет возвращаться тем же путем (так же через ADC #2), так как ADC #1 и ADC #2 используют уникальные адреса для SNAT.

### Отказ линка между ADC #1 и Internal router

Отказ линка между ADC #1 и Internal router более сложная ситуация: если OSPF-соседство между External router и ADC #1 сохраняется, то ADC #1 будет продолжать анонсировать маршрут до адреса виртуального сервера, хотя реальные сервера для системы балансировки ADC #1 уже недостижимы.

Таким образом, мы получаем классический пример black hole для трафика:

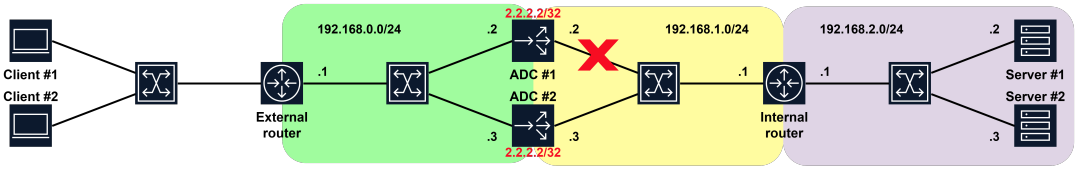



Рис. 6

Чтобы избежать этой проблемы, необходимо, чтобы ситуация отказа только одного из линков была невозможной, т.е. нужно обеспечить отказоустойчивое подключение к сети для хоста виртуализации, на котором работает ADC.

Вторым способом решить указанную проблему является *технология RHI*, позволяющая отозвать маршрут при недостижимости апстрим-серверов.

Рекомендованная схема подключения:

- Объединить в одну LAG-группу все физические интерфейсы хоста виртуализации и использовать этот виртуальный LAG-интерфейс в качестве транка, то есть передавать по нему тегированными несколько виртуальных сетей (один VLAN для внешней (external) сети, второй VLAN для внутренней (internal) сети).
- Для резервирования коммутаторов можно использовать такие технологии как vPC, VSS или стекирование, пример представлен на схеме ниже.

Из соображений простоты на схеме не отображено резервирование внешнего и внутреннего маршрутизаторов.

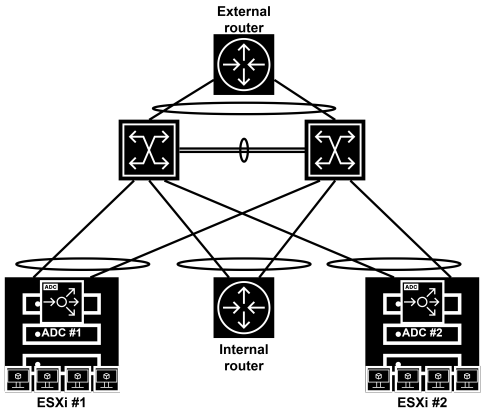



Рис. 7

**Предупреждение**

Маршрутизатор, выполняющий ECMP-балансировку, и обе системы Angie ADC могут располагаться в одной OSPF-зоне или в разных. Если при этом маршрутизатор является OSPF ABR (Area Border Router), а ADC #1 и ADC #2 находятся в разных OSPF-зонах (видны через разные интерфейсы ABR), то ECMP-балансировка может не поддерживаться некоторыми вендорами сетевого оборудования.

Такой дизайн сети не рекомендуется:

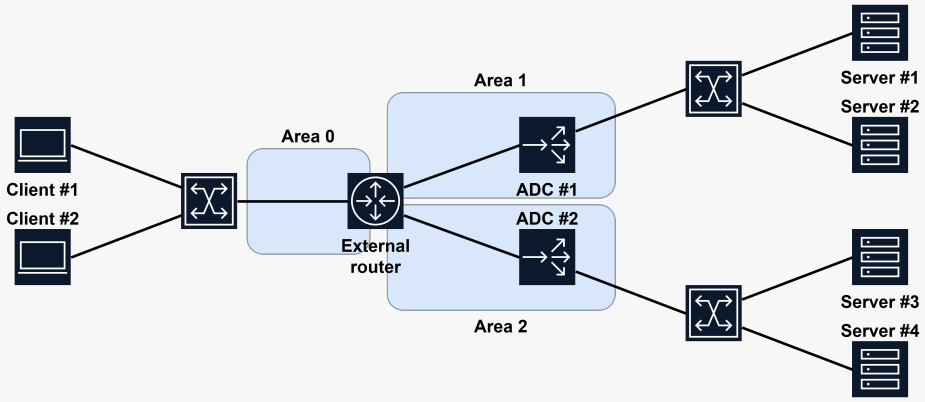



Рис. 8

### Настройки маршрутизации

Приведем ключевые настройки маршрутизации для External router, Internal router, ADC #1 и ADC #2.

### Ключевые настройки маршрутизации для External router

```
!Имя устройства
hostname external
!Интерфейс в сторону систем балансировки
interface GigabitEthernet0/0
ip address 192.168.0.1 255.255.255.0
router ospf 1
!Включение OSPF на интерфейсе в сторону систем балансировки
network 192.168.0.0 0.0.0.255 area 0
!Анонсирование маршрута по умолчанию
default-information originate always
end
```

### Ключевые настройки маршрутизации для Internal router

```
!Имя устройства
hostname internal
!Интерфейс в сторону систем балансировки
interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
!Интерфейс в сторону подсети с реальными серверами
interface GigabitEthernet1/0
ip address 192.168.2.1 255.255.255.0
router ospf 1
!Добавление в OSPF информации о подсети с серверами
redistribute connected subnets
!Включение OSPF на интерфейсе в сторону систем балансировки
```

```
network 192.168.1.0 0.0.0.255 area 0
end
```

### Ключевые настройки маршрутизации системы балансировки ADC #1

```
!Имя устройства
hostname adc1
!Интерфейс в сторону маршрутизатора external router
interface ens33
description external
ip address 192.168.0.2/24
!Включение OSPF на интерфейсе
ip ospf 1 area 0
exit
!Интерфейс в сторону маршрутизатора internal router
interface ens37
description internal
ip address 192.168.1.2/24
!Включение OSPF на интерфейсе
ip ospf 1 area 0
exit
!Loopback-интерфейс для назначения адресов виртуальных серверов
interface lo
!Адрес тестового виртуального сервера
ip address 2.2.2.2/32
exit
!Префикс-лист, который используется для отбора префиксов для добавления в OSPF
ip prefix-list c2o seq 5 permit 2.2.2.2/32
!Route-map, с помощью которой производится отбор префиксов для добавления в OSPF
route-map c2o permit 10
match ip address prefix-list c2o
exit
!Настройки процесса маршрутизации OSPF
router ospf 1
!Необходимо вручную задать идентификатор маршрутизатора для OSPF
!Иначе автоматика может выбрать одинаковые идентификаторы для ADC#1 и ADC#2
ospf router-id 2.2.2.2
!Добавление информации о подключенных сетях в OSPF
redistribute connected route-map c2o
exit
end
```

### Ключевые настройки маршрутизации системы балансировки ADC #2

```
!Имя устройства
hostname adc2
!Интерфейс в сторону маршрутизатора external router
interface ens33
description external
ip address 192.168.0.3/24
!Включение OSPF на интерфейсе
ip ospf 1 area 0
exit
!Интерфейс в сторону маршрутизатора internal router
interface ens37
description internal
```

```
ip address 192.168.1.3/24
!Включение OSPF на интерфейсе
ip ospf 1 area 0
exit
!Loopback-интерфейс для назначения адресов виртуальных серверов
interface lo
!Адрес тестового виртуального сервера
ip address 2.2.2.2/32
exit
!Префикс-лист, который используется для отбора префиксов для добавления в OSPF
ip prefix-list c2o seq 5 permit 2.2.2.2/32
!Route-map, с помощью которой производится отбор префиксов для добавления в OSPF
route-map c2o permit 10
match ip address prefix-list c2o
exit
!Настройки процесса маршрутизации OSPF
router ospf 1
!Необходимо вручную задать идентификатор маршрутизатора для OSPF
!Иначе автоматика может выбрать одинаковые идентификаторы для ADC#1 и ADC#2
ospf router-id 22.22.22.22
!Добавление информации о подключенных сетях в OSPF
redistribute connected route-map c2o
exit
end
```

### Проверка работы

Выполним проверку работы системы маршрутизации, когда ADC #1 и ADC #2 находятся в полностью работоспособном состоянии.

**Примечание**

В листингах ниже отсутствуют префиксы, не относящиеся к обсуждаемым вопросам отказоустойчивости.

Маршрутизатор External router видит маршрут к адресу виртуального сервера через обе системы балансировки.

```
external#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
Gateway of last resort is not set
2.0.0.0/32 is subnetted, 1 subnets
O E2     2.2.2.2 [110/20] via 192.168.0.2, 00:41:28, GigabitEthernet0/0
via 192.168.0.3, 00:41:15, GigabitEthernet0/0
external#sho ip os da ex 2.2.2.2
OSPF Router with ID (192.168.0.1) (Process ID 1)
Type-5 AS External Link States
Routing Bit Set on this LSA in topology Base with MTID 0
```

```
LS age: 898
Options: (No TOS-capability, No DC, Upward)
LS Type: AS External Link
Link State ID: 2.2.2.2 (External Network Number )
Advertising Router: 2.2.2.2
LS Seq Number: 80000003
Checksum: 0xAA0C
Length: 36
Network Mask: /32
Metric Type: 2 (Larger than any link state path)
MTID: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0

LS age: 885
Options: (No TOS-capability, No DC, Upward)
LS Type: AS External Link
Link State ID: 2.2.2.2 (External Network Number )
Advertising Router: 22.22.22.22
LS Seq Number: 80000003
Checksum: 0xB4A7
Length: 36
Network Mask: /32
Metric Type: 2 (Larger than any link state path)
MTID: 0
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

По умолчанию Angie ADC будет устанавливать соединение с апстрим-сервером с адреса того интерфейса, через который этот сервер достижим, в соответствии с таблицей маршрутизации (`192.168.1.2` для ADC #1 и `192.168.1.3` для ADC #2). Оба адреса принадлежат подсети `192.168.1.0/24`, которая для маршрутизатора Internal router является непосредственно подключенной (Directly Connected). Анонсировать другие адреса не нужно. Если используется SNAT pool, то каждая из систем балансировки объявит в сеть дополнительные адреса, соответствующие пулу.

Отключим теперь ADC #1 полностью и проверим маршрутизацию.

Маршрутизатор External router:

```
external#sho ip ro os
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
+ - replicated route, % - next hop override
Gateway of last resort is not set
2.0.0.0/32 is subnetted, 1 subnets
O E2     2.2.2.2 [110/20] via 192.168.0.3, 01:40:10, GigabitEthernet0/0
```

После возвращения в строй ADC #1 маршрутизация через это устройство восстанавливается.

#### Применение с IP transparency

Все описанное выше относится к режиму работы системы балансировки трафика с отключенной опцией IP transparency. При включении опции IP Transparency Angie ADC сохраняет исходный адрес отправителя в IP-пакетах, позволяя серверам на сетевом уровне различать клиентские подключения. Однако в этом случае маршрутизатор Internal router будет возвращать трафик через неправильную систему балансировки. Исходный адрес клиента может быть передан серверу либо с помощью заголовка X-FORWARDED-FOR, либо с помощью специального proxy protocol.

### 4.2.4 Статическая маршрутизация

Статическую маршрутизацию можно настроить через CLI (режим настройки configuration).

Синтаксис: `set ip <ip_input>`. Значения для `<ip_input>`:

- `rule from <address> table <value>` — установка правила.
- `route <address> {nexthop <address> | interface <name>} [table <value>] [type local]` — установка маршрута. Минимально необходимо задать адрес и `nexthop` или `interface`, остальные параметры (таблица маршрутизации и тип маршрута) опциональны.

Пример:

```
$$ configuration
## set ip rule from 0.0.0.0/0 table 100
## set ip route 0.0.0.0/0 nexthop 10.90.90.90
```

### 4.2.5 Использование протокола BFD для уменьшения времени реакции

#### 4.2.5.1 Введение

Стандартные таймеры протоколов маршрутизации слишком велики и не подходят для современных сетей. Уменьшить время реакции протокола на какое-либо событие (непрямое падение, indirect failure) можно двумя способами:

- за счет уменьшения значения таймеров (может привести к увеличению нагрузки);
- с помощью специализированного протокола — BFD (предпочтительный способ).

BFD (Bidirectional Forwarding Detection) — это легковесный протокол, используемый для быстрого обнаружения отказов сетевых каналов между двумя устройствами. Он работает независимо и комплементарно к протоколам маршрутизации и помогает быстро определить, что связь между узлами нарушена. Если в инфраструктуре важна быстрая реакция на сбои, BFD значительно уменьшает время сходимости сети, так как позволяет получить субсекундное время реакции.

Как работает BFD?

1. Установление сессии: два сетевых устройства обмениваются контрольными пакетами BFD для установления связи.
2. Мониторинг: оба узла отправляют друг другу небольшие пакеты с фиксированным интервалом.
3. Обнаружение отказа: если один из узлов перестает получать пакеты от соседа в течение заданного времени, соединение объявляется "down".
4. Реакция: протоколы маршрутизации отслеживают состояние BFD-сессий и разрывают свои соединения, что позволяет переключиться на резервный канал менее, чем за секунду после сбоя.

**Примечание**

Для настройки используется командная строка Angie ADC CLI.

#### 4.2.5.2 Настройка с OSPF

Для протокола OSPF поддержка BFD активируется на интерфейсе с помощью команды `ip ospf bfd`.

Пример конфигурации интерфейса:

```
interface ens33
ip address 192.168.0.2/24
ip ospf 1 area 0
ip ospf bfd
ip ospf bfd profile test
```

Если параметры работы протокола BFD на каком-либо интерфейсе должны отличаться от стандартных, то новые параметры можно указать путем привязки профиля.

Протокол OSPF осуществляет динамическое обнаружение соседей на интерфейсе, а затем протокол BFD пытается установить с ними соседство:

```
angie-va# sho bfd peers
BFD Peers:
peer 192.168.0.3 vrf default interface ens33
ID: 192599120
Remote ID: 626157519
Active mode
Status: up
Uptime: 4 second(s)
Diagnostics: ok
Remote diagnostics: ok
Peer Type: dynamic
RTT min/avg/max: 0/0/0 usec
Local timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms
Echo transmission interval: disabled
Remote timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms
```

Протокол BFD может установить сессию с каким-либо соседом через определенный интерфейс и одновременно с этим не установить сессию с другим соседом. Это ожидаемое поведение для соседей, не обладающих поддержкой BFD. OSPF-соединение при этом не разрывается.

Если протокол BFD смог установить сессию с каким-либо из OSPF-соседей, то доступность этого соседа будет определяться с помощью протокола BFD.

Если протокол BFD не смог установить сессию с каким-либо из OSPF-соседей, то доступность этого соседа будет определяться с помощью встроенных механизмов OSPF (протокол Hello). При этом команда `sho bfd peers` будет отображать такого соседа в статусе `down`:

```
angie-va# sho bfd peers
BFD Peers:
peer 192.168.0.3 vrf default interface ens33
ID: 192599120
Remote ID: 0
Active mode
Status: down
```

```
Downtime: 4 minute(s), 0 second(s)
Diagnostics: ok
Remote diagnostics: ok
Peer Type: dynamic
RTT min/avg/max: 0/0/0 usec
Local timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms
Echo transmission interval: disabled
Remote timers:
Detect-multiplier: 3
Receive interval: 1000ms
Transmission interval: 1000ms
Echo receive interval: disabled
```

#### 4.2.5.3 Настройка с BGP

По умолчанию при указании BGP-соседа BFD-сессия между маршрутизаторами не устанавливается.

Пример такой конфигурации:

```
angie-va# sho run
! Часть конфига, не относящаяся к BGP/BFD, удалена для краткости.
router bgp 3
no bgp ebgp-requires-policy
neighbor 192.168.0.2 remote-as 2
exit
!
angie-va# sho ip bg su
IPv4 Unicast Summary:
BGP router identifier 3.3.3.3, local AS number 3 VRF default vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 1, using 24 KiB of memory
Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/
↪PfxRcd   PfxSnt Desc
192.168.0.2     4          2         8         9        0    0    0 00:05:35
↪  0        0 N/A
! BGP сессия успешно установлена, хотя никаких BFD-соседей у системы нет.
Total number of neighbors 1

angie-va# sho bfd pe
BFD Peers:
angie-va#
```

Чтобы активировать установление BFD-сессии с BGP-соседом, установите в секции настройки маршрутизатора BGP опцию `bfd` для соседа:

```
router bgp 2
no bgp ebgp-requires-policy
neighbor 192.168.0.3 remote-as 3
neighbor 192.168.0.3 bfd
exit
```

Чтобы BFD-сессия поднялась, необходимо включить поддержку BFD в настройках процесса марш-

рутизации BGP с обеих сторон. Еще одним способом является создание BFD-соседа вручную.

BGP-сессия успешно устанавливается и обновления передаются даже в ситуации, когда BFD-сессия не была установлена. Однако, если BFD-сессия была установлена, а затем разорвана, то автоматически будет разорвана и соответствующая BGP-сессия (после чего будут предприняты попытки переустановить BGP-сессию).

```
angie-va# sho bfd peers
BFD Peers:
peer 192.168.0.3 local-address 192.168.0.2 vrf default interface ens33
ID: 2213541843
Remote ID: 0
Active mode
Status: down
Downtime: 5 second(s)
Diagnostics: neighbor signaled session down
Remote diagnostics: neighbor signaled session down
Peer Type: dynamic
RTT min/avg/max: 0/0/0 usec
Local timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms
Echo transmission interval: disabled
Remote timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms

angie-va# sho ip bg su

IPv4 Unicast Summary:
BGP router identifier 2.2.2.2, local AS number 2 VRF default vrf-id 0
BGP table version 0
RIB entries 0, using 0 bytes of memory
Peers 1, using 24 KiB of memory

Neighbor        V         AS   MsgRcvd   MsgSent   TblVer  InQ OutQ  Up/Down State/
↪PfxRcd   PfxSnt Desc
192.168.0.3     4          3        27        30        0    0    0 00:00:06
↪  0        0 N/A

Total number of neighbors 1
```

#### 4.2.5.4 Настройка с VRRP

На данный момент нет интеграции с BFD.

#### 4.2.5.5 Настройка со статическими маршрутами

Angie ADC поддерживает условное добавление статического маршрута (в зависимости от BFD-статуса соседа, который указан в качестве next-hop). То есть, если BFD-сессия до устройства, адрес которого указан как next-hop, поднимается, то статический маршрут добавляется в таблицу маршрутизации. Если сессия остается в статусе `down` (или падает в процессе эксплуатации позднее), маршрут не добавляется (удаляется) из таблицы маршрутизации.

Добавить статический маршрут с проверкой доступности next-hop по BFD можно, например, с помощью следующей команды:

```
ip route 3.3.3.3/32 192.168.0.3 bfd
```

Сразу после создания такого статического маршрута соответствующая BFD-сессия находится в статусе `down`:

```
angie-va# sho bfd pee
BFD Peers:
peer 192.168.0.3 vrf default
ID: 1753656330
Remote ID: 0
Active mode
Status: down
Downtime: 6 second(s)
Diagnostics: ok
Remote diagnostics: ok
Peer Type: dynamic
RTT min/avg/max: 0/0/0 usec
Local timers:
Detect-multiplier: 3
Receive interval: 300ms
Transmission interval: 300ms
Echo receive interval: 50ms
Echo transmission interval: disabled
Remote timers:
Detect-multiplier: 3
Receive interval: 1000ms
Transmission interval: 1000ms
Echo receive interval: disabled
```

Чтобы сессия поднялась, на соседе нужно либо настроить собственный статический маршрут с указанием адреса исходящего интерфейса в качестве next-hop, либо вручную создать статический пир, соответствующий системе балансировки (см. ниже).

#### 4.2.5.6 Настройка независимого пира

Во всех приведенных выше случаях пир создавался динамически по требованию протокола динамической маршрутизации или в момент создания соответствующего статического маршрута. Ручное создание BFD-пира может потребоваться, например, в ситуации, когда необходимо поменять параметры работы какого-то одного соседа из группы автоматически обнаруженных.

Переход к созданию и настройке отдельного пира из режима конфигурирования протокола BFD:

```
angie-va# conf t
angie-va(config)# bfd
angie-va(config-bfd)# peer
A.B.C.D   IPv4 peer address
X:X::X:X  IPv6 peer address
angie-va(config-bfd)# peer
```

Некоторые настройки необходимо задать в момент создания пира (интерфейс, локальный адрес, multihop и vrf), остальные параметры можно будет изменить потом.

```
angie-va(config-bfd)# peer 192.168.0.3
<cr>
interface      Interface information
local-address  Configure local address
multihop       Configure multihop
vrf            Configure VRF
angie-va(config-bfd)# peer 192.168.0.3
```

```
angie-va(config-bfd-peer)#
detect-multiplier  Configure peer detection multiplier
echo               Configure peer echo intervals
echo-interval      Configure peer echo intervals
echo-mode          Configure echo mode
end                End current mode and change to enable mode
exit               Exit current mode and down to previous mode
find               Find CLI command matching a regular expression
list               Print command list
minimum-ttl        Expect packets with at least this TTL
no                 Negate a command or set its defaults
output             Direct vtysh output to file
passive-mode       Don't attempt to start sessions
profile            Use BFD profile settings
quit               Exit current mode and down to previous mode
receive-interval   Configure peer receive interval
shutdown           Disable BFD peer
transmit-interval  Configure peer transmit interval
```

После того, как с одной из сторон такой пир сконфигурирован, он будет находиться в статусе `down`, пока не будет создан симметричный пир с "противоположной" стороны.

#### 4.2.5.7 Профиль

Чтобы создать новый профиль, отредактировать или удалить существующий профиль, необходимо перейти из режима глобальной конфигурации в режим настройки протокола BFD, а дальше уже либо удалять профиль, либо переходить в режим создания или настройки конкретного профиля.

```
angie-va(config)# bfd
angie-va(config-bfd)#
end      End current mode and change to enable mode
exit     Exit current mode and down to previous mode
find     Find CLI command matching a regular expression
list     Print command list
no       Negate a command or set its defaults
output   Direct vtysh output to file
peer     Configure peer
profile  BFD profile.
quit     Exit current mode and down to previous mode
angie-va(config-bfd)# profile
BFDPROF  BFD profile name.
test
angie-va(config-bfd)# no profile test
angie-va(config-bfd)# profile test_new
angie-va(config-bfd-profile)#
detect-multiplier  Configure peer detection multiplier
echo               Configure peer echo intervals
echo-interval      Configure peer echo interval
echo-mode          Configure echo mode
end                End current mode and change to enable mode
exit               Exit current mode and down to previous mode
find               Find CLI command matching a regular expression
list               Print command list
minimum-ttl        Expect packets with at least this TTL
no                 Negate a command or set its defaults
output             Direct vtysh output to file
passive-mode       Don't attempt to start sessions
quit               Exit current mode and down to previous mode
```

```
receive-interval   Configure peer receive interval
shutdown           Disable BFD peer
transmit-interval  Configure peer transmit interval
angie-va(config-bfd-profile)#
```

### 4.2.6 Настройка RHI

RHI (Route Health Injection) позволяет динамически управлять маршрутами, анонсируемыми протоколами динамической маршрутизации, например BGP и OSPF, на основе данных о состоянии апстримов в балансировщике нагрузки. RHI позволяет автоматически отзывать префиксы при недоступности апстримов, минимизируя возможные циклы объявлений и отзывов. Апстрим считается недоступным, если все его бэкенды не прошли проверки работоспособности (health probes).

Для работы RHI необходимо:

1. Настроить BGP или OSPF в зависимости от свойств сети (через командную строку, примеры см. ниже).
2. Задать проверки работоспособности для балансировщика нагрузки (в веб-интерфейсе Angie ADC, пример см. ниже).
3. Задать конфигурацию RHI (в веб-интерфейсе Angie ADC, инструкцию см. ниже).

#### 4.2.6.1 Настройка BGP

Настройка BGP проводится через интерфейс командной строки.

##### Пример

```
hostname angie-va
no ipv6 forwarding
ip prefix-list myapp1_pair seq 5 permit 2.2.2.2/32
interface lo
 ip address 2.2.2.2/32
exit
router bgp 1
 no bgp ebgp-requires-policy
 neighbor 10.0.0.1 remote-as 2
 address-family ipv4 unicast
  redistribute connected route-map myapp1_pair
 exit-address-family
exit
route-map myapp1_pair permit 10
 match ip address prefix-list myapp1_pair
exit
```

##### Проверка

Angie ADC должен увидеть свои объявления:

```
angie-va# sh ip bgp
BGP table version is 14, local router ID is 2.2.2.2, vrf id 0
Default local pref 100, local AS 1
Status codes:  s suppressed, d damped, h history, u unsorted, * valid, > best, =␣
↪multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

```
    Network          Next Hop            Metric LocPrf Weight Path
*>  2.2.2.2/32       0.0.0.0                  0         32768 ?

Displayed 1 routes and 1 total paths
```

Соседний маршрутизатор должен увидеть объявления от Angie ADC:

```
abstract-router# sh ip bgp
BGP table version is 14, local router ID is 10.0.0.1, vrf id 0
Default local pref 100, local AS 2
Status codes:  s suppressed, d damped, h history, u unsorted, * valid, > best, =␣
↪multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

    Network          Next Hop            Metric LocPrf Weight Path
*>  2.2.2.2/32       10.21.20.6               0             0 1 ?

Displayed 1 routes and 1 total paths
```

В случае если объявления не видны, можно попробовать сбросить сессию BGP на обоих маршрутизаторах:

```
clear ip bgp * soft
```

#### Отзыв префиксов

Отзыв префиксов реализован через автоматическое изменение префикс-листов.

Например:

```
ip prefix-list myapp1_pair seq 5 deny 2.2.2.2/32
```

#### Объявление префиксов после восстановления работы апстрима

При восстановлении работоспособности апстримов префикс-листы не меняются автоматически.

Чтобы вручную вернуть префикс-лист к изначальному виду, выполните команды:

```
conf t
 ip prefix-list myapp1_pair seq 5 permit 2.2.2.2/32
```

**Примечание**

Если вы хотите, чтобы префиксы автоматически анонсировались после восстановления апстримов, установите флажок `Объявлять префиксы при восстановлении апстримов` в настройках конфигурации RHI.

### 4.2.6.2 Настройка OSPF

Настройка OSPF также проводится через интерфейс командной строки.

### Пример

```
hostname angie-va
no ipv6 forwarding
ip prefix-list myapp1_pair seq 5 permit 2.2.2.2/32
interface ens33
 ip address 192.168.0.20/24
 ip ospf 1 area 0
exit
interface lo
 ip address 2.2.2.2/32
exit
router ospf 1
 ospf router-id 192.168.0.20
 redistribute connected route-map myapp1_pair
exit
route-map myapp1_pair permit 10
match ip address prefix-list myapp1_pair
exit
```

### Проверка

Если серверы в бэкенде доступны:

```
angie-va#sho ip ospf 1 database external self-originate
OSPF Instance: 1
       OSPF Router with ID (2.2.2.2)
                AS External Link States
  LS age: 28
  Options: 0x2  : *|-|-|-|-|-|E|-
  LS Flags: 0xb
  LS Type: AS-external-LSA
  Link State ID: 2.2.2.2 (External Network Number)
  Advertising Router: 192.168.0.20
  LS Seq Number: 80000001
  Checksum: 0x139b
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

Если серверы в бэкенде недоступны:

```
angie-va#sho ip ospf 1 database external self-originate
OSPF Instance: 1
       OSPF Router with ID (192.168.0.20)
                AS External Link States
  LS age: 3600
  Options: 0x2  : *|-|-|-|-|-|E|-
  LS Flags: 0x8b
  LS Type: AS-external-LSA
  Link State ID: 2.2.2.2 (External Network Number)
  Advertising Router: 2.2.2.2
  LS Seq Number: 80000001
  Checksum: 0x139b
```

```
  Length: 36
  Network Mask: /32
        Metric Type: 2 (Larger than any link state path)
        TOS: 0
        Metric: 20
        Forward Address: 0.0.0.0
        External Route Tag: 0
```

#### Отзыв префиксов

Отзыв маршрута в OSPF происходит следующим образом: после удаления префикса OSPF изменяет значение возраста LSA в поле `LS age` на `3600`, информируя соседей, что LSA устарела и нужно перестать ее использовать. Через некоторое время эта LSA будет полностью удалена из LSDB.

#### Объявление префиксов после восстановления работы апстрима

При восстановлении работоспособности апстримов префикс-листы не меняются автоматически.

Чтобы вручную вернуть префикс-лист к изначальному виду, выполните команды:

```
conf t
 ip prefix-list myapp1_pair seq 5 permit 2.2.2.2/32
```

**Примечание**

Если вы хотите, чтобы префиксы автоматически анонсировались после восстановления апстримов, установите флажок `Объявлять префиксы при восстановлении апстримов` в настройках конфигурации RHI.

### 4.2.6.3 Настройка балансировщика нагрузки

Конфигурация балансировщика нагрузки настраивается в веб-интерфейсе Angie ADC (см. *инструкцию*).

Для работы RHI необходимо настроить следующие параметры:

- *upstream_probe*;
- параметры для получения информации о состоянии балансировщика нагрузки.

#### Пример

Приведены только строки, добавленные в дефолтную конфигурацию.

```
### Часть конфигурации опущена для краткости
}
### Бэкенды
upstream myapp1 {
    zone myapp1 1m;
    server server1;
    server server2;
}
### Переменная для health probe
map $upstream_status $good {
    200     "1";
}
server {
### Часть конфигурации опущена для краткости
```

```
    location / {
        proxy_pass http://myapp1;
        # Апстрим-пробы
        upstream_probe myapp1_probe
            interval=1s
            test=$good
            essential
            mode=always;
    }
### Часть конфигурации опущена для краткости
}
}
```

#### 4.2.6.4 Настройка конфигурации RHI

Конфигурация RHI настраивается в веб-интерфейсе Angie ADC.

Чтобы настроить конфигурацию, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC и перейдите в раздел `Управление трафиком` → `Конфигурация RHI`.

   Откроется окно настройки RHI.

2. Задайте префиксы. Для этого нажмите `Добавить` и укажите следующие параметры:

   - `VIP-адрес (CIDR)` — виртуальный IP-адрес для апстрима. Указывается в формате `X.X.X.X/mask`.

     **Примечание**

     Значение должно быть уникальным в рамках одного префикс-листа.

   - `Приоритет` — номер последовательности, определяющий порядок обработки правил. Маршрут с меньшим приоритетом обрабатывается первым.

     **Примечание**

     Значение должно быть уникальным в рамках одного префикс-листа.

   - `Имя префикс-листа` — имя префикс-листа для маршрутизации.
   - `Имя апстрима` — апстрим, для которого настраивается отзыв префикса.

3. Установите флажок `Объявлять префиксы при восстановлении апстримов`, если хотите, чтобы Angie ADC автоматически анонсировал префиксы в маршрутизируемую сеть при восстановлении работоспособности апстримов. По умолчанию флажок снят.

4. Нажмите `Сохранить`.

   Настройки RHI будут сохранены и сразу применены.

##### Удаление префиксов

Вы можете удалить неиспользуемые префиксы из конфигурации RHI, нажав на значок удаления напротив этих префиксов.

**Примечание**

При удалении префиксов через интерфейс Angie ADC они удаляются из базы данных и перестают влиять на объявление и отзыв маршрутов, но сохраняются в списках, отображаемых через командную строку.

Чтобы полностью удалить префикс-лист, необходимо выполнить следующие команды:

```
conf t
no ip prefix-list <имя префикс-листа>
```

Это поведение будет скорректировано в следующих релизах.

### 4.2.7 Настройка ECMP

В этой статье рассматривается:

- ECMP-распределение трафика со стороны сетевого оборудования между несколькими узлами Angie ADC;
- распределение трафика самой системой балансировки Angie ADC между несколькими путями в сторону клиентов или серверов.

В конце также будет рассмотрена возможность UCMP-балансировки.

ECMP (Equal-Cost Multi-Path) – это механизм, который позволяет равномерно распределять трафик между несколькими путями, если они имеют одинаковую стоимость (метрику). В контексте Angie ADC это означает, что входящий трафик может приходить на разные узлы Angie ADC, а также сама система балансировки Angie ADC может отправлять трафик через разные пути с одинаковой стоимостью.

Такой подход может использоваться:

- Если необходимо распределять трафик в сторону серверов или клиентов между несколькими доступными путями.
- При горизонтальном масштабировании Angie ADC, когда трафик распределяется между несколькими узлами Angie ADC с помощью сетевого оборудования, а затем узлы Angie ADC распределяют пользовательские сессии между конечными серверами.

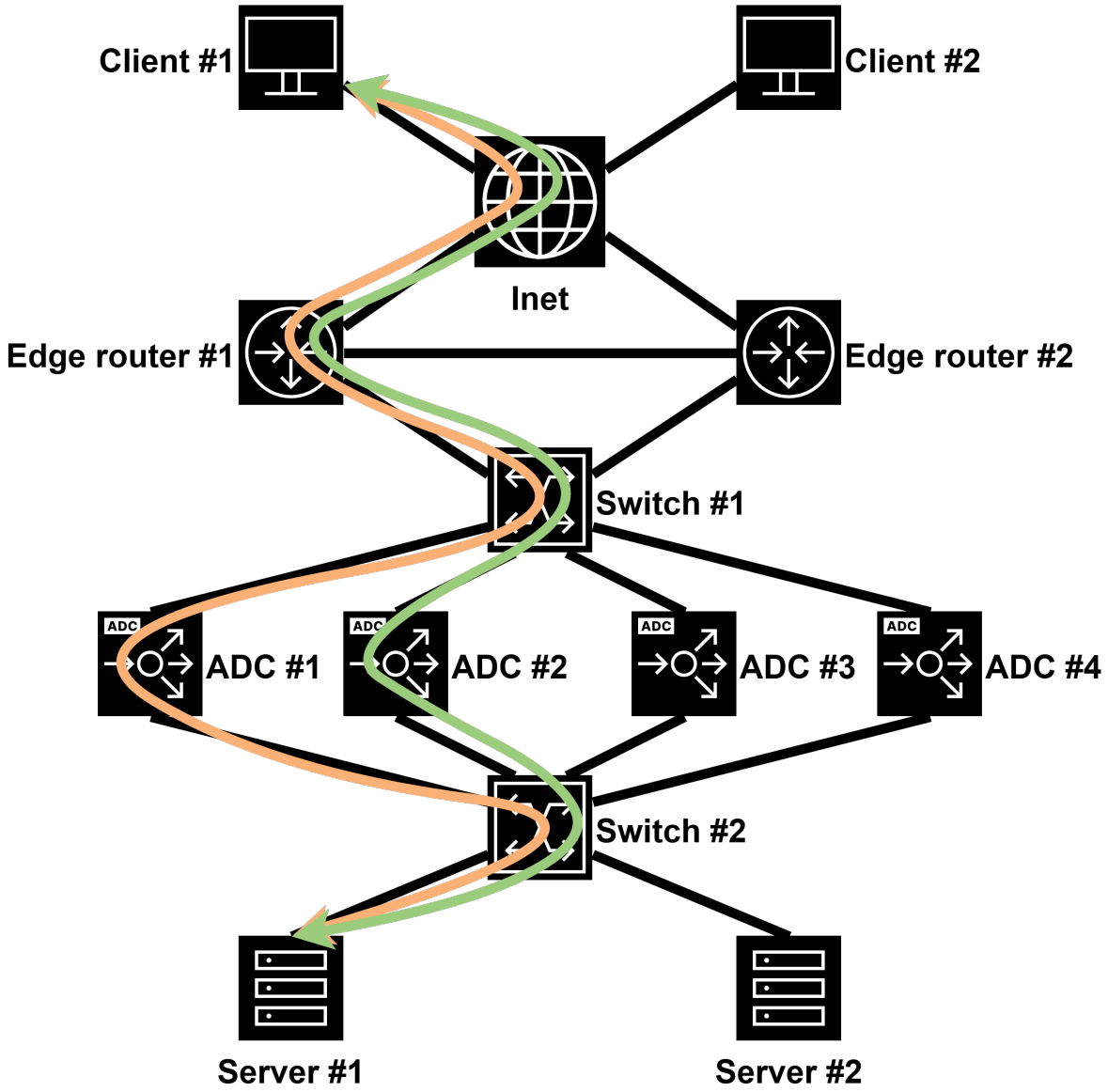
Angie ADC поддерживает ECMP-балансировку между 64 путями.

#### 4.2.7.1 Распределение трафика между несколькими Angie ADC

Чтобы трафик успешно передавался в Angie ADC по нескольким путям с одинаковой стоимостью, узлы Angie ADC должны «видеть» всю сессию полностью. То есть, весь прямой трафик в рамках одной сессии (первый поток) должен попадать на одну систему балансировки. Обратный трафик также должен возвращаться на нее, благодаря использованию адресов или пулов SNAT.

Трафик второй сессии (второй поток) может распределиться на другую систему балансировки, но будет полностью обрабатываться в ней.

Пример:


Client #1
Client #2
Inet
Edge router #1
Edge router #2
Switch #1
ADC
ADC #1
ADC #2
ADC #3
ADC #4
Switch #2
Server #1
Server #2

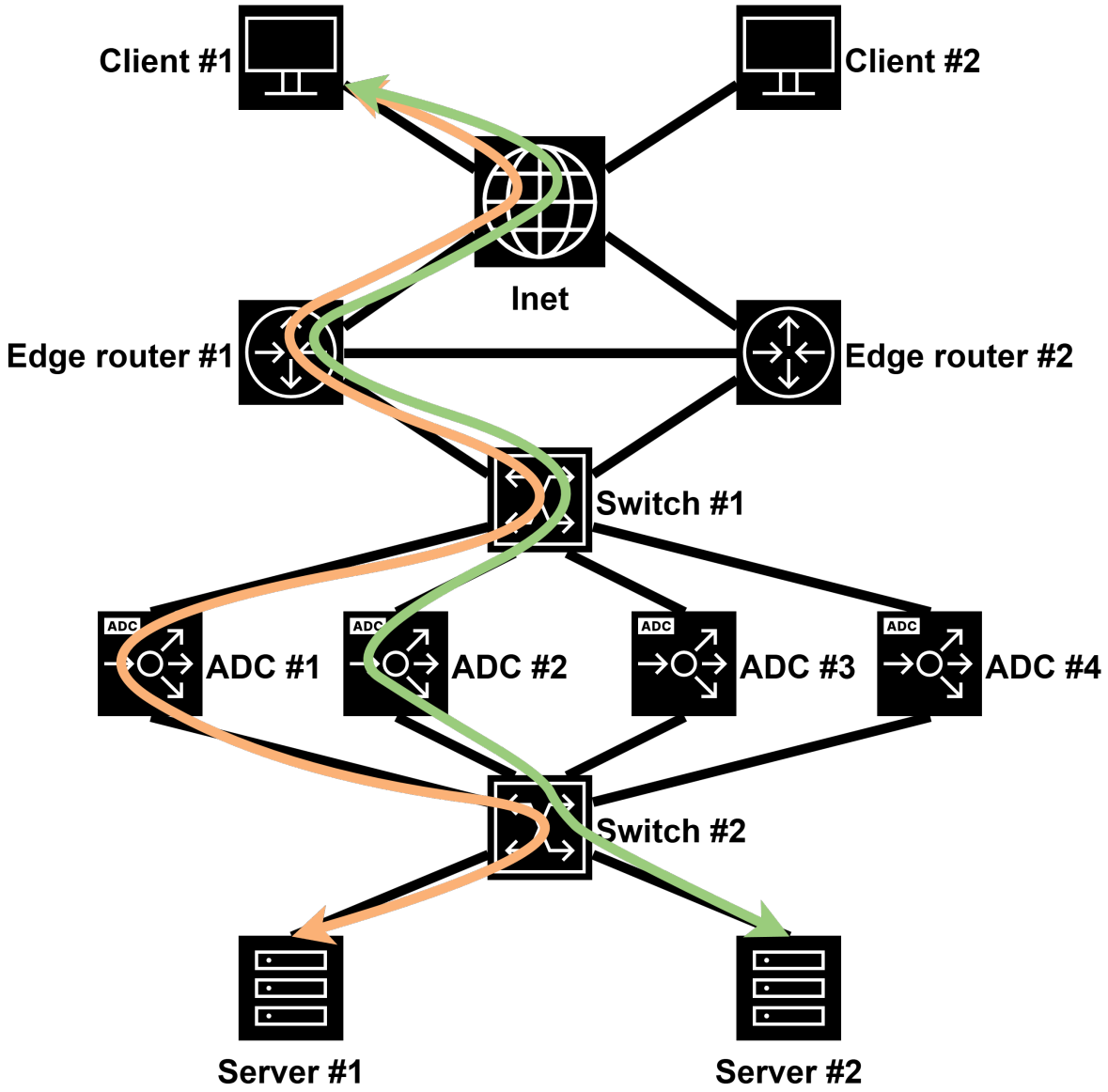



Рис. 1

Первая сессия показана оранжевой стрелкой, вторая – зеленой. Второй Angie ADC может выбрать тот же или другой бэкенд-сервер для обработки пользовательского запроса.

#### 4.2.7.2 Внешнее хранилище sticky

Когда критичным является распределение всех сессий от одного клиента на один бэкенд-сервер, используется механизм sticky. Angie ADC запоминает, куда было отправлено первое соединение от клиента, и все последующие подключения отправляет на этот же сервер. Если вторая сессия обрабатывается другим Angie ADC, который не обладает информацией о sticky, то для sticky можно использовать внешнее хранилище информации. В качестве такого хранилища может выступать используемый или выделенный узел Angie ADC.

Алгоритм использования внешнего хранилища для sticky будет таким:

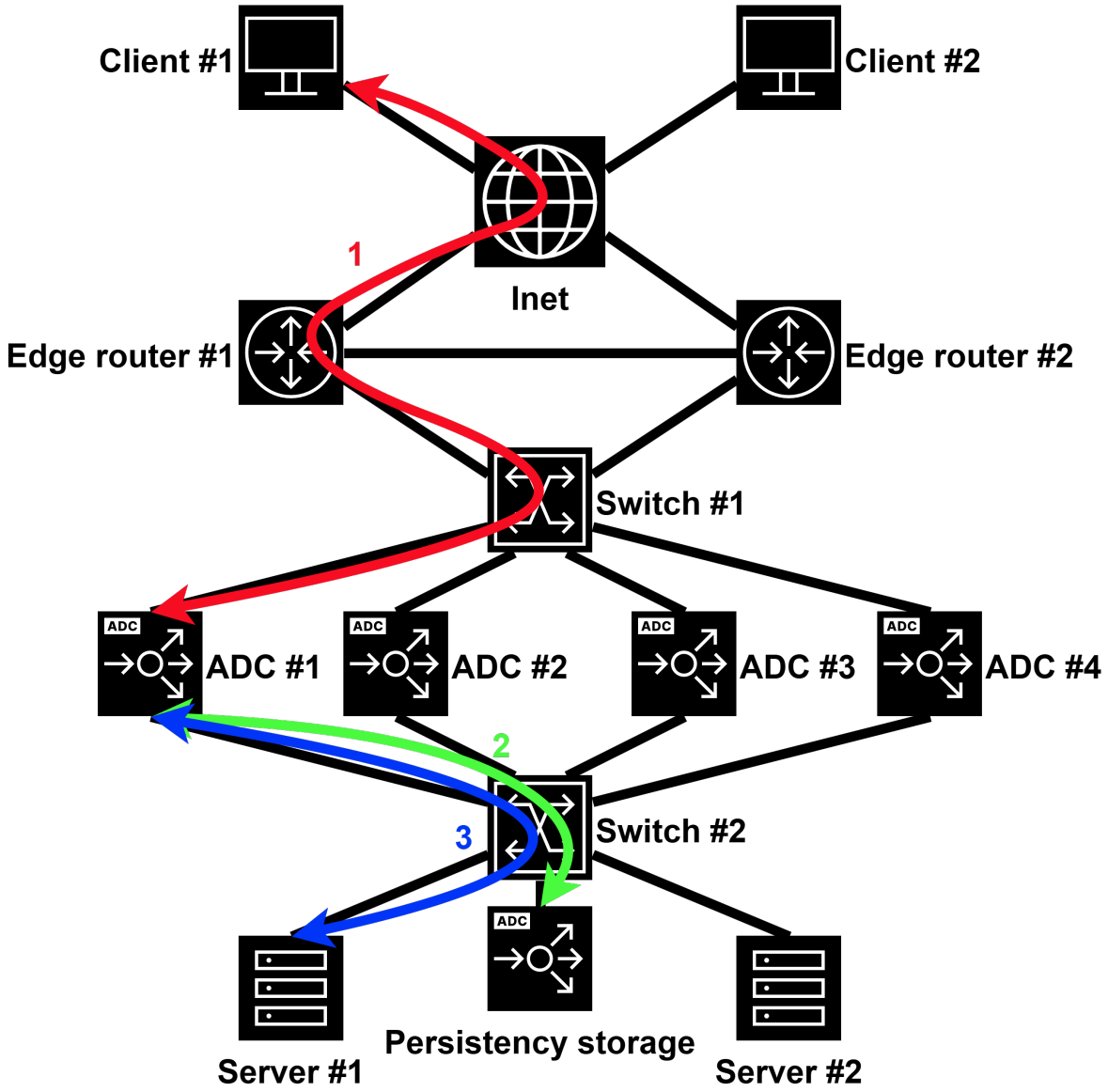
1. Пограничный маршрутизатор распределяет входящее подключение на один из узлов Angie ADC.
2. Получив пользовательский запрос, Angie ADC #1 выбирает сервер для перенаправления запроса (в соответствии с настроенным методом балансировки) и отправляет сообщение в хранилище sticky. Так как это первое подключение со стороны клиента, Angie ADC #1 сохраняет

sticky в хранилище и подтверждает правильность распределения.

3. Angie ADC #1 устанавливает вторую часть подключения (до бэкенд-сервера). Теперь пользовательское соединение полностью установлено и готово к работе.
4. Клиент устанавливает вторую сессию, которая попадает на Angie ADC #3.
5. Angie ADC #3 выбирает сервер для перенаправления запроса (в соответствии с настроенным методом балансировки) и отправляет сообщение в хранилище sticky. Если сервер, на который предлагается распределить нагрузку, совпадает с тем, который записан в хранилище, то хранилище подтверждает выбор системы балансировки. Если предложенный сервер не совпадает с записанным, то хранилище возвращает хэш-сумму «правильного» сервера.
6. Система балансировки Angie ADC #3 установит новую сессию в соответствии с ответом, полученным от хранилища.

Таким образом, клиент всегда будет устанавливать соединения с одним и тем же бэкенд-сервером.

Пример (стрелки пронумерованы в соответствии с шагами выше):



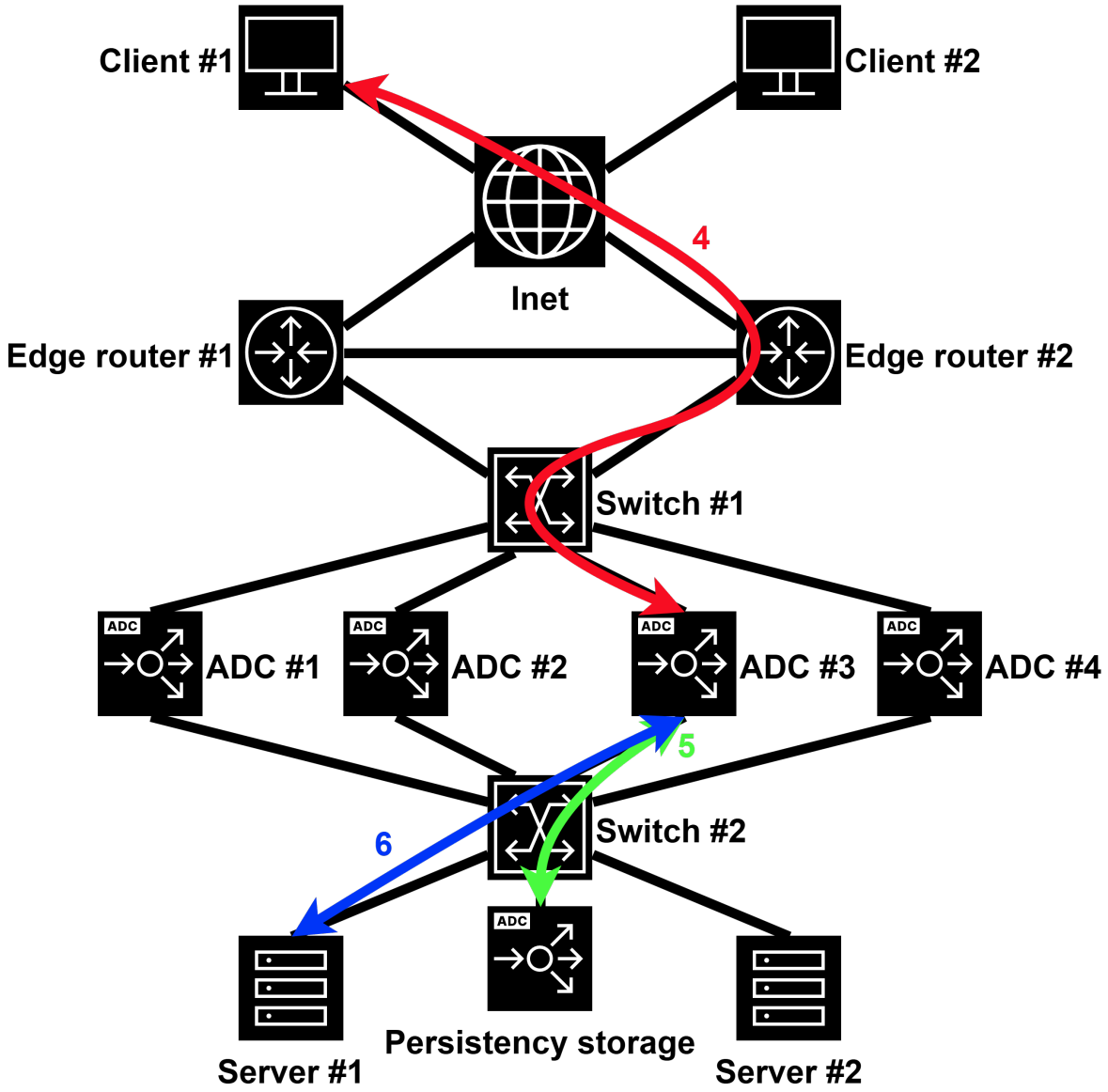



Рис. 2

#### 4.2.7.3 Распределение трафика между несколькими маршрутизаторами

Ниже приведен пример небольшой корпоративной сети, где из соображений отказоустойчивости или для преодоления ограничений производительности граничных маршрутизаторов реализована схема с двумя независимыми каналами в интернет, которые терминируются на двух независимых маршрутизаторах.

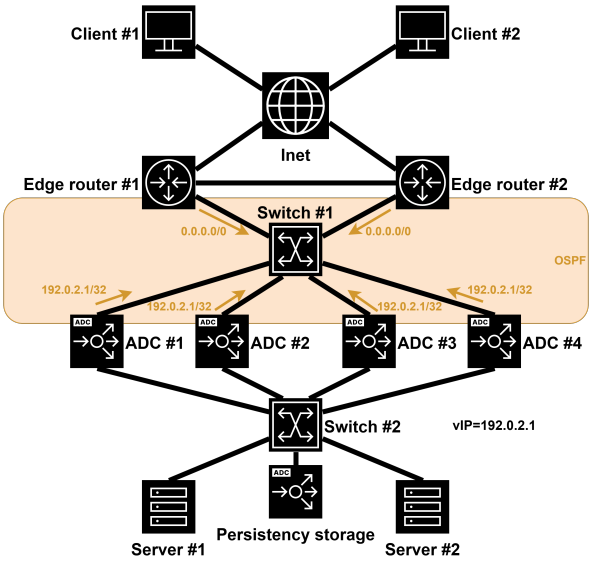



Рис. 3

Каждая из систем балансировки Angie ADC получает маршрут по умолчанию от каждого из граничных маршрутизаторов. Если параметры этих маршрутов одинаковы (для OSPF это типы маршрутов и метрики), то Angie ADC будет равномерно распределять трафик между граничными маршрутизаторами, то есть будет выполнять ECMP-балансировку.

Ниже приведен пример двух маршрутов по умолчанию, полученных с помощью протокола OSPF:

```
angie-adc1# sho ip ro os`
Codes: K - kernel route, C - connected, L - local, S - static,`
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,`
T - Table, v - VNC, V - VNC-Direct, F - PBR,`
f - OpenFabric, t - Table-Direct,`
> - selected route, * - FIB route, q - queued, r - rejected, b - backup`
t - trapped, o - offload failure`
O[1]>* 0.0.0.0/0 [110/1] via 192.168.0.201, ens33, weight 1, 00:00:32`
* via 192.168.0.202, ens33, weight 1, 00:00:32`
```

В операционной системе этот же маршрут виден так:

```
[root@angie-adc1 angie-adc1]# ip r
default nhid 23 proto ospf metric 20
nexthop via 192.168.0.202 dev ens33 weight 1
nexthop via 192.168.0.201 dev ens33 weight 1
```

#### 4.2.7.4 UCMP-балансировка

Некоторые протоколы динамической маршрутизации (например, EIGRP, BGP, IS-IS) поддерживают балансировку по путям с разной стоимостью – UCMP (Unequal Cost Multi-Path). В примере ниже приведен сценарий, реализованный с помощью BGP с использованием Extended Community Bandwidth:

```
angie-adc1# sho ip bgp
BGP table version is 23, local router ID is 192.168.0.153, vrf id 0
Default local pref 100, local AS 1
Status codes: s suppressed, d damped, h history, u unsorted, * valid, > best, =␣
↪multipath,
i internal, r RIB-failure, S Stale, R Removed
```

```
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
Network Next Hop Metric LocPrf Weight Path
*> 192.0.2.1/32 192.168.0.201 0 0 65002 i
*=   192.168.0.202 0 0 65002 i
Displayed 1 routes and 2 total paths
angie-adc1# sho ip bg 192.0.2.1/32
BGP routing table entry for 192.0.2.1/32, version 23
Paths: (2 available, best #1, table default)
Advertised to non peer-group peers:
192.168.0.201 192.168.0.201
2
192.168.0.202 from 192.168.0.202 (192.168.0.202)
Origin IGP, metric 0, valid, external, multipath, bestpath-from-AS 2, best (Older␣
↪Path)
Extended Community: LB:1:12500000 (100.000 Mbps)
Last update: Fri Feb 28 17:59:16 2025
2
192.168.0.201 from 192.168.0.201 (192.168.0.201)
Origin IGP, metric 0, valid, external, multipath
Extended Community: LB:1:6250000 (50.000 Mbps)
Last update: Fri Feb 28 17:59:16 2025
```

Пример тестировался на РЕД ОС версии 7.3. На данный момент UCMP не поддерживается этой операционной системой, то есть со стороны data-plane могут быть два исхода:

- маршрут будет установлен в RIB с балансировкой ECMP;
- маршрут не будет установлен в RIB, что зависит от других настроек протокола BGP.

```
angie-adc1# sho ip ro
Codes: K - kernel route, C - connected, L - local, S - static,
R - RIP, O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
T - Table, v - VNC, V - VNC-Direct, F - PBR,
f - OpenFabric, t - Table-Direct,
> - selected route, * - FIB route, q - queued, r - rejected, b - backup
t - trapped, o - offload failure
K>* 0.0.0.0/0 [0/100] via 192.168.0.2, ens33, src 192.168.0.153, 01:30:36
B>* 192.0.2.1/32 [20/0] via 192.168.0.201, ens33, weight 1, 00:42:48
*                       via 192.168.0.202, ens33, weight 1, 00:42:48
C>* 192.168.0.0/24 is directly connected, ens33, 01:30:36
L>* 192.168.0.153/32 is directly connected, ens33, 01:30:36
```

Если маршрут устанавливается в RIB, то со стороны операционной системы таблица маршрутизации будет выглядеть так:

```
[root@angie-adc1 etc]# ip r
default via 192.168.0.2 dev ens33 proto dhcp src 192.168.0.153 metric 100
192.0.2.1 nhid 37 proto bgp metric 20
nexthop via 192.168.0.202 dev ens33 weight 1
nexthop via 192.168.0.201 dev ens33 weight 1
192.168.0.0/24 dev ens33 proto kernel scope link src 192.168.0.153 metric 100
```

На данный момент Angie ADC не поддерживает UCMP-балансировку на уровне передачи данных (data-plane), только ECMP. Однако, если сетевая инфраструктура поддерживает UCMP, то такая схема распределения трафика может использоваться. UCMP-балансировка в таком случае может быть полезной, например, когда есть значительный перекос в мощностях систем балансировки, участвующих в обработке трафика:

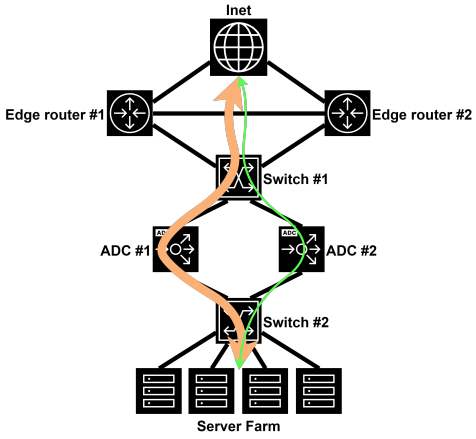

Inet
Edge router #1
Edge router #2
Switch #1
ADC #1
ADC #2
Switch #2
Server Farm

Рис. 4

ГЛАВА 5

# Мониторинг и диагностика

Здесь описываются:

- Просмотр статистики балансировщика и системных метрик, мониторинг в Console Light, SNMP и экспорт метрик — см. раздел *Метрики и статистика*.
- Мониторинг доступности узлов и бэкендов кластера — см. раздел *Доступность*.
- Диагностика, сбор логов и обращение в техническую поддержку — см. раздел *Диагностика и логирование*.

## 5.1 Метрики и статистика

Angie ADC предоставляет широкий спектр возможностей мониторинга:

- статистика балансировщика нагрузки *в веб-интерфейсе*;
- мониторинг балансировщика нагрузки *в веб-консоли Console Light* в режиме реального времени;
- системные метрики Angie ADC *в веб-интерфейсе*.

Дополнительно доступны:

- *отправка метрик по протоколу SNMP*;
- *экспорт метрик во внешний Prometheus*;
- модуль Prometheus;
- модуль Metric;
- метрики балансировщика нагрузки в API.

### 5.1.1 Статистика балансировщика нагрузки

Angie ADC предоставляет доступ к детализированным графикам со статистикой по балансировщику нагрузки. Просмотреть статистику можно в разделе `Мониторинг` → `Балансировщик`.

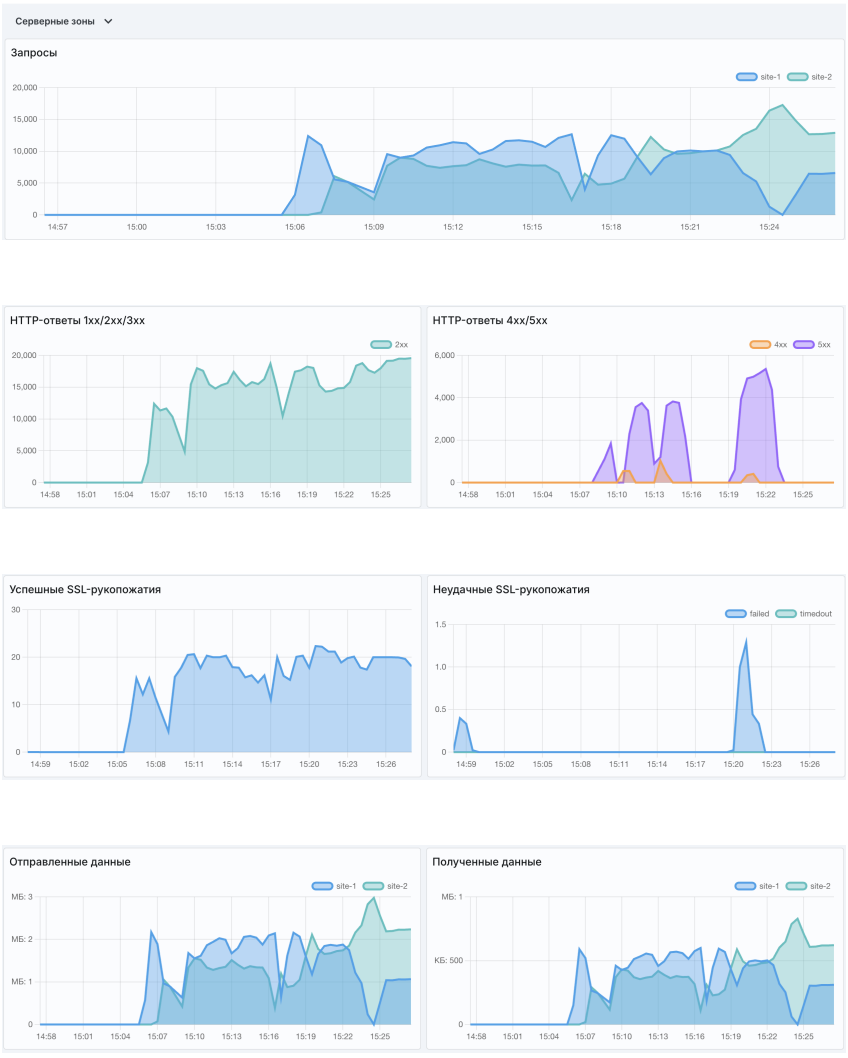
> **Примечание**
>
> Пользователь с ролью `Оператор` может просматривать только метрики тех зон, которые были выделены для него администратором *при предоставлении доступов* к конфигурации балансировщика нагрузки.

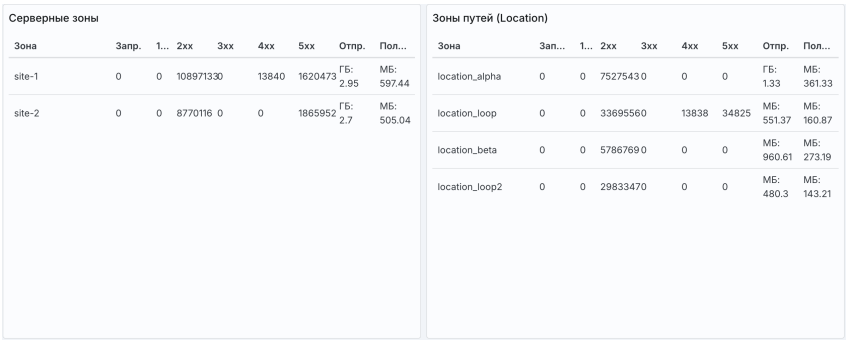
Кнопка `Перейти к интерактивному мониторингу` позволяет перейти в консоль Console Light и просмотреть ключевые показатели нагрузки и производительности сервера в реальном времени. Подробнее см. *Мониторинг в Console Light*. Кнопка доступна только для пользователей с ролями `Администратор` и `Супервизор`.

В раскрывающемся списке `За 30 минут` в верхней части экрана можно выбрать временной период для отображения статистики. Доступные периоды: 30 минут, 3 часа, 6 часов, 12 часов, 24 часа, 48 часов, неделя, две недели.

#### 5.1.1.1 Виджет "Серверные зоны"

Виджет `Серверные зоны` отображает статистику по серверным зонам разделяемой памяти:

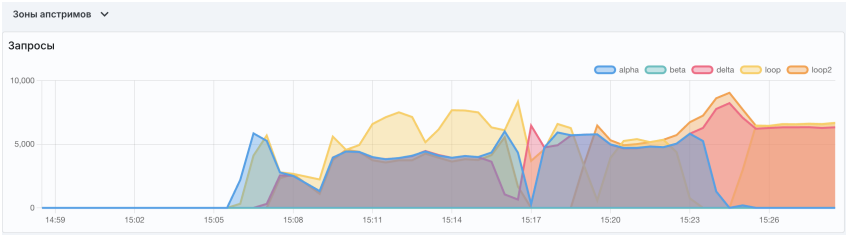


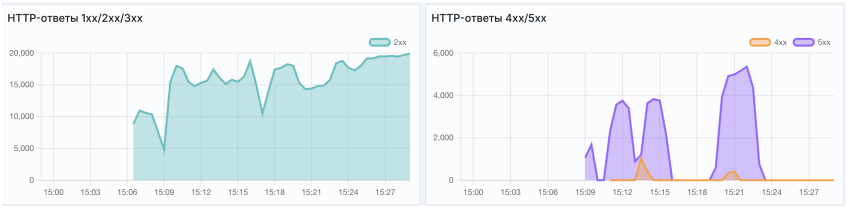


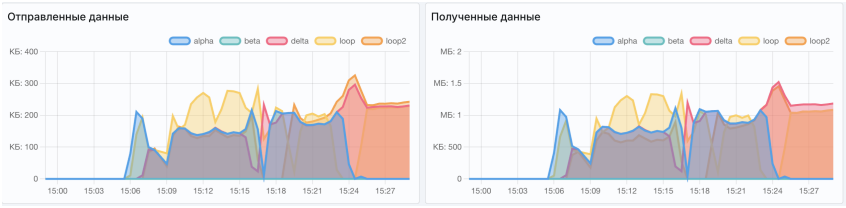

| | |
|---|---|
| `Запросы` | График числа запросов с разделением по отдельным серверам |
| `HTTP-ответы 1xx/ 2xx/3xx` | Графики числа HTTP-ответов с определенными кодами состояния с разделением по группам кодов (1xx, 2xx, 3xx) |
| `HTTP-ответы 4xx/5xx` | Графики числа HTTP-ответов с определенными кодами состояния с разделением по группам кодов (4xx, 5xx) |
| `Отправленные данные` | График объема отправленных данных с разделением по отдельным серверам |
| `Полученные данные` | График объема полученных данных с разделением по отдельным серверам |
| `Успешные SSL-рукопожатия` | График числа успешных SSL-рукопожатий с разделением по отдельным серверам |
| `Неудачные SSL-рукопожатия` | График числа неудачных SSL-рукопожатий с разделением по отдельным серверам |
| `Серверные зоны` | Таблица серверных зон со статистикой запросов, ответов и данных с разделением по отдельным зонам |
| `Зоны путей (Location)` | Таблица зон location со статистикой запросов, ответов и данных с разделением по отдельным location |

#### 5.1.1.2 Виджет "Зоны апстримов"

Виджет `Зоны апстримов` отображает статистику по зонам разделяемой памяти для апстримов:






| | |
|---|---|
| `Запросы` | График числа запросов с разделением по отдельным серверам |
| `HTTP-ответы 1xx/ 2xx/3xx` | Графики числа HTTP-ответов с определенными кодами состояния с разделением по группам кодов (1xx, 2xx, 3xx) |
| `HTTP-ответы 4xx/5xx` | Графики числа HTTP-ответов с определенными кодами состояния с разделением по группам кодов (4xx, 5xx) |
| `Отправленные данные` | График объема отправленных данных с разделением по отдельным серверам |
| `Полученные данные` | График объема полученных данных с разделением по отдельным серверам |

### 5.1.2 Мониторинг состояния системы

Просмотреть информацию о состоянии системы можно в веб-интерфейсе (`Мониторинг` → `Система`).

Системные метрики Angie ADC позволяют отслеживать состояние системы на разных уровнях:

- на уровне ОС (процессор, память, сеть);
- на уровне оборудования (только для аппаратного балансировщика Angie ADC: показания датчиков напряжения, температуры и пр.);
- на уровне хранения (информация о накопителях и их диагностика).

**Примечание**

На каждом узле Angie ADC отображаются только метрики этого узла.

Экран мониторинга отображает метрики системы и детализированные графики со статистикой за выбранный временной интервал.

В раскрывающемся списке `За 30 минут` в верхней части экрана вы можете изменить временной интервал для отображения статистики. Доступные временные интервалы: 30 минут, 3 часа, 6 часов, 12 часов, 24 часа, 48 часов, неделя, две недели.

#### 5.1.2.1 Сводная иконография

В верхней части экрана отображается сводная информация о состоянии системы.

| | |
|---|---|
| `RAM (free)` | Объем свободной оперативной памяти устройства (ГБ) |
| `ROM (free)` | Объем свободной постоянной памяти устройства (ГБ) |
| `Fan (avg.)` | Скорость вращения вентиляторов: среднее значение в оборотах в минуту (RPM) по всем вентиляторам устройства |
| `Temp. (avg.)` | Средняя температура по всем датчикам устройства (по Цельсию) |
| `Voltage` | Текущее напряжение блоков питания |

#### 5.1.2.2 Центральный процессор

| | |
|---|---|
| `Uptime` | Время непрерывной работы системы с момента последнего запуска. |
| `CPU (core)` | Количество ядер процессора. |
| `Load Avg.` | Усредненная загрузка системы за 1 минуту / 5 минут / 15 минут. Интерпретируется относительно числа ядер, например, на сервере с 4 ядрами значение 2.00 означает, что система загружена на 50%. |
| `CPU (load)` (график) | График использования процессора в процентах за выбранный временной интервал. |
| `Load Avg.` (график) | График усредненной загрузки системы за 1 минуту / 5 минут / 15 минут за выбранный временной интервал. Интерпретируется относительно числа ядер, например, на сервере с 4 ядрами значение 2.00 означает, что система загружена на 50%. |

#### 5.1.2.3 Обработка данных

| | |
|---|---|
| `Received` | Среднее количество данных, полученных в секунду (КБ/с) |
| `Transmitted` | Среднее количество данных, переданных в секунду (КБ/с) |
| `Collisions (avg.)` | Среднее количество коллизий в сети |
| `Input Packet Errors (avg.)` | Среднее количество ошибок входящих пакетов |
| `Output Packet Errors (avg.)` | Среднее количество ошибок исходящих пакетов |
| `Received` (график) | График среднего количества данных, полученных в секунду, за выбранный временной интервал с распределением по интерфейсам |
| `Transmitted` (график) | График среднего количества данных, переданных в секунду, за выбранный временной интервал с распределением по интерфейсам |

#### 5.1.2.4 Сетевые интерфейсы

Таблица `Interfaces` отображает список подключенных интерфейсов.

| | |
|---|---|
| `Interface` | Имя сетевого интерфейса |
| `Adminstate` | Состояние сетевого интерфейса, заданное администратором (`up` или `down`) |
| `Operstate` | Фактическое состояние сетевого интерфейса на данный момент (`up` или `down`) |
| `Collisions, %` | Процент коллизий за выбранный временной интервал |
| `Input Packet Errors, %` | Процентная доля принятых пакетов, в которых содержатся ошибки, за выбранный временной интервал |
| `Output Packet Errors, %` | Процентная доля переданных пакетов, в которых содержатся ошибки, за выбранный временной интервал |

#### 5.1.2.5 ОЗУ

| | |
|---|---|
| `Total` | Объем оперативной памяти, установленной на сервере (ГБ) |
| `Available` | Объем свободной оперативной памяти (ГБ) |
| `Used` | Объем используемой оперативной памяти (ГБ) |
| `ECC correctable errors` | Ошибки памяти, которые были обнаружены и автоматически исправлены (шт.) |
| `ECC uncorrectable errors` | Обнаруженные ошибки памяти, которые не удалось исправить (шт.) |
| `RAM (used)` | График использования оперативной памяти в процентах за выбранный временной интервал |

#### 5.1.2.6 Файловая система

Круговая диаграмма `Review file system` отображает заполнение файловой системы в процентах (`Used` — использованное пространство, `Free` — свободное пространство).

Таблица `File system` содержит список смонтированных файловых систем и их параметры.

| | |
|---|---|
| `filesystem` | Имя файловой системы |
| `type` | Тип файловой системы |
| `size` | Общий размер файловой системы |
| `used` | Объем использованного пространства |
| `avail` | Объем свободного пространства |
| `use, %` | Процент использования файловой системы |
| `mountpoint` | Точка монтирования файловой системы |

#### 5.1.2.7 Накопители (ROM)

Таблица `ROM (devices)` отображает состояние накопителей.

| | |
|---|---|
| `smartctl_status` | Состояние накопителя по данным SMART (Self-Monitoring, Analysis and Reporting Technology) |
| `power_on` | Общее время работы с момента включения |
| `temp` | Текущая температура |
| `size` | Объем накопителя |
| `device` | Имя устройства |
| `interface` | Тип интерфейса подключения |
| `model_name` | Модель устройства |
| `serial_number` | Серийный номер |

#### 5.1.2.8 S.M.A.R.T. атрибуты диска

Таблица `Reallocated_Sector_Ct` отображает количество переназначенных (перемещенных в резервную область) секторов диска:

| | |
|---|---|
| `device` | Имя устройства |
| `raw` | Необработанное значение атрибута SMART |
| `thresh` | Пороговое значение, при достижении которого состояние атрибута считается критическим |
| `value` | Нормализованное текущее значение атрибута |
| `worst` | Наихудшее нормализованное значение атрибута за выбранный временной интервал |

Таблица `Current_Pending_Sector` отображает количество секторов, которые считаются нестабильными и ожидают переназначения:

| | |
|---|---|
| `device` | Имя устройства |
| `raw` | Необработанное значение атрибута SMART |
| `thresh` | Пороговое значение, при достижении которого состояние атрибута считается критическим |
| `value` | Нормализованное текущее значение атрибута |
| `worst` | Наихудшее нормализованное значение атрибута за выбранный временной интервал |

#### 5.1.2.9 Датчики

Таблица `IPMI (temperature)` отображает список датчиков с температурами. Данные получены через интерфейс IPMI (Intelligent Platform Management Interface):

| | |
|---|---|
| `Sensor` | Идентификатор датчика температуры |
| `Temp.` | Текущее значение температуры |

Таблица `IPMI (fan)` отображает информацию о вентиляторах. Данные получены через интерфейс IPMI (Intelligent Platform Management Interface):

| | |
|---|---|
| `Sensor` | Идентификатор датчика вентилятора |
| `RPM` | Текущая скорость вращения вентилятора в оборотах в минуту (RPM) |

Таблица `IPMI (voltage)` отображает параметры напряжения. Данные получены через интерфейс IPMI (Intelligent Platform Management Interface):

| | |
|---|---|
| `Sensor` | Идентификатор датчика напряжения |
| `Voltage` | Текущее значение напряжения |

Таблица `IPMI (power state)` отображает состояние питания устройств. Данные получены через интерфейс IPMI (Intelligent Platform Management Interface):

| | |
|---|---|
| `Sensor` | Идентификатор датчика питания |
| `State` | Текущее состояние питания устройства (`0` — выключено, `1` — включено). |

Таблица `IPMI (sensor state)` отображает состояние всех остальных отслеживаемых датчиков, помимо перечисленных выше, в зависимости от типа оборудования. Данные получены через интерфейс IPMI (Intelligent Platform Management Interface):

| | |
|---|---|
| `Sensor` | Идентификатор датчика |
| `State` | Текущее значение датчика (`0` или `1`). |

### 5.1.3 Мониторинг в Console Light

Console Light — это консоль для мониторинга активности в реальном времени, отображающая ключевые показатели нагрузки и производительности сервера. Консоль основана на возможностях API-интерфейса Angie ADC; данные мониторинга активности генерируются в реальном времени.

Пример развернутой и настроенной консоли: https://console.angie.software/

При использовании в составе продукта Angie ADC консоль Console Light устанавливается и настраивается автоматически.

#### 5.1.3.1 Переход в Console Light

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Мониторинг` → `Балансировщик`.

   Откроется экран со статистикой для этого балансировщика.

2. В верхней части экрана нажмите кнопку `Перейти к интерактивному мониторингу`.

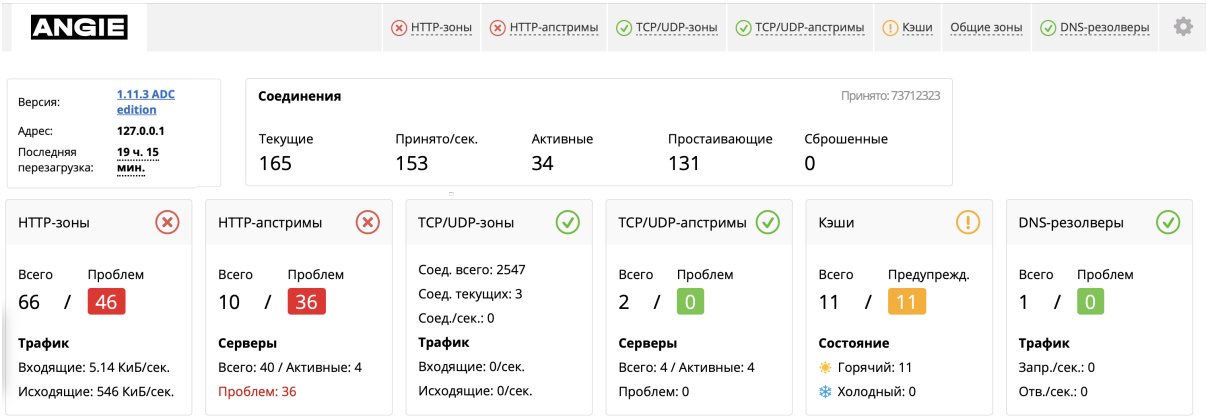
   Откроется облегченная консоль Console Light с данными мониторинга активности в реальном времени для этого балансировщика.

#### 5.1.3.2 Интерфейс

Console Light представляет собой единый экран с набором вкладок, каждая из которых содержит ряд виджетов с данными мониторинга.

> **Совет**
>
> В разделах ниже описания элементов интерфейса даны в порядке слева направо.

#### 5.1.3.3 Вкладка "Angie"




Это основная вкладка, где в сводном виде отображаются основные показатели мониторинга Angie ADC, сведенные на основе данных из нескольких разделов API.

> **Примечание**
>
> Виджеты со статистикой отображаются, если настроены соответствующие блоки в *конфигурации балансировщика нагрузки*.

##### Виджет "<версия>"

Здесь выводится номер версии Angie ADC, а также адрес сервера и время последней перезагрузки конфигурации.

##### Виджет "Соединения"

Здесь представлена основная статистика серверных соединений, формируемая на основе раздела API `/status/connections/`:

| | |
|---|---|
| `Текущие` | Текущее количество соединений |
| `Принято/сек.` | Число принимаемых за секунду соединений |
| `Активные` | Число активных соединений |
| `Простаивающие` | Число соединений в состоянии ожидания |
| `Сброшенные` | Количество сброшенных соединений |

Также доступно:

| | |
|---|---|
| `Принято` | Общее число соединений, принятых с последней перезагрузки сервера |

### Виджет "HTTP-зоны"

**Предупреждение**

Требует задать директиву adc__status_zone в контексте `server` или `location`.

Здесь представлена статистика зон разделяемой памяти контекста `http`, формируемая на основе раздела API `/status/http/server_zones/`:

| | |
|---|---|
| `Всего` | Общее количество зон |
| `Проблем` | Количество зон с какими-либо проблемами |
| `Трафик` | Скорость входящего и исходящего трафика в байтах в секунду |

### Виджет "HTTP-апстримы"

**Предупреждение**

Требует задать директиву adc__u_zone в блоке adc__u_upstream в контексте `http`.

Здесь представлена статистика апстримов контекста `http`, формируемая на основе раздела API `/status/http/upstreams/`:

| | |
|---|---|
| `Всего` | Общее количество апстримов |
| `Проблем` | Количество апстримов с какими-либо проблемами |
| `Серверы` | Статистика серверов с разделением по состоянию |

### Виджет "TCP/UDP-зоны"

**Предупреждение**

Требует задать следующие директивы:

- `status_zone` в контексте server или stream;
- `limit_conn` в контексте server или stream;
- adc__limit_conn_zone в контексте `stream`.

Пример:

```
stream {

    # ...
    limit_conn_zone $connection zone=limit-conn-stream:10m;

    server {

        # ...
        limit_conn limit-conn-stream 1;
        status_zone foo;
    }
}
```

Здесь представлена статистика зон разделяемой памяти контекста `stream`, формируемая на основе раздела API `/status/stream/server_zones/`:

| | |
|---|---|
| `Соед. всего` | Общее количество клиентских соединений |
| `Соед. текущих` | Текущее количество клиентских соединений |
| `Соед./сек.` | Количество обрабатываемых в секунду соединений |

#### Виджет "TCP/UDP-апстримы"

**Предупреждение**

Требует задать директиву adc__s_u_zone в блоке adc__s_u_upstream в контексте `stream`.

Здесь представлена статистика апстримов контекста `stream`, формируемая на основе раздела API `/status/stream/upstreams/`:

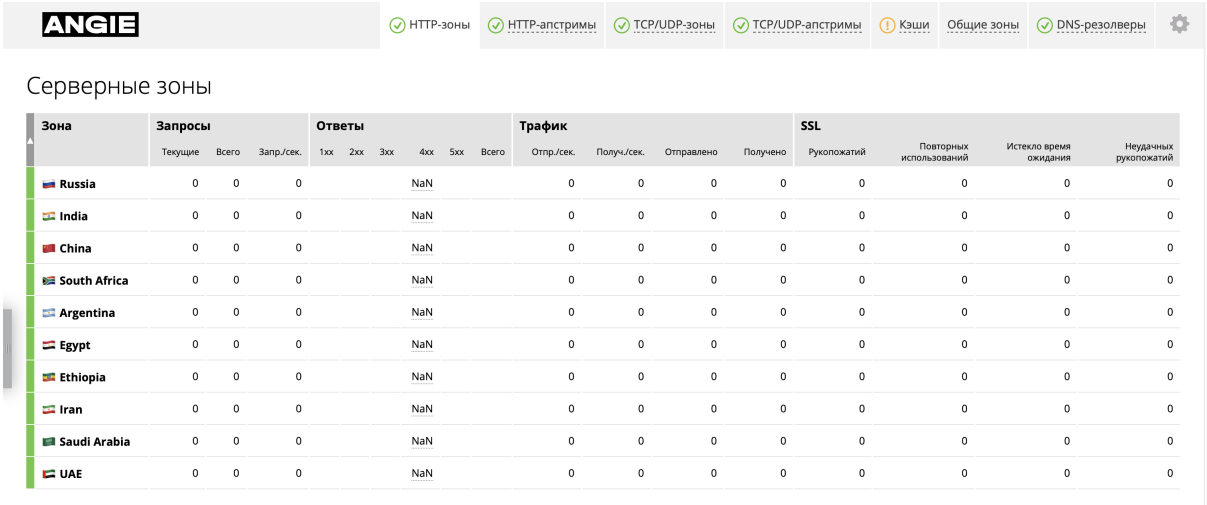
| | |
|---|---|
| `Всего` | Общее количество апстримов |
| `Проблем` | Количество апстримов с какими-либо проблемами |
| `Серверы` | Статистика серверов с разделением по состоянию |

### 5.1.3.4 Вкладка "HTTP-зоны"

**Предупреждение**

Требует задать директиву adc__status_zone в контексте `server` или `location`.

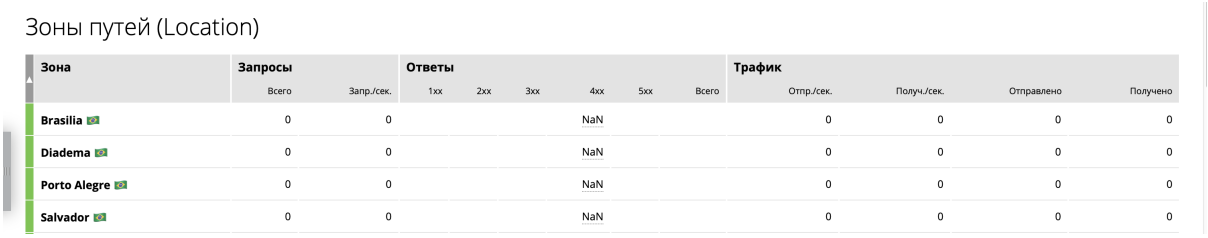
#### Раздел "Серверные зоны"




Здесь в сводном виде отображается статистика мониторинга зон разделяемой памяти для контекста `server` в `http`, формируемая на основе раздела API `/status/http/server_zones/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Зона`, чтобы отсортировать зоны по алфавиту или порядку в конфигурации. |
| `Запросы` | Общее количество запросов, а также число запросов в секунду |
| `Ответы` | Количество ответов с разбиением по кодам статуса, а также их общее количество |
| `Трафик` | Скорость исходящего и входящего трафика в байтах в секунду, а также общие объемы исходящего и входящего трафика в байтах |
| `SSL` | Суммарные показатели количества: успешных SSL-рукопожатий; повторных использований SSL-сессий; SSL-рукопожатий с истекшим таймаутом; неуспешных SSL-рукопожатий |

### Раздел "Зоны путей (Location)"


Зоны путей (Location)

| Зона | Запросы | | Ответы | | | | | | Трафик | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Всего | Запр./сек. | 1xx | 2xx | 3xx | 4xx | 5xx | Всего | Отпр./сек. | Получ./сек. | Отправлено | Получено |
| Brasilia | 0 | 0 | | | | NaN | | | 0 | 0 | 0 | 0 |
| Diadema | 0 | 0 | | | | NaN | | | 0 | 0 | 0 | 0 |
| Porto Alegre | 0 | 0 | | | | NaN | | | 0 | 0 | 0 | 0 |
| Salvador | 0 | 0 | | | | NaN | | | 0 | 0 | 0 | 0 |

Здесь в сводном виде отображается статистика мониторинга зон разделяемой памяти для контекста `location` в `http`, формируемая на основе раздела API `/status/http/location_zones/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Зона`, чтобы отсортировать зоны по алфавиту или порядку в конфигурации. |
| `Запросы` | Общее количество запросов, а также число запросов в секунду |
| `Ответы` | Количество ответов с разбиением по кодам статуса, а также их общее количество |
| `Трафик` | Скорость исходящего и входящего трафика в байтах в секунду, а также общие объемы исходящего и входящего трафика в байтах |

### Раздел "Зоны ограничения соединений (Limit Conn)"


Зоны ограничения соединений (Limit Conn)

| Зона | Передано | Отклонено | Сброшено | Пропущено |
|---|---|---|---|---|
| limit-conn-http | 0 | 0 | 0 | 0 |

Здесь приведена статистика зон `limit_conn` в контексте `http`, формируемая на основе раздела API `/status/http/limit_conns/`. Для каждой зоны представлены следующие данные:

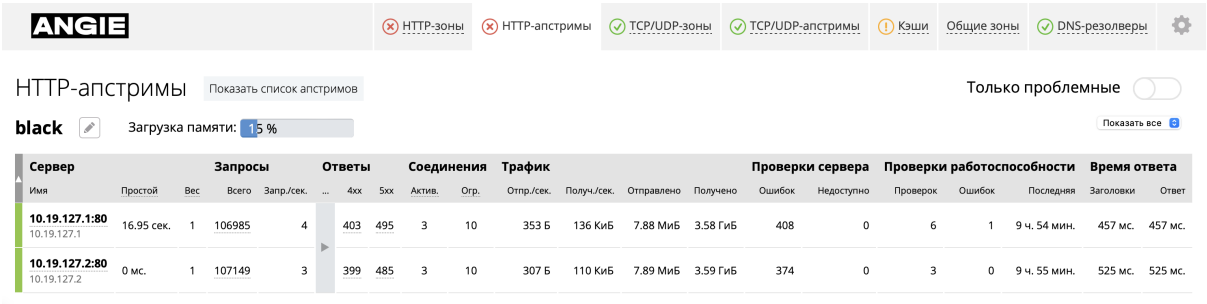
| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните значок рядом с пунктом `Зона`, чтобы открыть или закрыть график со следующими показателями. |
| `Передано` | Общее количество проксированных соединений |
| `Отклонено` | Общее количество сброшенных соединений |
| `Сброшено` | Общее количество соединений, сброшенных из-за переполнения хранилища зоны |
| `Пропущено` | Общее количество соединений, переданных с нулевым или превосходящим 255 байт ключом |

### Раздел "Зоны ограничения запросов (Limit Req)"


Здесь приведена статистика зон `limit_reqs` в контексте `http`, формируемая на основе раздела API `/status/http/limit_reqs/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните значок рядом с пунктом `Зона`, чтобы открыть или закрыть график со следующими показателями. |
| `Передано` | Общее количество проксированных соединений |
| `Задержано` | Общее количество задержанных соединений |
| `Отклонено` | Общее количество сброшенных соединений |
| `Сброшено` | Общее количество соединений, сброшенных из-за переполнения хранилища зоны |
| `Пропущено` | Общее количество соединений, переданных с нулевым или превосходящим 255 байт ключом |

### 5.1.3.5 Вкладка "HTTP-апстримы"




**Предупреждение**

Требует задать директиву adc__u_zone в блоке adc__u_upstream в контексте `http`.

На этой вкладке в сводном виде отображается статистика мониторинга апстримов контекста `http`, формируемая на основе раздела API `/status/http/upstreams/`. В режиме отладки также отображается процент загрузки памяти.

- Кнопка `Показать список апстримов` переключает краткий список апстримов с указанием числа проблемных апстримов и пиров.
- Переключатель `Только проблемные` переключает режим вывода статистики по проблемным апстримам.
- Раскрывающийся список с правой стороны таблицы каждого апстрима позволяет отфильтровать серверы в определенном состоянии (`Активные`, `Проблемные`, `На проверке`, `Выключенные`, `Занятые`, `Режим ожидания`, `Восстанавливается`, `Разгружаемые`).

Для каждого апстрима, помимо имени и загрузки зоны разделяемой памяти, представлены следующие данные:

| | |
|---|---|
| `Сервер` | Имена, общее время простоя и веса серверов апстрима<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Сервер`, чтобы отсортировать серверы по их состоянию или порядку в конфигурации. |
| `Запросы` | Общее количество и скорость обработки запросов |
| `Ответы` | Количество ответов с разбиением по кодам статуса |
| `Соединения` | Количество активных соединений и их максимальный предел, если он задан |
| `Трафик` | Скорость исходящего и входящего трафика в байтах в секунду, а также общие объемы исходящего и входящего трафика в байтах |
| `Проверки сервера` | Количество неуспешных обращений к серверу и число раз, когда сервер считался недоступным (объект `health` в API) |
| `Проверки работоспособности` | Общее количество проверок сервера, количество неуспешных проверок, а также время последней проверки |
| `Время ответа` | Время от начала запроса до отправки первого байта ответа; общее время от начала запроса до завершения отправки всего ответа (объект `health` в API) |

#### 5.1.3.6 Вкладка "TCP/UDP-зоны"

**Предупреждение**

Требует задать следующие директивы:

- `status_zone` в контексте server или stream;
- `limit_conn` в контексте server или stream;
- adc__limit_conn_zone в контексте `stream`.

Пример:

```
stream {

    # ...
    limit_conn_zone $connection zone=limit-conn-stream:10m;

    server {

        # ...
        limit_conn limit-conn-stream 1;
        status_zone foo;
    }
}
```

### Раздел "TCP/UDP-зоны"

TCP/UDP-зоны

| Зона | Соединения | | | Сессии | | | | Трафик | | | | SSL | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Текущих | Всего | Соед./сек. | 2xx | 4xx | 5xx | Всего | Отпр./сек. | Получ./сек. | Отправлено | Получено | Рукопожатий | Неудачных рукопожатий | Повторных использований |
| **sing_chorus** | 3 | 403 | 0 | 372 | 0 | 28 | 400 | 0 | 0 | 4.41 ГБ | 6.51 МБ | 375 | 0 | 180 |

Здесь в сводном виде отображается статистика мониторинга зон разделяемой памяти контекста `server` в `stream`, формируемая на основе раздела API `/status/stream/server_zones/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны |
| `Соединения` | Текущее и общее количество соединений, а также число соединений в секунду |
| `Сессии` | Количество сессий с разбиением по кодам статуса, а также их общее число |
| `Трафик` | Скорость исходящего и входящего трафика в байтах в секунду, а также общие объемы исходящего и входящего трафика в байтах |
| `SSL` | Суммарные показатели количества: успешных SSL-рукопожатий; неуспешных SSL-рукопожатий; повторных использований SSL-сессий |

### Раздел "Зоны ограничения соединений (Limit Conn)"

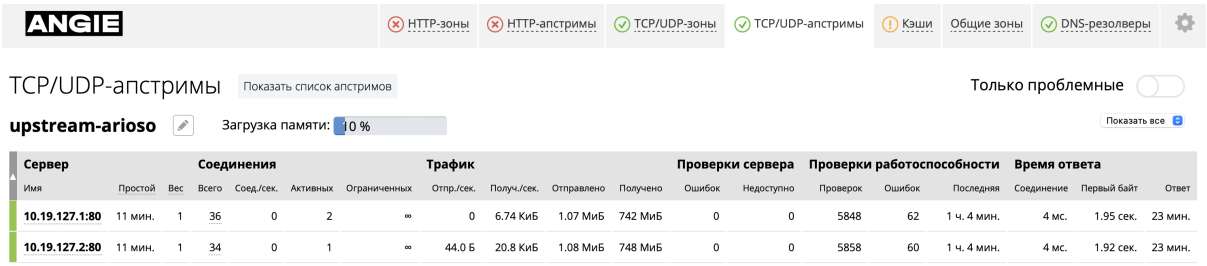
Зоны ограничения соединений (Limit Conn)

| | Зона | Передано | Отклонено | Сброшено | Пропущено |
|---|---|---|---|---|---|
| | **limit-conn-stream** | 403 | 0 | 0 | 0 |

Здесь приведена статистика зон `limit_conn` в контексте `stream`, формируемая на основе раздела API `/status/stream/limit_conns/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните значок рядом с пунктом `Зона`, чтобы открыть или закрыть график со следующими показателями. |
| `Передано` | Общее количество проксированных соединений |
| `Отклонено` | Общее количество сброшенных соединений |
| `Сброшено` | Общее количество соединений, сброшенных из-за переполнения хранилища зоны |
| `Пропущено` | Общее количество соединений, переданных с нулевым или превосходящим 255 байт ключом |

### 5.1.3.7 Вкладка "TCP/UDP-апстримы"




**Предупреждение**

Требует задать директиву adc__s_u_zone в блоке adc__s_u_upstream в контексте `stream`.

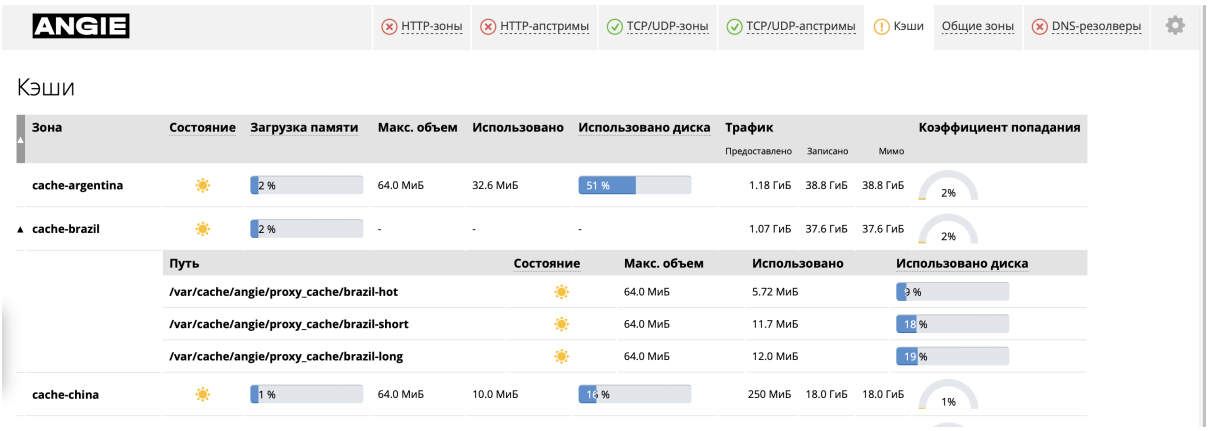
На этой вкладке в сводном виде отображается статистика мониторинга апстримов контекста `stream`, формируемая на основе раздела API `/status/stream/upstreams/`. В режиме отладки также отображается процент загрузки памяти.

- Кнопка `Показать список апстримов` переключает отображение краткого списка апстримов с указанием числа проблемных апстримов и пиров.
- Переключатель `Только проблемные` включает и отключает режим вывода статистики по проблемным апстримам.
- Раскрывающийся список с правой стороны таблицы каждого апстрима позволяет отфильтровать серверы в определенном состоянии (`Активные`, `Проблемные`, `На проверке`, `Выключенные`, `Занятые`, `Режим ожидания`, `Восстанавливается`, `Разгружаемые`).

Для каждого апстрима представлены следующие данные:

| | |
|---|---|
| `Сервер` | Имена, общее время простоя и веса серверов апстрима<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Сервер`, чтобы отсортировать серверы по их состоянию или порядку в конфигурации. |
| `Соединения` | Количество активных соединений и их максимальный предел, если он задан |
| `Трафик` | Скорость исходящего и входящего трафика в байтах в секунду, а также общие объемы исходящего и входящего трафика в байтах |
| `Проверки сервера` | Количество неуспешных обращений к серверу и число раз, когда сервер считался недоступным (объект `health` в API) |
| `Проверки работоспособности` | Общее количество проверок сервера, количество неуспешных проверок, а также время последней проверки |
| `Время ответа` | Время, затраченное на установку соединения с бэкендом; время от начала запроса до получения первого байта ответа; общее время, прошедшее от начала запроса до получения последнего байта ответа (объект `health` в API) |

### 5.1.3.8 Вкладка "Кэши"




**Предупреждение**

Требует задать директиву adc__proxy_cache_path в контексте `http`.

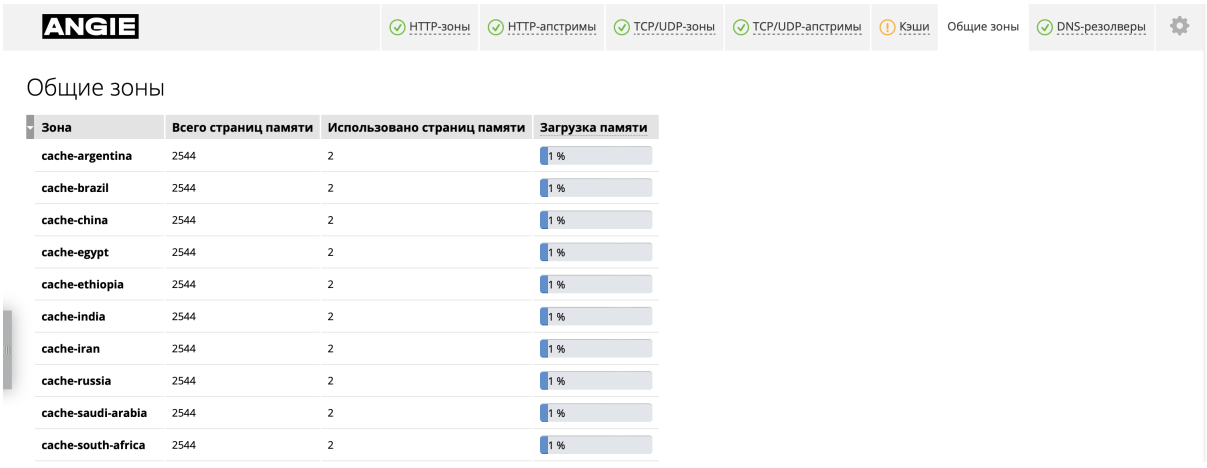
На этой вкладке в сводном виде отображается статистика мониторинга зон `proxy_cache` контекста `http`, формируемая на основе раздела API `/status/http/caches/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните значок рядом с пунктом `Зона`, чтобы открыть или закрыть списки шардов для всех зон, где они есть. |
| `Состояние` | Состояние кэша: холодный (метаданные загружаются в память) или горячий (метаданные загружены) |
| `Загрузка памяти` | Коэффициент использования памяти |
| `Макс. объем` | Максимальный объем кэша на диске в байтах; знак бесконечности, если объем кэша не ограничен |
| `Использовано` | Текущий объем кэша на диске в байтах |
| `Использовано диска` | Коэффициент использования диска |
| `Трафик` | Объем трафика (в байтах), переданного из кэша, записанного в кэш и возвращенного в обход кэша |
| `Коэффициент попадания` | Коэффициент попадания в кэш (отношение переданного из кэша трафика к общему объему) |

Для зоны с включенным шардингом значения `Макс. объем`, `Использовано` и `Использовано диска` отслеживаются отдельно для каждого шарда; в строке самой зоны выводятся прочерки, а зона обозначена как раскрывающийся список, в котором перечислены отдельные шарды:

| | |
|---|---|
| `Путь` | Путь шарда на диске |
| `Состояние` | Состояние шарда: холодный (метаданные загружаются в память) или горячий (метаданные загружены) |
| `Макс. объем` | Максимальный объем шарда на диске в байтах; знак бесконечности, если объем шарда не ограничен |
| `Использовано` | Текущий объем шарда на диске в байтах |
| `Использовано диска` | Коэффициент использования диска |

### 5.1.3.9 Вкладка "Общие зоны"




На этой вкладке в сводном виде отображается статистика мониторинга **всех** зон разделяемой памяти для всех контекстов. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Зона`, чтобы отсортировать зоны по размеру или порядку в конфигурации. |
| `Всего страниц памяти` | Общее количество страниц памяти |
| `Использовано страниц памяти` | Количество используемых страниц памяти |
| `Загрузка памяти` | Коэффициент использования памяти для зоны |

#### 5.1.3.10 Вкладка "DNS-резолверы"

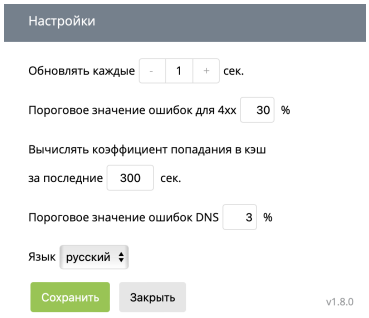

**Предупреждение**

Требует задать директиву adc__resolver в контексте `http`.

На этой вкладке в сводном виде отображается статистика запросов в зонах разделяемой памяти DNS, формируемая на основе раздела API `/status/resolvers/`. Для каждой зоны представлены следующие данные:

| | |
|---|---|
| `Зона` | Имя зоны<br>**Совет**<br>Щелкните стрелку рядом с пунктом `Зона`, чтобы отсортировать зоны по состоянию или порядку в конфигурации. |
| `Запросы` | Количество запросов типа A и AAAA, SRV, PTR |
| `Ответы` | Количество ответов с разделением по соответствующим кодам (`Успешные`, `Ошибок формата`, `Сервер не завершил запрос`, `Ошибок имени`, `Запрос не поддерживается`, `Запрос отклонен` и прочих) |

#### 5.1.3.11 Виджет "Настройки"



Позволяет настроить общие параметры консоли:

- Частоту обновления данных. Значение по умолчанию — 1 сек.
- Пороговое соотношение статусов `4xx`. По достижении порога в соответствующих разделах, посвященных ответам сервера, появляются "желтые" предупреждения. Значение по умолчанию — 7%.
- Временное окно для вычисления соотношения успешных попаданий в кэш. Значение по умолчанию — 300 сек.
- Порог учета ошибок для резолвера. По достижении указанного порога резолвер станет "красным". Значение по умолчанию — 3%.
- Язык интерфейса консоли. Доступные варианты: английский и русский. По умолчанию язык консоли выбирается на основе локали, установленной в браузере.

#### 5.1.3.12 Панель управления консолью

На всех вкладках в середине левой части страницы есть выдвигающаяся панель с двумя кнопками  . Верхняя приостанавливает и вновь запускает обновление данных из API, а нижняя позволяет обновить данные вручную, когда обновление приостановлено.

### 5.1.4 Экспорт метрик

Экспорт метрик из Angie ADC во внешний Prometheus выполняется с помощью механизма `federation`. Для экспорта доступны:

- системные метрики Node Exporter;
- метрики Angie ADC.

#### 5.1.4.1 Экспорт метрик Node Exporter

Чтобы настроить сбор метрик от Node Exporter, необходимо добавить в конфигурационный файл внешнего Prometheus (блок `scrape_configs`) следующую конфигурацию:

```
scrape_configs:
  - job_name: adc_node
    honor_labels: false
    metrics_path: /federate
    params:
      match[]:
        - '{job="node"}'
    static_configs:
      - targets:
        - <адрес_ADC-инстанса1>:8080
        - <адрес_ADC-инстанса2>:8080
```

После обновления конфигурации необходимо перезапустить сервис Prometheus.

Для визуализации собранных данных можно использовать готовый дашборд Grafana: https://grafana.com/grafana/dashboards/1860-node-exporter-full/.

#### 5.1.4.2 Экспорт метрик балансировщика нагрузки

Чтобы настроить сбор метрик от балансировщика нагрузки, необходимо добавить в конфигурационный файл Prometheus (блок `scrape_configs`) следующую конфигурацию:

```
scrape_configs:
  - job_name: adc_angie
    honor_labels: false
    metrics_path: /federate
```

```
    params:
      match[]:
        - '{job="local-lb"}'
    static_configs:
      - targets:
        - <адрес_ADC-инстанса1>:8080
        - <адрес_ADC-инстанса2>:8080
```

После обновления конфигурации необходимо перезапустить сервис Prometheus.

Для визуализации собранных данных можно использовать готовый дашборд Grafana: https://grafana.com/grafana/dashboards/20719-angie-dashboard/.

#### 5.1.4.3 Экспорт системных метрик Angie ADC

```
scrape_configs:
  - job_name: hardware
    honor_labels: false
    metrics_path: /federate
    params:
      match[]:
        - '{job="node"}'
        - '{job="smartctl"}'
        - '{job="ipmi"}'
    static_configs:
      - targets:
        - <адрес_ADC-инстанса1>:8080
        - <адрес_ADC-инстанса2>:8080
```

Подробнее о системных метриках см. *Мониторинг состояния системы*.

### 5.1.5 SNMP

Angie ADC может встраиваться в инфраструктуру мониторинга корпоративного уровня (например, Zabbix, Nagios). Поддерживается отправка стандартных метрик операционной системы и собственных метрик Angie ADC (см. ниже) по протоколам SNMPv2c и SNMPv3 (UDP-порт `161`). Протоколы SNMPv2c и SNMPv3 могут работать одновременно. При использовании протокола SNMPv3 необходимо предварительно добавить *пользователя SNMP* в веб-интерфейсе Angie ADC, чтобы предоставить безопасный доступ для системы мониторинга к SNMP-метрикам. Настройки пользователей распространяются на все узлы *кластера*.

| Протокол | SNMPv2c | SNMPv3 |
|---|---|---|
| Передача данных | в открытом виде | в зашифрованном виде |
| Аутентификация трафика | не требуется | требуется |

**Примечание**

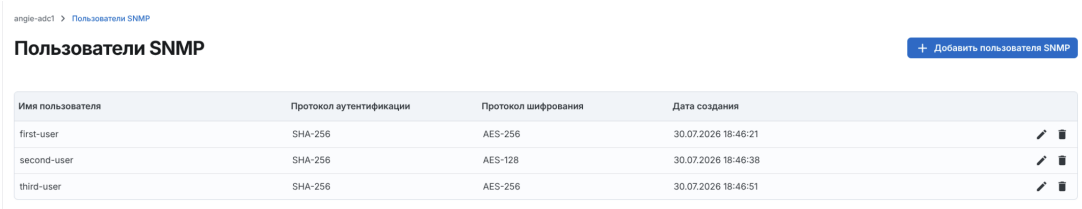
Для интеграции с системами мониторинга предоставляется MIB-файл, который можно скачать в веб-интерфейсе Angie ADC (`Мониторинг` → `Система` → `Скачать SNMP MIB-файл` в верхней правой части экрана).

#### 5.1.5.1 Пользователи SNMP

**Примечание**

Перед настройкой пользователей SNMP необходимо создать кластер Angie ADC. Подробнее см. в статье *Создание и масштабирование кластера управления*.

При использовании протокола SNMPv3 необходимо создать пользователей SNMP (т.е. те системы мониторинга, которые будут иметь доступ к данным).




**Примечание**

Изменение списка пользователей SNMPv3 вызывает кратковременный перезапуск службы snmpd (1-2 секунды) на узлах кластера.

**Добавление пользователя SNMP:**

1. В веб-интерфейсе Angie ADC перейдите в раздел `Мониторинг` → `Пользователи SNMP`.
2. Нажмите `Добавить пользователя SNMP`.
3. В открывшемся окне заполните поля формы для добавления нового пользователя:
   - `Имя пользователя` — уникальный идентификатор пользователя SNMPv3, например `zabbix-user`.
   - `Секрет аутентификации` — зашифрованный секрет аутентификации (AES-256-GCM, Base64).
   - `Протокол шифрования` — поддерживаются AES128 и AES256.
   - `Секрет шифрования` — зашифрованный секрет шифрования (AES-256-GCM, Base64).
4. Нажмите `Добавить`.

**Просмотр пользователей SNMP:**

1. В веб-интерфейсе Angie ADC перейдите в раздел `Мониторинг` → `Пользователи SNMP`.
2. В открывшемся окне отображаются добавленные пользователи SNMP.

   Для каждого пользователя отображается следующая информация:
   - Имя пользователя — уникальный идентификатор пользователя SNMPv3.
   - Протокол аутентификации — всегда SHA256.
   - Протокол шифрования — поддерживаются AES128 и AES256.
   - Дата создания — Дата и время создания записи.

**Редактирование и удаление пользователей SNMP:**

1. В веб-интерфейсе Angie ADC перейдите в раздел `Мониторинг` → `Пользователи SNMP`.
2. В открывшемся окне отображаются добавленные пользователи SNMP.

3. Нажмите значок редактирования / удаления напротив нужного пользователя.

#### 5.1.5.2 Метрики операционной системы

Поддерживаются все системные метрики `Net-SNMP`.

Пример запроса объема доступной физической памяти:

```
snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.2021.4.6.0
```

Пример ответа:

```
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 743028 kB
```

Ниже приведены примеры самых распространенных метрик, которые можно отслеживать:

| Метрика | OID | Тип | Описание |
|---|---|---|---|
| diskIOLoad.1 | .1.3.6.1.4.1.2021.13.1 | Inte | Процент времени, когда диск был занят выполнением операций ввода-вывода. |
| diskIOReads.1 | .1.3.6.1.4.1.2021.13.1 | Cou | Количество операций чтения с диска. |
| diskIOWrites.1 | .1.3.6.1.4.1.2021.13.1 | Cou | Количество операций записи на диск. |
| dskErrorFlag.1 | .1.3.6.1.4.1.2021.9.1. | Inte | Ошибка диска: становится `1`, если место на диске заканчивается. |
| dskPercent.1 | .1.3.6.1.4.1.2021.9.1.9 | Inte | Процент занятого места на основном разделе. |
| ifInDiscards.X | .1.3.6.1.2.1.2.2.1.13.X | Cou | Количество входящих пакетов, отброшенных из-за переполнения буферов. |
| ifInErrors.X | .1.3.6.1.2.1.2.2.1.14.X | Cou | Ошибки сети. Указывает на входящие пакеты с ошибками или проблемы на физическом уровне. |
| ifInOctets.X | .1.3.6.1.2.1.2.2.1.10.X | Cou | Входящий трафик. Объем полученных данных на сетевом интерфейсе. |
| ifOutOctets.X | .1.3.6.1.2.1.2.2.1.16.X | Cou | Исходящий трафик. Объем переданных данных через сетевой интерфейс. |
| hrSystemProcesses.0 | .1.3.6.1.2.1.25.1.6.0 | Gau | Общее количество запущенных процессов. Резкий рост может указывать на зависшие скрипты. |
| hrSystemUptime.0 | .1.3.6.1.2.1.25.1.1.0 | Tick | Время непрерывной работы сервера. Используется для обнаружения перезагрузки сервера. |
| laLoad.1 | .1.3.6.1.4.1.2021.10.1 | Stri | Средняя нагрузка (load average) на CPU за одну минуту. |
| memAvailReal.0 | .1.3.6.1.4.1.2021.4.6.0 | Inte | Объем доступной физической памяти. |
| memCached.0 | .1.3.6.1.4.1.2021.4.15 | Inte | Память, используемая файловым кэшем для ускорения операций чтения. |
| memTotalFreeSwap.0 | .1.3.6.1.4.1.2021.4.4.0 | Inte | Свободный Swap. При снижении до `0` возможен запуск OOM Killer. |
| ssCpuIdle.0 | .1.3.6.1.4.1.2021.11.1 | Inte | Процент свободного ресурса. Близкое к `0` значение указывает на перегрузку CPU. |
| ssCpuRawWait.0 | .1.3.6.1.4.1.2021.11.5 | Cou | Время ожидания ввода-вывода. Рост значения указывает на проблемы с дисковой подсистемой или БД. |

#### 5.1.5.3 Собственные метрики Angie ADC

Angie ADC позволяет получать по SNMP собственные метрики из Prometheus. У собственных метрик будет базовый OID `.1.3.6.1.4.1.2021.50`.

Пример запроса метрики базовой доступности сервиса `up`:

```
snmpget -v2c -c public 127.0.0.1 1.3.6.1.4.1.2021.50.2.1
```

Ответ:

```
UCD-SNMP-MIB::ucdavis.50.2.1 = Gauge32: 1
```

Ответ `1` означает, что сервис доступен.

Пример запроса всего дерева собственных метрик:

```
snmpwalk -v2c -c public 127.0.0.1 1.3.6.1.4.1.2021.50
```

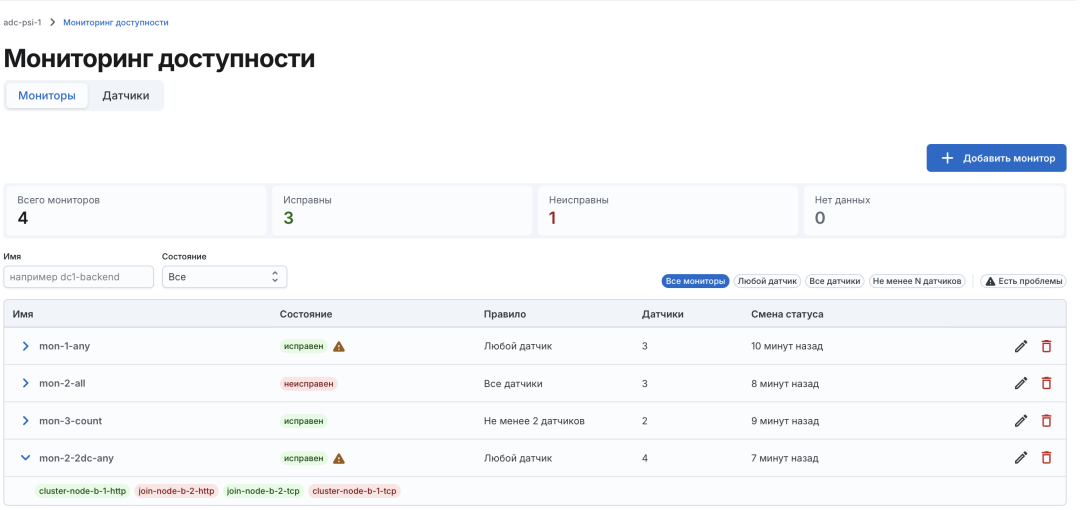
Если Prometheus не доступен или метрик нет, в ответе может быть `NONE` или ошибка.

Ниже приведены метрики Prometheus, поддерживаемые по SNMP на данный момент. Список поддерживаемых метрик будет пополняться.

| Метрика из Prometheus | OID | Тип | Описание |
|---|---|---|---|
| angie_connections_ | .1.3.6.1.4.1.2021.50.1 | Cou | Принятые соединения — мониторинг общего потока входящих подключений. |
| angie_connections_ | .1.3.6.1.4.1.2021.50.1 | Gau | Активные сессии — мониторинг количества пользователей на сайте в данную секунду. |
| angie_connections_ | .1.3.6.1.4.1.2021.50.1 | Cou | Сброшенные соединения — рост этого показателя указывает на достижение лимитов сервера. |
| angie_http_request | .1.3.6.1.4.1.2021.50.1 | Cou | Всего HTTP-запросов — позволяет рассчитать RPS (количество запросов в секунду). |
| go_goroutines | .1.3.6.1.4.1.2021.50.2 | Gau | Потоки — мониторинг здоровья Go-приложения (защита от утечек памяти). |
| go_memstats_alloc | .1.3.6.1.4.1.2021.50.2 | Gau | Память процесса — позволяет отслеживать, сколько оперативной памяти выделено и используется Go-процессом сейчас. |
| scrape_duration_se | .1.3.6.1.4.1.2021.50.2 | Gau | Время сбора данных в секундах — позволяет оценить производительность. |
| scrape_samples_scr | .1.3.6.1.4.1.2021.50.2 | Gau | Количество полученных строк метрик. Резкое падение показателя может означать, что часть метрик потеряна. |
| up | .1.3.6.1.4.1.2021.50.2 | Gau | Доступность — базовая health-метрика, `1` — сервис доступен, `0` — недоступен. |

## 5.2 Доступность

Мониторинг доступности позволяет отслеживать доступность узлов и бэкендов в кластере Angie ADC.



Для отслеживания состояния бэкендов на узлах кластера настраиваются датчики, выполняющие проверки работоспособности бэкендов. Далее можно создать мониторы, которые будут объединять несколько датчиков и по правилу агрегации определять общий статус доступности объектов.

Датчики поддерживают следующие типы проверок работоспособности бэкендов:

| | |
|---|---|
| `HTTP` | Активная проверка по протоколу HTTP. |
| `TCP` | Активная проверка по протоколу TCP. |
| `ICMP` | Проверка доступности с помощью ICMP Echo Request (ping). |
| `Angie status` | Проверка на основе данных о состоянии бэкендов, полученных через Status API Angie. Подходит для случаев, когда бэкенды не имеют собственного URL для проверки. |

Мониторы поддерживают следующие типы правил агрегации датчиков:

| | |
|---|---|
| `Любой датчик` | Монитор исправен, если работает хотя бы один датчик. |
| `Все датчики` | Монитор исправен, если работают все датчики. |
| `Не менее N датчиков` | Монитор исправен, если работают N и более датчиков. |

Состав датчиков в мониторе зависит от того, доступность чего необходимо отслеживать. В один монитор можно объединять как датчики одного узла, так и датчики с разных узлов кластера.

Примеры:

- **Определение доступности дата-центра для глобальной балансировки**: создайте монитор дата-центра и включите в него все датчики всех узлов этого дата-центра по правилу `Любой датчик`. Исправность монитора будет означать, что дата-центр в целом доступен для глобальной балансировки. Простой пример см. *Глобальная балансировка с учетом мониторов доступности*.
- **Определение доступности сервиса на нескольких узлах**: создайте монитор сервиса и включите в него все датчики этого сервиса со всех узлов, где он запущен, по правилу `Любой датчик`. Исправность монитора будет означать доступность сервиса в целом, не на конкретном узле.
- **Углубленная проверка одного бэкенда**: создайте монитор бэкенда и включите в него несколько датчиков этого бэкенда с разными типами проверок по правилу `Все датчики`. Исправность монитора будет означать работоспособность бэкенда сразу на нескольких уровнях.

В этом разделе:

- *Настройка датчиков и мониторов*
- *Мониторинг доступности*

### 5.2.1 Настройка датчиков и мониторов

Датчики и мониторы настраиваются в веб-интерфейсе Angie ADC (`Система` → `Доступность`).

Датчик выполняет проверку работоспособности бэкенда на выбранном узле кластера с заданной периодичностью и формирует состояние: доступен бэкенд или нет. Монитор объединяет несколько датчиков и по правилу агрегации отдает общий статус доступности приложения, узла или бэкенда в зависимости от набора датчиков.

В этой статье:

- *Добавление датчика*;
- *Добавление монитора*;
- *Редактирование датчика*;

- *Редактирование монитора*;
- *Удаление датчика*;
- *Удаление монитора*.

#### 5.2.1.1 Добавление датчика

Чтобы добавить датчик:

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Доступность`.
2. На вкладке `Датчики` нажмите `Добавить датчик`.
3. Задайте параметры датчика:

| | |
|---|---|
| `Имя ресурса` | Имя проверяемого ресурса в произвольной форме. Допускаются строчные латинские буквы, цифры и дефис, не более 63 символов. Имя ресурса после создания датчика изменить нельзя.<br><br>**Примечание**<br>Имя датчика формируется автоматически из имени узла и имени ресурса; итоговое имя датчика *{узел}-{ресурс}* не должно превышать 63 символа. Имя датчика уникально в пределах кластера. |
| `Узлы` | Узлы кластера, на которых выполняется проверка. Для каждого выбранного узла создается отдельный датчик. |
| `Включен` | Переключатель, включающий и выключающий датчик. По умолчанию датчик включен. |
| `Проверка` | Протокол проверки ресурса. Поддерживаются следующие типы проверок:<br>• `HTTP` — активная проверка по протоколу HTTP.<br>• `TCP` — активная проверка по протоколу TCP.<br>• `ICMP` — проверка доступности с помощью ICMP Echo Request (ping).<br>• `Angie status` — проверка на основе данных о состоянии бэкендов, полученных через Status API Angie. Подходит для случаев, когда бэкенды не имеют собственного URL для проверки.<br>Набор параметров зависит от типа проверки: |

#### Параметры проверки HTTP

| | |
|---|---|
| `URL` | URL, по которому выполняется запрос проверки, например `https://10.20.0.14/health`. |
| `Метод` | HTTP-метод запроса. Значение по умолчанию `GET`. |
| `Ожидаемые коды` | Коды ответа, при которых проверка считается успешной, например `200`. |
| `Заголовки запроса` | Заголовки, добавляемые к запросу проверки. Необязательный параметр. |
| `Ожидаемое содержимое ответа` | Подстрока, которую должно содержать тело ответа для успешного прохождения проверки. Необязательный параметр. |
| `Интервал проверки` | Как часто выполняется проверка. Значение по умолчанию `1s`. |
| `Таймаут` | Время ожидания ответа. Если за это время ответ не получен, проверка считается неуспешной. Значение по умолчанию `500ms`. |
| `Порог сбоя (проверок)` | Число подряд неуспешных проверок, после которого датчик переходит в состояние `не работает`. Значение по умолчанию `2`. |
| `Порог успеха (проверок)` | Число подряд успешных проверок, после которого датчик возвращается в состояние `работает`. Значение по умолчанию `2`. |
| `Устаревание (stale_afte` | Время, по истечении которого состояние без обновления считается устаревшим (`нет данных`). Значение по умолчанию `5s`. |

#### Параметры проверки TCP

| | |
|---|---|
| `Хост` | IP-адрес или имя проверяемого сервиса. |
| `Порт` | Порт, на котором выполняется проверка. |
| `Интервал проверки` | Как часто выполняется проверка. Значение по умолчанию `1s`. |
| `Таймаут` | Время ожидания ответа. Если за это время ответ не получен, проверка считается неуспешной. Значение по умолчанию `500ms`. |

4. Нажмите `Создать`.

   Для каждого выбранного узла будет создан датчик. В таблице датчиков для каждого датчика будет отображаться результат проверки в столбце `Состояние` (`работает`, `не работает` или `нет данных`).

#### 5.2.1.2 Добавление монитора

Чтобы добавить монитор:

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Доступность`.
2. На вкладке `Мониторы` нажмите `Добавить монитор`.
3. Задайте параметры монитора:

| | |
|---|---|
| `Имя` | Имя монитора. Допускаются строчные латинские буквы, цифры и дефис, не более 63 символов. Должно быть уникальным. Имя монитора после создания изменить нельзя. |
| `Правило агрегации` | Определяет, при каком состоянии датчиков монитор имеет статус `исправен`:<br>• `Любой датчик` — монитор исправен, если работает хотя бы один датчик.<br>• `Все датчики` — монитор исправен, если работают все датчики.<br>• `Не менее N датчиков` — монитор исправен, если работают N и более датчиков. Значение N не может превышать число датчиков монитора. |
| `Датчики` | Имена датчиков, входящих в монитор. Нужно указать хотя бы один датчик. |

4. Нажмите `Добавить`.

В таблице мониторов отобразится добавленный монитор. Для каждого монитора отображается его статус (`исправен`, `неисправен` или `нет данных`).

#### 5.2.1.3 Редактирование датчика

Чтобы изменить датчик, на вкладке `Датчики` нажмите значок редактирования в строке датчика. В окне `Редактировать датчик` можно изменить тип и параметры проверки, периодичность и пороги. Имя датчика изменить нельзя.

#### 5.2.1.4 Редактирование монитора

Чтобы изменить монитор, на вкладке `Мониторы` нажмите значок редактирования в строке монитора. В окне `Редактировать монитор` можно изменить правило агрегации и список датчиков. Имя монитора изменить нельзя.

#### 5.2.1.5 Удаление датчика

Чтобы удалить датчик, на вкладке `Датчики` нажмите значок удаления в строке датчика и подтвердите операцию.

> **Примечание**
>
> Если удаляемый датчик используется в мониторе, его состояние в мониторе начнет отображаться как `нет данных`.

#### 5.2.1.6 Удаление монитора

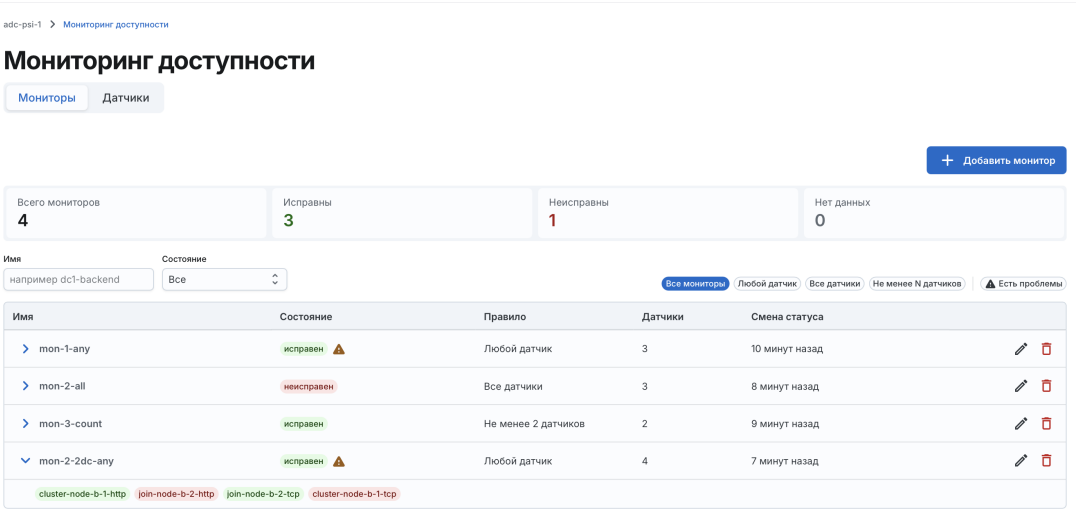
Чтобы удалить монитор, на вкладке `Мониторы` нажмите значок удаления в строке монитора и подтвердите операцию. Монитор будет удален из кластера.

> **Примечание**
>
> Монитор, используемый в конфигурации GSLB, нельзя удалить. Сначала монитор необходимо удалить из настроек бэкенда.

### 5.2.2 Мониторинг доступности

Состояние датчиков и мониторов отображается в веб-интерфейсе Angie ADC (`Система` → `Доступность`) на вкладках `Мониторы` и `Датчики`. Мониторинг можно проводить с любого узла кластера Angie ADC.

#### 5.2.2.1 Мониторы




В верхней части экрана отображается сводная статистика по мониторам:

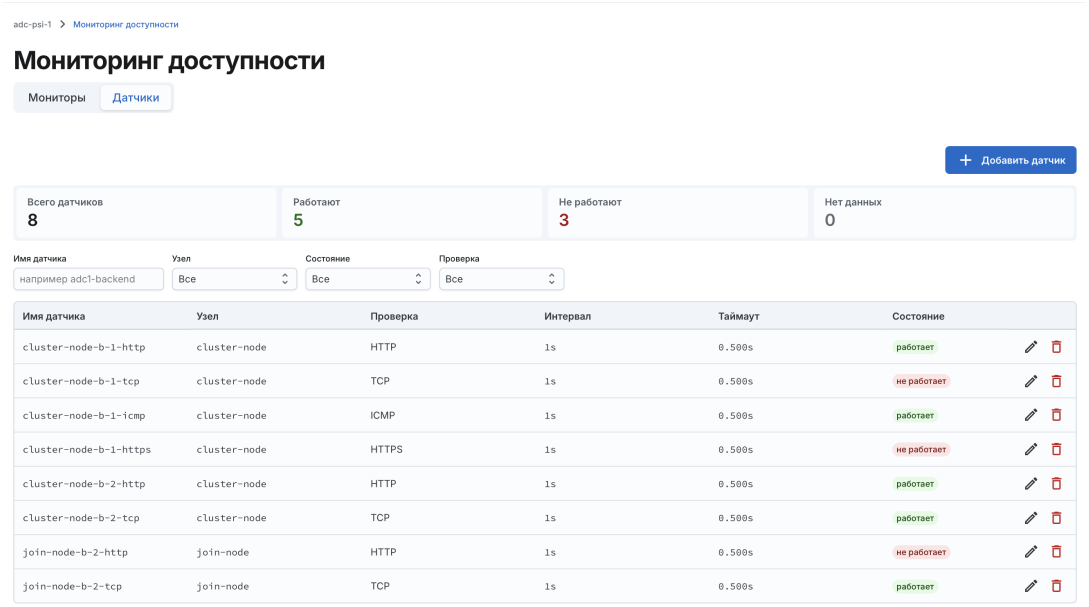
- `Всего мониторов`
- `исправны`
- `неисправны`
- `нет данных`

Ниже отображается таблица с данными по каждому монитору. Таблицу можно отфильтровать по имени монитора, состоянию и правилу агрегации.

Для каждого монитора в таблице отображаются:

| | |
|---|---|
| `Имя` | Имя монитора. |
| `Состояние` | Текущий статус монитора: `исправен`, `неисправен` или `нет данных`.<br><br>**Примечание**<br>Индикация `есть проблемы` выводится для монитора, когда правило агрегации позволяет вывести статус `исправен`, но при этом часть датчиков не работает или по ним нет данных. |
| `Правило` | Правило агрегации статусов датчиков в мониторе. |
| `Датчики` | Число датчиков в мониторе.<br><br>**Совет**<br>Чтобы увидеть входящие датчики и их статусы, разверните строку монитора. |
| `Смена статуса` | Время последнего изменения статуса монитора. |

#### 5.2.2.2 Датчики




В верхней части экрана отображается сводная статистика по датчикам:

- `Всего датчиков`
- `работают`
- `не работают`
- `нет данных`

Ниже отображается таблица с данными по каждому датчику. Таблицу можно отфильтровать по имени, узлу, состоянию и типу проверки.

Для каждого датчика в таблице отображаются:

| | |
|---|---|
| `Имя` | Имя датчика. |
| `Узел` | Узел кластера, на котором выполняется проверка. |
| `Проверка` | Тип проверки: `HTTP`, `TCP`, `ICMP` или `Angie status`. |
| `Интервал` | Периодичность выполнения проверки. |
| `Таймаут` | Время ожидания ответа. |
| `Состояние` | Текущий статус датчика: `работает`, `не работает` или `нет данных`. |

## 5.3 Диагностика и логирование

В Angie ADC доступны следующие средства диагностики:

Веб-интерфейс:

- *Базовая диагностика сети*.
- *Журнал ошибок балансировщика*.

CLI:

- *Журналы событий Angie ADC*.
- *Сбор диагностической информации* (команда `diag`).
- Расширенная диагностика (контекст `diagnostics`).

Мастер настройки:

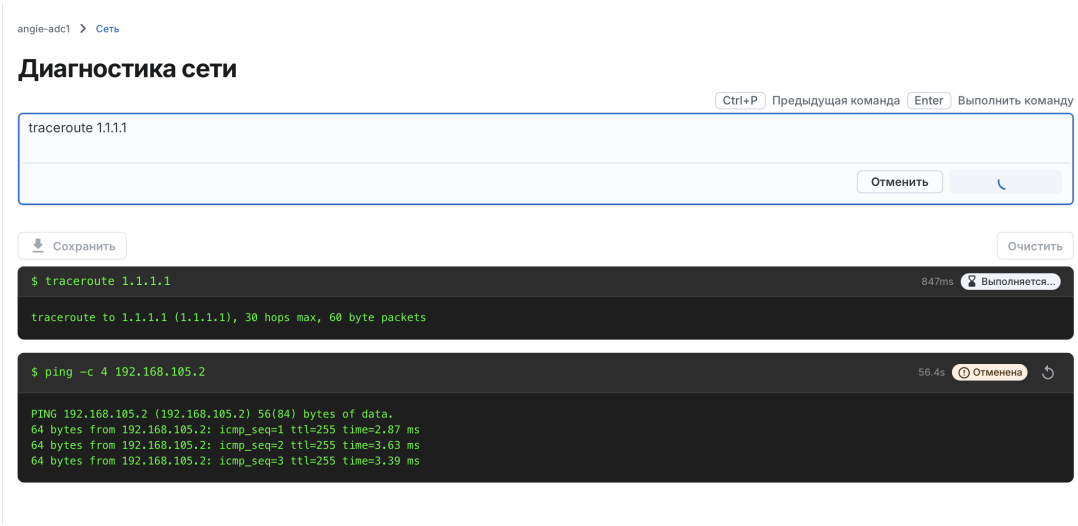
- *Аварийный сбор диагностической информации*.

Техническая поддержка для клиентов:

- https://support.angie.software
- support@angie.software

### 5.3.1 Диагностика сети

Вы можете проводить базовую диагностику сети через веб-интерфейс (`Диагностика` → `Сеть`).




Поддерживаются команды `ping`, `curl` и `traceroute`. На вкладке отображается окно для ввода команды и окно вывода, в котором в режиме реального времени отображается процесс выполнения команды. При необходимости можно выгрузить отчет в HTML-формате (кнопка `Сохранить`).

Команды `ping` и `traceroute` поддерживают все стандартные аргументы (см. документацию ping, traceroute на man7.org).

Команда `curl` имеет ограничения:

- Разрешены все поддерживаемые протоколы, кроме *file* (см. документацию curl на man7.org).
- Разрешены только следующие аргументы:
  - `--v` (`--verbose`)
  - `--s` (`--silent`)
  - `--S` (`--show-error`)
  - `--i` (`--include`)
  - `--I` (`--head`)
  - `--L` (`--location`)
  - `--k` (`--insecure`)
  - `--m` (`--max-time`)
  - `--x` (`--proxy`)
  - `--u` (`--user`)
  - `--H` (`--header`)
  - `--A` (`--user-agent`)
  - `--e` (`--referer`)
  - `--d` (`--data`)
  - `--b` (`--cookie`)
  - `--G` (`--get`)
  - `--X` (`--request`)
  - `--w` (`--write-out`)

Примеры:

```
curl https://example.com      # проверка доступности веб-сервиса по HTTP/HTTPS
ping 192.0.2.10               # проверка сетевой доступности узла и времени отклика
traceroute 192.0.2.10         # определение маршрута до удаленного узла
```

**Примечание**

Также доступна расширенная диагностика через CLI (контекст adc_cli-diagnostics, поддерживаемые команды: `arping`, `diag`, `nslookup`, `ping`, `pingc`, `ss`, `ssldump`, `tcpdump`, `traceroute`).

### 5.3.2 Диагностика через CLI

Диагностические данные о работе системы можно получить через CLI.

Для этого:

1. Подключитесь к CLI Angie ADC.
2. Перейдите в контекст `diagnostics` и выполните команду `diag`.

Команда собирает системные метрики, конфигурацию системы и системный журнал за последние 24 часа и сохраняет их в файл `diag_<timestamp>.log` в каталоге `/var/transfer`.

Подробнее о расширенной диагностике см. справочник diagnostics.

**Примечание**

Если веб-интерфейс и CLI недоступны, используйте *аварийный сбор данных*.

### 5.3.3 Аварийная диагностика

Если веб-интерфейс и CLI Angie ADC недоступны, вы можете скачать архив с диагностической информацией через мастер настройки. Аварийную выгрузку необходимо выполнять отдельно на каждом узле Angie ADC.

Предусловия:

- Есть доступ к консоли виртуальной машины Angie ADC через гипервизор.
- Есть сетевая связность с Angie ADC.
- На Angie ADC настроен и доступен по сети хотя бы один IP-адрес.

Чтобы получить архив:

1. Войдите *в мастер настройки*.
2. В основном меню выберите пункт `emergency_diag`.

   Команда соберет системные метрики, конфигурацию системы и системный журнал и сформирует архив.
3. Дождитесь создания архива. После этого в мастере отобразятся имя архива, порт SFTP и срок, в течение которого архив доступен для скачивания (15 минут). Через 15 минут архив будет удален.

   **Примечание**

   При повторном запуске команды `emergency_diag` в течение 15 минут текущий архив будет удален и будет создан новый архив.
4. Подключитесь по SFTP под учетной записью `setup`:

   ```
   $ sftp -P 2223 setup@<IP-адрес>
   ```

   **Примечание**

   Если был настроен интерфейс управления, используйте его IP-адрес.
5. Проверьте наличие архива (команда `ls`) и скачайте его:

   ```
   sftp> get diag_YYYYMMDD_HHMMSS.tar.gz
   ```

См. также *Диагностика через CLI*.

### 5.3.4 Журналы событий и экспорт

Angie ADC записывает все события системы и действия пользователей в веб-интерфейсе в журналы событий. Вы можете просмотреть журналы событий через CLI и настроить отправку событий на удаленный syslog-сервер для дальнейшего анализа и аудита.

#### 5.3.4.1 Просмотр журналов событий

Журналы событий Angie ADC можно посмотреть через CLI.

Синтаксис: `logs <entity> [{follow|transfer}]`.

Поддерживается фильтрация.

Значения для `<entity>`:

| | |
|---|---|
| `aaa-manager` | События модуля, отвечающего за аутентификацию и авторизацию внешних пользователей Angie ADC |
| `adc-gslb` | События модуля GSLB (глобальная балансировка) |
| `adc-server` | События модуля, обеспечивающего выполнение всего предоставляемого функционала Angie ADC (входная точка) |
| `adc-system` | События модуля, отвечающего за выполнение системных команд Angie ADC |
| `adc-tracker` | События модуля, отвечающего за RHI-функционал |
| `auth-dataplane` | События модуля, отвечающего за *аутентификацию клиентов* в балансировщике нагрузки |
| `certificate-manager` | События сервиса управления сертификатами |
| `ctrl-access` | События из access_log модуля, отвечающего за кластерное взаимодействие |
| `ctrl-error` | События из error_log модуля, отвечающего за кластерное взаимодействие |
| `gslb-component` | События модуля конфигурации GSLB |
| `kern` | События ядра |
| `lb-access` | События из access_log балансировщика нагрузки |
| `lb-error` | События из error_log балансировщика нагрузки |
| `lb-ls` | Позволяет получить список журналов балансировщика нагрузки (файлы должны иметь расширение `.log`)<br>Пример:<br>`$$ logs lb-ls`<br>`access.log`<br>`error.log`<br>`healthmonitoring.log`<br>`$$` |
| `mgmt-access` | События из access_log модуля, отвечающего за управление Angie ADC |
| `mgmt-error` | События из error_log модуля, отвечающего за управление Angie ADC |
| `session-store` | События модуля, отвечающего за хранение sticky-сессий в паре высокой доступности |
| `snmp-manager` | События модуля SNMP |
| `storage` | События из storage_log модуля, отвечающего за управление Angie ADC |
| `supervisor` | События модуля, отвечающего за управление кластеризацией Angie ADC |
| `upgrade` | События обновления Angie ADC |
| `view <filename>` | Позволяет просмотреть произвольный журнал событий, например, добавленный вручную в настройках конфигурации балансировщика. Файл должен находиться в папке `/var/log/angie-lb/` и иметь расширение `.log`.<br>Пример:<br>`$$ logs view error.log` |

Дополнительные параметры:

| | |
|---|---|
| `follow` | Вывод записей в режиме реального времени. |
| `transfer` | Копирование записей в директорию `transfer`. |

#### 5.3.4.2 Экспорт логов

Angie ADC позволяет экспортировать события в работе системы на удаленные серверы по syslog-протоколу. Для работы функциональности необходимо настроить syslog-серверы, на которые будут отправляться события, и указать уровни критичности событий, определяющие, какие события будут отправляться на удаленный сервер.

Поддерживаются следующие уровни критичности событий:

- `EMERGENCY` — только события, приводящие к аварийной ситуации и полному отказу системы.
- `ALERT` — критические ошибки безопасности или выход из строя важных сервисов.
- `CRITICAL` — серьезные проблемы, которые требуют немедленного исправления.
- `ERROR` — ошибки, влияющие на работу системы, но не приводящие к ее остановке.
- `WARNING` — предупреждения о возможных проблемах.
- `NOTICE` — уведомления о событиях.
- `INFO` — информационные сообщения (сообщения аудита дополнительно имеют помету `audit`).
- `DEBUG` — события отладки.

##### Создание нового syslog-сервера

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Логирование`.

   Откроется окно `Логирование`. В этом окне отображаются настроенные syslog-серверы.

2. Чтобы добавить новый syslog-сервер, нажмите `Добавить сервер`. В открывшемся окне `Новый сервер` заполните следующие поля:

   - Имя сервера (до 255 символов).
   - Доменное имя или IP-адрес (до 255 символов).
   - Порт (по умолчанию используется 514).
   - Протокол: `TCP` или `UDP` (по умолчанию: `UDP`).
   - Уровни критичности: выберите уровни критичности событий, определяющие, какие события будут отправляться на сервер (доступные уровни см. выше).

3. Нажмите `Добавить сервер`.

   Новый сервер отобразится в списке настроенных syslog-серверов.

4. Чтобы протестировать соединение с новым сервером, нажмите на имя этого сервера в таблице. Откроется окно с параметрами выбранного сервера.

5. Нажмите `Проверить соединение`. В случае установки соединения отобразится сообщение `Подключение прошло успешно`.

##### Просмотр настроенных syslog-серверов

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Логирование`.

   Откроется окно `Логирование`. В этом окне отображаются настроенные syslog-серверы.

2. Нажмите на конкретный сервер, чтобы просмотреть его параметры.

#### Изменение параметров syslog-сервера

1. Перейдите в веб-интерфейс `Система` → `Логирование` → `<имя_сервера>`.

   Откроется окно с параметрами выбранного сервера.

2. Чтобы включить или выключить передачу событий на выбранный syslog-сервер, переведите переключатель `Включить` в соответствующее положение.

3. Чтобы изменить параметры сервера, внесите необходимые правки и нажмите `Сохранить`.

4. Чтобы протестировать соединение с сервером, нажмите `Проверить соединение`.

5. Чтобы удалить сервер, нажмите значок удаления напротив этого сервера.

### CLI

Можно выполнять следующие операции:

- Создание syslog-сервера.
- Получение списка syslog-серверов.
- Изменение параметров syslog-сервера.
- Удаление syslog-сервера.
- Проверка подключения к syslog-серверу.

Доступные команды CLI см. в описании команды syslog.

## 5.3.5 Техническая поддержка

Если у вас возникла техническая проблема, но нужное решение не удалось найти в других разделах, задайте нам вопрос на форуме сообщества или в Telegram-канале.

Техническая поддержка для клиентов:

- https://support.angie.software
- support@angie.software

**Примечание**

Документация Angie ADC доступна в машиночитаемом виде для LLM-инструментов (`llms.txt`, `llms-full.txt`, Markdown-версии страниц) — подробности в разделе ИИ-документация. Карточка Angie ADC в реестре Context7: https://context7.com/websites/angie_software_adc.

ГЛАВА 6

# Администрирование и безопасность

Здесь описывается администрирование и обеспечение безопасности Angie ADC: системные настройки (включая HTTPS и сертификаты), управление пользователями и доступами, а также резервное копирование и восстановление узлов.

В этом разделе:



## 6.1 Настройка системы

Здесь описываются общие системные настройки Angie ADC: имя устройства и другие параметры конфигурации, включение HTTPS и загрузка сертификатов, синхронизация времени по NTP.

В этом разделе:



### 6.1.1 Настройка конфигурации

В этой статье:



#### 6.1.1.1 Hostname

Имя устройства Angie ADC задается при развертывании системы. В дальнейшем имя устройства отображается в верхней части экрана веб-интерфейса.

#### Изменение имени устройства

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Конфигурация`.
2. В поле `Hostname (имя устройства)` введите новое имя устройства и нажмите `Сохранить`.

### CLI

Можно выполнять следующие операции:

- Получение имени устройства.
- Изменение имени устройства.

Доступные команды CLI см. в описании команды set (CLI).

#### Мастер настройки

Можно выполнять следующие операции:

- Просмотр имени устройства.
- Изменение имени устройства.

Подробнее см. *Мастер настройки*.

### 6.1.1.2 Сетевые интерфейсы

Вы можете просмотреть и изменить параметры *настроенных сетевых интерфейсов* в веб-интерфейсе.

#### Просмотр настроенных сетевых интерфейсов

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Конфигурация`.

   В открывшемся окне в блоке `Сетевые интерфейсы` отображается список настроенных сетевых интерфейсов.

   Для каждого интерфейса отображаются следующие параметры:

   - Имя сетевого интерфейса.
   - MAC-адрес (формат: `"XX:XX:XX:XX:XX:XX"`).
   - Тип конфигурации: `static`, `DHCP`.
   - Массив IPv4-адресов в формате CIDR.
   - Массив IPv6-адресов в формате CIDR.
   - IP-адрес шлюза по умолчанию (глобальный для всех интерфейсов), если установлен.
   - Массив IP-адресов DNS-серверов.
   - Массив доменов поиска DNS.

#### Изменение параметров сетевого интерфейса

**Предупреждение**

Не рекомендуется изменять настройки интерфейсов узлов, объединенных в пару высокой доступности, иначе пара может потерять связность и работоспособность.

1. В веб-интерфейсе Angie ADC перейдите `Система` → `Конфигурация`.

2. В блоке `Сетевые интерфейсы` нажмите на сетевой интерфейс, параметры которого вы хотите изменить.

   Откроется окно с параметрами выбранного сетевого интерфейса.

   > **Примечание**
   >
   > Если для интерфейса был указан способ получения настроек по DHCP, то в окне редактирования будет отображаться подсказка с DHCP-параметрами интерфейса.

3. Внесите необходимые изменения и нажмите `Сохранить`.

#### CLI

Можно выполнять следующие операции:

- Просмотр списка сетевых интерфейсов.
- Изменение параметров сетевых интерфейсов.

Доступные команды CLI см. в описании команды set (CLI).

#### Мастер настройки

Можно выполнять следующие операции:

- Просмотр и настройка сетевых интерфейсов, бондов и VLAN-интерфейсов.

Подробнее см. *Мастер настройки*.

### 6.1.2 Настройка HTTPS

Angie ADC поддерживает загрузку сертификата управления (management) для защищенного подключения к веб-интерфейсу Angie ADC (HTTPS). По умолчанию используются порты 8080/8443, их можно *поменять*. Поддерживается загрузка и использование только одного сертификата управления (management). При загрузке нового сертификата предыдущий автоматически удаляется.

> **Примечание**
>
> Для работы с сертификатами через веб-интерфейс необходимо предварительно создать кластер Angie ADC. Подробнее см. в статье *Создание и масштабирование кластера управления*.

#### 6.1.2.1 Загрузка сертификата управления и включение защищенного соединения

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Конфигурация`.
2. В блоке `Управление HTTPS` нажмите `Загрузить сертификат и включить HTTPS`.
3. В открывшемся окне введите данные сертификата и ключа в формате PEM вручную или загрузите их из файлов по кнопке `Загрузить из файла`.
4. Нажмите `Загрузить`.

   Сертификат будет загружен. В блоке `Управление HTTPS` будет отображаться статус HTTPS (`выключен`) и данные о сертификате: дата загрузки и срок действия.
5. Нажмите `Включить HTTPS`, чтобы включить использование защищенного соединения. После этого веб-интерфейс автоматически откроется по HTTPS.

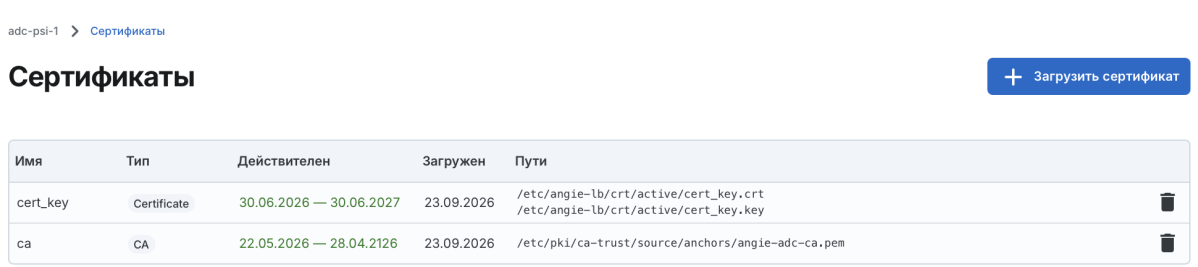
Сертификат управления и ключ можно также загружать через CLI.

### 6.1.3 Управление сертификатами

Angie ADC поддерживает загрузку сертификатов через веб-интерфейс (`Система` → `Сертификаты`). Можно загружать:

- сертификаты для обработки TLS-соединений балансируемого трафика; в систему можно загрузить несколько сертификатов такого типа, поддерживается загрузка бандла сертификатов.
- доверенные корневые и промежуточные сертификаты (CA): используются при проверке сертификатов удаленных сервисов при исходящих TLS-подключениях, например при подключении к корпоративному LDAP по LDAPS; каждый доверенный сертификат загружается отдельно, загрузка бандла из нескольких доверенных сертификатов не поддерживается.

Набор загруженных сертификатов распространяется на все узлы кластера.




#### 6.1.3.1 Загрузка сертификата данных

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Сертификаты`.
2. Нажмите `Загрузить сертификат` и заполните следующие поля:
   - `Имя сертификата` — произвольное имя сертификата, которое будет отображаться в системе.
   - `Тип` — `Certificate` для сертификата данных.
   - `Сертификат` — содержимое файла сертификата.
   - `Приватный ключ` — содержимое файла приватного ключа.

   Содержимое сертификата и ключа можно ввести вручную или загрузить из файла по кнопке `Загрузить из файла` под соответствующим полем.
3. Нажмите `Загрузить`.

Сертификат будет загружен. На странице `Сертификаты` будут отображаться данные о сертификате: имя, тип, срок действия, дата загрузки и пути к сертификату и ключу.

> **Примечание**
>
> После загрузки сертификата необходимо вручную настроить его использование в *конфигурации балансировщика нагрузки*.

Например:

```
server {
    listen       80;
    listen       [::]:80;
    listen       443 ssl;
    listen       [::]:443 ssl;
    server_name  localhost;
```

```
    ssl_protocols       TLSv1.2 TLSv1.3;
    ssl_ciphers         ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-
↪SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384;
    ssl_certificate     /etc/angie-lb/crt/active/example.crt;
    ssl_certificate_key /etc/angie-lb/crt/active/example.key;
    ssl_session_cache   shared:SSL:10m;
    ssl_session_timeout 10m;

    if ($scheme = http) {
        return 301 https://$host:443$request_uri;
    }

    ## location
}
```

#### 6.1.3.2 Загрузка доверенного сертификата

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Сертификаты`.
2. Нажмите `Загрузить сертификат` и заполните следующие поля:
   - `Имя сертификата` — произвольное имя сертификата, которое будет отображаться в системе.
   - `Тип` — `CA` для доверенного сертификата.
   - `Сертификат` — содержимое файла сертификата (PEM).

   Содержимое сертификата можно ввести вручную или загрузить из файла по кнопке `Загрузить из файла` под полем.
3. Нажмите `Загрузить`.

Сертификат будет загружен. На странице `Сертификаты` будут отображаться данные о сертификате: имя, тип, срок действия, дата загрузки и путь к сертификату.

#### 6.1.3.3 Удаление сертификатов

Чтобы удалить сертификат:

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Сертификаты`. Откроется список сертификатов.
2. Нажмите значок удаления рядом с сертификатом, который вы хотите удалить.

Сертификат будет удален из списка.

#### 6.1.3.4 Управление через CLI

Сертификаты можно также загружать и просматривать через CLI:

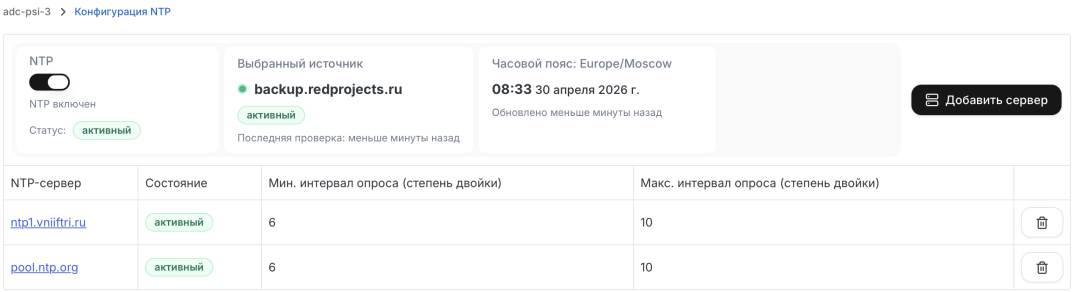
- Сертификаты данных
- Доверенные сертификаты

### 6.1.4 Настройка NTP

NTP (Network Time Protocol) позволяет синхронизировать системное время на всех компонентах системы Angie ADC. Вы можете добавить один или несколько NTP-серверов, которые будут использоваться как источники времени.

**Важно**

Синхронизация времени на узлах с единым внешним источником критически важна для работы *кластера*.




В интерфейсе Angie ADC отображается статус службы NTP и состояние серверов (источников времени).

Статусы службы NTP:

- `активный` — служба NTP включена.
- `выключен` — служба NTP выключена.
- `есть проблемы` — возникли проблемы с синхронизацией времени.
- `не отвечает` — служба недоступна.
- `данные устарели` — не удалось получить свежие данные.
- `применение настроек...` — идет применение изменений.

У NTP-серверов могут быть следующие статусы:

- `активный` — сервер доступен.
- `есть проблемы` — возникли проблемы с синхронизацией времени.
- `не отвечает` — сервер недоступен.
- `неизвестно` — статус сервера не определен.
- `применение настроек...` — идет применение изменений.
- `NTS не подтвержден` — сервер не поддерживает NTS, синхронизация времени невозможна.

#### 6.1.4.1 Настройка конфигурации NTP

1. В веб-интерфейсе Angie ADC перейдите `Система` → `NTP`.

   Откроется окно `Конфигурация NTP`.

2. Включите переключатель `NTP`, чтобы запустить службу синхронизации NTP.

3. Чтобы добавить NTP-сервер, нажмите `Добавить сервер` и заполните необходимые поля:

   - Адрес NTP-сервера, например `ntp1.vniiftri.ru`.
   - Минимальный и максимальный интервалы для частоты опроса сервера. Частота опроса может меняться динамически: при нестабильном времени сервер опрашивается чаще, при стабильной синхронизации — реже.

     В качестве значения указывается степень двойки, интервал опроса NTP-сервера рассчитывается как 2 в степени n секунд.

     Например:

- `6` → $2^6 = 64$ секунды;
- `10` → $2^{10} = 1024$ секунды (~17 минут).

4. Установите флажок `NTS`, если хотите использовать аутентифицированную синхронизацию времени. Если серевер не поддерживает NTS, синхронизация времени будет невозможна, у сервера отобразится статус `NTS не подтвержден`.
5. Нажмите `Добавить`.
6. Чтобы изменить параметры сервера, нажмите на имя сервера в таблице. Откроется окно с параметрами сервера. Внесите необходимые правки и сохраните изменения.
7. Чтобы удалить сервер, нажмите значок удаления в таблице напротив этого сервера.

#### 6.1.4.2 CLI

Настройка выполняется в контексте `system`. Можно выполнять следующие операции:

- Просмотр конфигурации и статуса NTP (`ntp`).
- Включение NTP (`ntp enable`).
- Выключение NTP (`ntp disable`).
- Добавление или изменение NTP-сервера (`ntp server <host> [minpoll <value>] [maxpoll <value>] [nts]`). Параметр `nts` включает аутентифицированную синхронизацию времени (NTS).
- Удаление NTP-сервера (`ntp remove server <host>`).

Подробнее см. ntp.

## 6.2 Управление пользователями и доступами

В этом разделе описывается управление пользователями Angie ADC: добавление локальных пользователей и подключение внешних через LDAP, RADIUS или TACACS, назначение ролей, требования к паролям и настройка безопасности собственной учетной записи.

В этом разделе:

- *Пользователи*
- *Аутентификация*
- *Авторизация*
- *Парольная политика*
- *Настройка учетной записи*
- *Аудит действий пользователей*

### 6.2.1 Пользователи

#### 6.2.1.1 Типы пользователей и роли

В Angie ADC могут работать *локальные пользователи* системы и *внешние пользователи*, аутентифицированные через внешний сервер аутентификации.

Пользователям можно назначить одну из следующих **ролей**:

- `Администратор`: обладает полными правами на все действия в системе;
- `Супервизор`: имеет ограниченные права, может только просматривать информацию в системе, например, проводить аудит;

- `Оператор`: имеет ограниченные права, может редактировать файлы конфигурации балансировщика, к которым ему предоставлен доступ, и просматривать статистику зон, к которым ему предоставлен доступ. Доступы предоставляет администратор. Остальные настройки Angie ADC оператору не доступны и не отображаются в веб-интерфейсе.

**Примечание**

При двух неудачных попытках входа в течение одного часа любой пользователь будет временно заблокирован.

#### 6.2.1.2 Локальные пользователи

По умолчанию предустановлен локальный пользователь `admin` с правами администратора.

**Примечание**

Пользователь `admin` нужен для создания пары высокой доступности.

Локальные пользователи системы отображаются в веб-интерфейсе в разделе `Система` → `Пользователи`. Для каждого пользователя представлена следующая информация:

| | |
|---|---|
| `Логин` | Логин пользователя в системе |
| `Имя` | Собственное имя пользователя |
| `Фамилия` | Фамилия пользователя |
| `UUID` | Уникальный идентификатор пользователя |
| `Статус` | Статус пользователя в системе:<br>• `Активный` — пользователь имеет доступ к системе.<br>• `Неактивный` — у пользователя временно отключен доступ к системе, например, на время отпуска.<br>• `Удален` — пользователь удален, но может отображаться в списке, если включен переключатель `Показывать удаленных пользователей`. |
| `Роль` | Роль пользователя в системе:<br>• `Администратор` — обладает полными правами на все действия в системе.<br>• `Супервизор` — имеет ограниченные права, может только просматривать информацию в системе, например, проводить аудит;<br>• `Оператор` — имеет ограниченные права, может редактировать файлы конфигурации балансировщика, к которым ему предоставлен доступ, и просматривать статистику зон, к которым ему предоставлен доступ. Доступы предоставляет администратор. Остальные настройки Angie ADC оператору не доступны и не отображаются в веб-интерфейсе. |
| `Комментарий` | Примечание |

#### 6.2.1.3 Внешние пользователи

Angie ADC поддерживает аутентификацию и авторизацию пользователей через внешний сервер. При использовании внешней аутентификации и авторизации данные о пользователях хранятся на внешнем сервере. У системы нет доступа к данным пользователя.

На данный момент поддерживаются провайдеры RADIUS, LDAP и TACACS. Подробнее см. *Аутентификация* и *Авторизация*.

#### 6.2.1.4 Добавление локального пользователя

Чтобы добавить нового локального пользователя:

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Пользователи`.

   Откроется список пользователей.

2. В открывшемся окне нажмите кнопку `+ Добавить пользователя`.

3. Укажите логин, пароль, имя и фамилию нового пользователя, а также его роль и нажмите `Добавить`.

   Требования:

   - Логин — допускаются латинские буквы, цифры, а также специальные символы: `_ - . @ ! # $ % ^ & * ( ) = +`.
   - Пароль — допускаются латинские буквы, цифры, а также специальные символы: `_ - . @ ! # $ % ^ & * ( ) = +`. Пароль должен соответствовать текущей *парольной политике*. Пароль можно ввести вручную или сгенерировать. Пароль не может совпадать с логином без учета регистра.

     > **Примечание**
     >
     > Кнопка копирования пароля будет работать, если используется защищенное подключение к веб-интерфейсу (*HTTPS*).

   - Имя — допускаются только буквы латиницы и кириллицы, а также дефис и пробел.
   - Фамилия — допускаются только буквы латиницы и кириллицы, а также дефис и пробел.
   - Роль:
     - `Администратор` — обладает полными правами на все действия в системе;
     - `Супервизор` — имеет ограниченные права, может только просматривать информацию в системе, например, проводить аудит;
     - `Оператор` — имеет ограниченные права, может редактировать файлы конфигурации балансировщика, к которым ему предоставлен доступ, и просматривать статистику зон, к которым ему предоставлен доступ. Доступы предоставляет администратор. Остальные настройки Angie ADC оператору не доступны и не отображаются в веб-интерфейсе.

   Новый пользователь отобразится в таблице.

4. Передайте пользователю логин и пароль безопасным способом.

#### 6.2.1.5 Изменение учетных данных и статуса локального пользователя

Чтобы изменить данные локального пользователя:

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Пользователи`.

   Откроется список пользователей.

2. Нажмите значок редактирования рядом с пользователем, данные или статус которого вы хотите изменить.

3. В открывшемся окне укажите новые данные и нажмите `Сохранить`.

Информация о пользователе будет обновлена.

#### 6.2.1.6 Сброс пароля локального пользователя

Администратор может вручную сбросить пароль пользователя и задать для него новый пароль. После сброса пароля пользователю необходимо заново войти в систему.

1. Перейдите в раздел `Система` → `Пользователи`.
2. Нажмите значок "ключ" (`Сбросить пароль`) напротив нужного пользователя.
3. В открывшемся окне введите новый пароль для пользователя и повторите его. Можно ввести пароль вручную или сгенерировать его с учетом текущей парольной политики.

> **Примечание**
>
> Кнопка копирования пароля будет работать, если используется защищенное подключение к веб-интерфейсу (*HTTPS*).

4. Нажмите `Сбросить пароль`.

   Старый пароль будет сброшен, новый пароль будет сохранен.
5. Передайте пользователю новый пароль безопасным способом.

#### 6.2.1.7 Сброс настроек двухфакторной аутентификации локального пользователя

Администратор может вручную сбросить настройки двухфакторной аутентификации для локального пользователя, например, если пользователь больше не имеет доступа к подключенному приложению для аутентификации. После сброса настроек двухфакторной аутентификации пользователю необходимо заново войти в систему. При входе в систему пользователь должен настроить двухфакторную аутентификацию заново, если этого требует *парольная политика*.

1. Перейдите в раздел `Система` → `Пользователи`.
2. Нажмите значок "экран с крестиком" (`Сбросить TOTP`) напротив нужного пользователя.

   Настройки двухфакторной аутентификации будут сброшены.

#### 6.2.1.8 Удаление локального пользователя

Чтобы удалить локального пользователя:

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Пользователи`. Откроется список пользователей.
2. Нажмите значок удаления рядом с пользователем, запись о котором вы хотите удалить.

Пользователь будет удален из списка, но будет отображаться в списке пользователей, если включен переключатель `Показывать удаленных пользователей`. В дальнейшем вы можете просмотреть список удаленных пользователей при аудите системы.

> **Примечание**
>
> Если вы захотите вернуть удаленного пользователя в систему, необходимо изменить его статус на `Активный`.

### 6.2.2 Аутентификация

#### 6.2.2.1 Типы аутентификации

Angie ADC поддерживает два типа аутентификации пользователей:

- *Локальная аутентификация.*

При использовании локальной аутентификации данные о пользователях хранятся в системе. Вы можете просмотреть их в веб-интерфейсе (`Система` → `Пользователи`). Для входа в веб-интерфейс пользователю необходимо выбрать локальный тип аутентификации и ввести свои логин и пароль.

- *Внешняя аутентификация.*

  При использовании внешней аутентификации данные о пользователях хранятся на внешнем сервере аутентификации. У системы нет доступа к данным пользователя. Для включения внешней аутентификации необходимо добавить в веб-интерфейс предварительно настроенный сервер аутентификации. Поддерживаются RADIUS, TACACS и LDAP. После этого при входе в веб-интерфейс будет доступен выбор аутентификации через заданный сервер.

> **Предупреждение**
>
> Необходимо также включить и настроить *авторизацию* для внешних пользователей. Если авторизация не настроена, пользователи не смогут получить доступ к системе.

#### 6.2.2.2 Включение внешней аутентификации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. Включите переключатель `Включить внешнюю аутентификацию` и выберите провайдера аутентификации.

> **Примечание**
>
> Одновременно поддерживается использование только одного провайдера внешней аутентификации. При добавлении нового провайдера аутентификации настройки предыдущего провайдера сбрасываются.

3. Заполните нужные поля.

   Для RADIUS-аутентификации укажите следующие параметры сервера:

   - `Сервер`: IP-адрес или доменное имя RADIUS-сервера.
   - `Порт`: порт подключения к RADIUS-серверу (по умолчанию `1812`).
   - `Протокол`: протокол передачи данных: `UDP` или `TCP` (по умолчанию: `UDP`).
   - `NAS-идентификатор`: идентификатор NAS (Network Access Server), передаваемый серверу RADIUS.
   - `Метод аутентификации`: поддерживается только `PAP`.
   - `Секрет`: общий секрет для шифрования паролей.

   Для TACACS-аутентификации укажите следующие параметры сервера:

   - `Адрес`: IP-адрес или доменное имя TACACS-сервера.
   - `Порт`: порт подключения к TACACS-серверу (по умолчанию `49`).
   - `Секрет`: общий секрет для шифрования паролей.

   Для LDAP-аутентификации укажите параметры LDAP-сервера, сервисной учетной записи и параметры поиска пользователей:

   - `Сервер`: IP-адрес или доменное имя LDAP-сервера.

- `Порт`: порт подключения к LDAP-серверу (по умолчанию `389`).
- `Режим защиты соединения` (по умолчанию `Без TLS`, также доступны варианты `LDAPS` и `StartTLS`).
- `DN сервисной учетной записи`: логин сервисной учетной записи, от имени которой система подключается к LDAP.
- `Пароль сервисной учетной записи`: пароль сервисной учетной записи для подключения к LDAP-серверу.
- `Базовый DN пользователя`: задает ветку на LDAP-сервере, в которой система будет искать пользователя, например `ou=users,dc=example,dc=com`.
- `Идентификатор логина`: поле в LDAP, которое содержит логин, например `uid`, `sAMAccountName` или `userPrincipalName`.
- `Фильтр поиска`: LDAP-фильтр для поиска пользователя, например `(&(objectClass=user)(sAMAccountName=%s))`.

**Примечание**

Фильтр LDAP определяет, в каком формате пользователь должен вводить логин при входе в систему:

- если используется фильтр `(&(objectClass=user)(sAMAccountName=%s))`, то логин будет без доменного имени, например `ivanov`;
- если используется фильтр `(&(objectClass=user)(userPrincipalName=%s)`, то логин будет полным: `ivanov@domain.com`.

4. Нажмите `Сохранить`.

Внешняя аутентификация с использованием настроенного провайдера будет включена. При входе в веб-интерфейс пользователю будет необходимо выбрать нужный тип аутентификации и ввести свои логин и пароль. Если пользователь с таким логином и паролем будет найден на заданном сервере, то аутентификация пройдет успешно.

#### 6.2.2.3 Изменение параметров внешней аутентификации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. Внесите необходимые изменения и нажмите `Сохранить`.

   Параметры внешней аутентификации будут обновлены.

#### 6.2.2.4 Отключение внешней аутентификации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. Переведите переключатель `Включить внешнюю аутентификацию` в положение "выключено".

Внешняя аутентификация будет отключена.

### 6.2.3 Авторизация

Angie ADC поддерживает ролевое разграничение доступа к функциям системы. Пользователям может быть назначена одна из следующих ролей:

- `Администратор`: обладает полными правами на все действия в системе;
- `Супервизор`: имеет ограниченные права, может только просматривать информацию в системе, например, проводить аудит;
- `Оператор`: имеет ограниченные права, может редактировать файлы конфигурации балансировщика, к которым ему предоставлен доступ, и просматривать статистику зон, к которым ему предоставлен доступ. Доступы предоставляет администратор. Остальные настройки Angie ADC оператору не доступны и не отображаются в веб-интерфейсе.

Поддерживается два типа авторизации пользователей:

- *Локальная авторизация.*

  Роли для локальных пользователей назначаются при их создании и хранятся локально в системе. По умолчанию в Angie ADC предустановлен локальный системный пользователь `admin` с правами администратора.

- *Внешняя авторизация.*

  Роли для внешних пользователей определяются через внешнего провайдера. Поддерживаются провайдеры *RADIUS*, *TACACS+* и *LDAP*.

  **Примечание**

  RADIUS-авторизация доступна только при аутентификации через RADIUS.

  Система не хранит данные о ролях внешних пользователей. Принятые решения по внешней авторизации записываются в журналы событий системы. Записи можно экспортировать на внешний syslog-сервер для дальнейшего аудита.

  **Предупреждение**

  Необходимо также включить и настроить *аутентификацию* для внешних пользователей. Если аутентификация не настроена, пользователи не смогут войти в систему.

  Совместимость внешних провайдеров аутентификации и авторизации:

| Провайдер аутентификации | Совместимые провайдеры авторизации |
|---|---|
| RADIUS | RADIUS, LDAP, TACACS+ |
| LDAP | LDAP, TACACS+ |
| TACACS | LDAP, TACACS+ |

  **Примечание**

  Одновременно поддерживается использование только одного провайдера. При добавлении нового провайдера настройки предыдущего провайдера сбрасываются.

В этой статье:

- *LDAP-авторизация*
- *TACACS-авторизация*

- *Выключение внешней авторизации*

#### 6.2.3.1 LDAP-авторизация

Роли внешних пользователей определяются на основе принадлежности к LDAP-группам.

Чтобы настроить LDAP-авторизацию для внешних пользователей, необходимо:

- включить внешнюю аутентификацию;
- включить внешнюю авторизацию и подключить LDAP-сервер;
- задать настройки сервисной учетной записи для подключения Angie ADC к LDAP-серверу;
- задать базовый DN и фильтры для поиска пользователей и групп на LDAP-сервере;
- настроить соответствие LDAP-групп и ролей пользователей в Angie ADC.

**Примечание**

На стороне LDAP-сервера также должны быть настроены пользователи, группы и сервисная учетная запись для Angie ADC.

##### Настройка LDAP-авторизации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. В блоке `Внешняя авторизация` переведите переключатель `Включить внешнюю авторизацию` в положение "включено" и выберите провайдера авторизации LDAP.

3. Задайте параметры подключения к серверу:

   - `Сервер`: IP-адрес или доменное имя LDAP-сервера.
   - `Порт`: порт подключения к LDAP-серверу (по умолчанию `389`).

4. Укажите параметры защиты соединения: `Без TLS` (по умолчанию) или `TLS (LDAPS)`/`startTLS`, если необходимо использовать безопасное соединение при подключении к серверу.

   **Примечание**

   При выборе режима защиты соединения `LDAPS` или `StartTLS` становится доступен флажок `Проверять TLS-сертификат`. По умолчанию флажок установлен.

5. Укажите DN учетной записи, с которой будет выполняться подключение к LDAP-серверу (`DN сервисной учетной записи (bind)`).

   Можно использовать учетную запись администратора LDAP-сервера, например `cn=admin,dc=example,dc=com`.

6. Укажите пароль сервисной учетной записи (`Пароль пользователя bind`).

7. В блоке `Поиск пользователя` укажите следующие параметры:

   - `Базовый DN пользователей`, например `ou=users,dc=example,dc=com`;
   - `Фильтр поиска пользователей`, например `(uid=%s)`.

8. В блоке `Поиск групп` укажите следующие параметры:

   - `Базовый DN групп`, например `ou=groups,dc=example,dc=com`;

- `Фильтр поиска групп`, например `(member=%s)`.

9. В блоке `Маппинг ролей LDAP` нажмите `Добавить правило` и задайте соответствие LDAP-групп пользователей ролям в Angie ADC. Например, если на LDAP-сервере для администраторов создана группа `dn: cn=<group_name>,ou=groups,dc=example,dc=com`, то укажите значение `cn=<group_name>,ou=groups,dc=example,dc=com` в поле `DN группы` и выберите роль `Администратор (admin)`.
10. Нажмите `Сохранить`.

Внешняя авторизация будет настроена и включена. При входе внешнего пользователя в систему на LDAP-сервере будет выполняться поиск групп, в которых он состоит. Если найденные группы совпадают с группами, указанными в маппинге, то пользователю будет назначена соответствующая роль. Если пользователь не найден ни в одной из групп или LDAP-группы недоступны или некорректны, доступ к системе будет запрещен.

#### 6.2.3.2 TACACS-авторизация

Роли для внешних пользователей определяются на основе соответствия атрибутам этих пользователей в TACACS+.

Чтобы настроить TACACS-авторизацию для внешних пользователей, необходимо:

- включить внешнюю аутентификацию;
- включить внешнюю авторизацию и подключить TACACS-сервер;
- настроить соответствие атрибутов TACACS+ ролям пользователей в Angie ADC.

На стороне TACACS-сервера также должны быть настроены пользователи и их атрибуты.

Пример настройки:

```
group = admin {
    service = shell {
        set priv-lvl = 15
    }
}

user = admin {
    password = clear admin
    member = admin
}
```

##### Настройка TACACS-авторизации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. В блоке `Внешняя авторизация` переведите переключатель `Включить внешнюю авторизацию` в положение "включено" и выберите провайдера авторизации TACACS.
3. Задайте параметры подключения к серверу:
   - `Адрес`: IP-адрес или доменное имя TACACS-сервера.
   - `Порт`: порт подключения к TACACS-серверу (по умолчанию `49`).
   - `Секрет`: общий секрет для шифрования паролей.
4. Укажите в порядке приоритета атрибуты ролей пользователя в TACACS+, например `priv-lvl`, затем `role`. По этим атрибутам Angie ADC будет определять роль для пользователя.

5. Укажите, какой сервис и протокол будут использоваться в запросе на авторизацию (AUTHOR). Параметры зависят от выбранного сервера. Например, для TACACS+ рекомендуется указать `service=shell` и `protocol=ip`.
6. Задайте соответствие значений атрибутов TACACS+ ролям в Angie ADC, например `priv-lvl=15` → `Администратор` или `role=adc-admin` → `Администратор`.
7. Нажмите `Сохранить`.

Внешняя авторизация будет настроена и включена. При входе внешнего пользователя в систему на TACACS-сервер будет отправлен запрос на авторизацию. Если атрибуты и значения в ответе TACACS совпадают с указанными в маппинге, то пользователю будет назначена соответствующая роль в Angie ADC. Если полученные данные не совпадают с указанными в маппинге или TACACS-сервер недоступен, доступ к системе будет запрещен.

#### 6.2.3.3 RADIUS-авторизация

Роли для внешних пользователей определяются на основе соответствия атрибутам этих пользователей в RADIUS.

Чтобы настроить RADIUS-авторизацию для внешних пользователей, необходимо:

- включить внешнюю аутентификацию через RADIUS;
- включить внешнюю авторизацию через RADIUS;
- настроить соответствие атрибутов RADIUS ролям пользователей в Angie ADC.

Допускается использование следующих атрибутов ролей:

- `Filter-Id` — роль пользователя.
- `Class` — группа, к которой относится пользователь.
- `Tunnel-Private-Group-ID` — VLAN ID (используется для VLAN-авторизации).
- `VSA` — Vendor-Specific в формате `vsa:<vendorID>:<vendorType>` (например `vsa:9:1`).

На стороне RADIUS-сервера также должны быть настроены пользователи и их атрибуты.

Пример:

```
user1 Cleartext-Password := "P@ssword"
    Filter-Id := "adc-admin",
    Service-Type := Administrative-User,
    Cisco-AVPair := "shell:priv-lvl=15",
    Reply-Message := "Welcome, Admin",
    Class := "AdminGroup",
    Tunnel-Type := VLAN,
    Tunnel-Medium-Type := IEEE-802,
    Tunnel-Private-Group-ID := "100"

user2 Cleartext-Password := "P@ssword"
    Filter-Id := "adc-readonly",
    Service-Type := NAS-Prompt-User,
    Cisco-AVPair := "shell:priv-lvl=7",
    Reply-Message := "Welcome, Supervisor",
    Class := "SupervisorGroup",
    Tunnel-Type := VLAN,
    Tunnel-Medium-Type := IEEE-802,
    Tunnel-Private-Group-ID := "200"
```

#### Настройка RADIUS-авторизации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. В блоке `Внешняя авторизация` переведите переключатель `Включить внешнюю авторизацию` в положение "включено" и выберите провайдера авторизации RADIUS.

3. Укажите атрибут роли пользователя в RADIUS, например `Filter-Id`, `vsa:9:1` или другой используемый атрибут. По этому атрибуту Angie ADC будет определять роль для пользователя.

4. Задайте соответствие значений атрибутов RADIUS ролям в Angie ADC, например для Filter-Id укажите `adc-admin` → `Администратор`, а при использовании `vsa:9:1` укажите `shell:priv-lvl=15` → `Администратор`.

5. Нажмите `Сохранить`.

Внешняя авторизация будет настроена и включена. При входе внешнего пользователя в систему через RADIUS будут получены атрибуты его роли. Если атрибуты и значения совпадают с указанными в маппинге, то пользователю будет назначена соответствующая роль в Angie ADC. Если данные не совпадают с указанными в маппинге или RADIUS-сервер недоступен, доступ к системе будет запрещен.

#### 6.2.3.4 Выключение внешней авторизации

1. Откройте веб-интерфейс Angie ADC. В панели навигации перейдите в раздел `Система` → `Аутентификация и авторизация`.

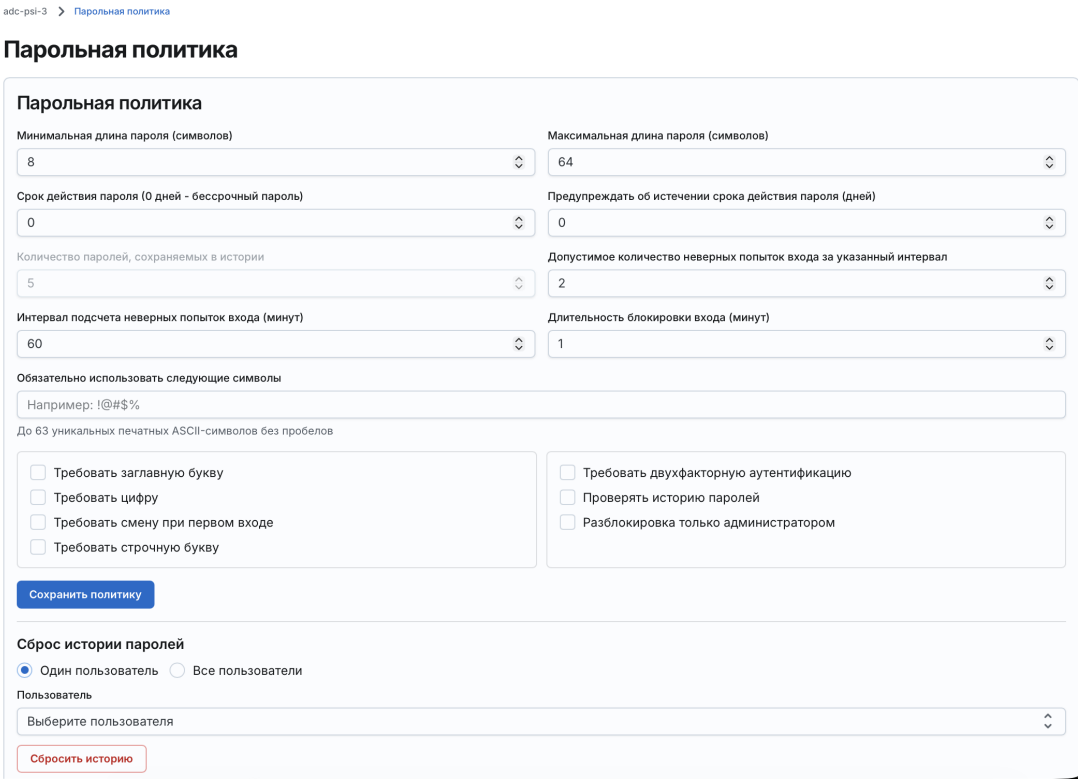
   Откроется страница с настройками аутентификации и авторизации для внешних пользователей.

2. В блоке `Внешняя авторизация` переведите переключатель `Включить внешнюю авторизацию` в положение "выключено".

Внешняя авторизация будет отключена. При входе в систему для внешних пользователей будет отображаться сообщение, что доступ запрещен.

### 6.2.4 Парольная политика

Angie ADC поддерживает парольную политику для локальных пользователей Angie ADC. Парольная политика начинает действовать сразу при запуске Angie ADC со значениями по умолчанию. Требования к паролям пользователей можно донастроить в соответствии с политикой безопасности в вашей организации в разделе `Система` → `Парольная политика`.




По умолчанию действуют следующие требования: пароль не может совпадать с логином без учета регистра, заданы минимальная длина пароля (8 символов) и максимальная длина пароля (64 символа), а также допустимое количество попыток входа и параметры блокировки пользователя:

- интервал, за который считаются неверные попытки входа пользователя (60 минут);
- допустимое количество попыток входа пользователя (2 попытки);
- длительность блокировки пользователя (1 минута).

**Примечание**

При таких параметрах пользователь, который в течение 60 минут три раза ввел неправильный пароль, будет заблокирован на одну минуту. При этом после разблокировки пользователя счетчик неправильных попыток **не сбрасывается**, пока не закончится указанный интервал (60 минут). При следующем вводе неправильного пароля в течение этих 60 минут пользователь будет заблокирован сразу.

**Важно**

Парольная политика распространяется только на *локальных пользователей Angie ADC* текущего тенанта на всех узлах *кластера*.

#### 6.2.4.1 Настройка парольной политики

1. Перейдите в раздел `Система` → `Парольная политика`.
2. Настройте **требования к составу** пароля:
   - Укажите значения в полях `Минимальная длина пароля (символов)` (по умолчанию 8 символов) и `Максимальная длина пароля` (по умолчанию 64 символа).

- Если необходимо задать более высокий уровень безопасности пароля, установите флажки `Требовать заглавную букву` (A-Z), `Требовать строчную букву` (a-z) и `Требовать цифру` (0-9).
- Если необходимо, укажите дополнительные требования в поле `Обязательно использовать следующие символы` (можно указать до 63 уникальных печатных ASCII-символов без пробелов).

3. Установите флажок `Требовать смену при первом входе`, чтобы система запрашивала смену предустановленного пароля при первом входе пользователя в веб-интерфейс.
4. Настройте **срок действия пароля** и напоминание о необходимости сменить пароль:
   - Укажите значение в поле `Срок действия пароля`.

     Значение `0` — срок действия пароля не ограничен (бессрочно).

     При указании срока действия пароля система будет напоминать о скором окончании срока за 7 дней до его истечения.
   - Укажите значение в поле `Предупреждать об истечении срока действия пароля`.

     Значение `0` — предупреждение отключено; любое другое значение — количество дней, прошедшее с последней смены пароля, после которого система будет напоминать о необходимости сменить пароль.
5. Настройте **требования к истории паролей** и **защиту от повторения**:
   - Установите флажок `Проверять историю паролей`, чтобы включить хранение истории паролей.
   - Укажите значение в поле `Количество паролей, сохраняемых в истории` (по умолчанию 5).

     Это значение будет определять, сколько паролей подряд не может повторяться. Значение `0` — количество хранимых паролей не ограничено.
6. Настройте допустимое **количество попыток входа** и **параметры блокировки** пользователя:
   - В поле `Интервал подсчета неверных попыток входа` укажите, за какое время система будет подсчитывать попытки входа пользователя (по умолчанию 60 минут).
   - В поле `Допустимое количество неверных попыток входа за указанный интервал` укажите разрешенное количество попыток входа за это время (по умолчанию 2).

     > **Примечание**
     >
     > При сбросе пароля администратором счетчик неправильных попыток ввода пароля тоже сбрасывается.

   - В поле `Длительность блокировки входа` укажите, на какое время блокировать пользователя, превысившего количество разрешенных попыток входа (по умолчанию 1 минута).
   - Установите флажок `Разблокировка только администратором`, если хотите, чтобы заблокированных пользователей можно было разблокировать только вручную. В таком случае поле `Длительность блокировки входа` не будет учитываться.
7. Если необходимо, установите флажок `Требовать двухфакторную аутентификацию`.

   При входе в веб-интерфейс пользователям будут предложены QR-код и секрет для подключения приложения-аутентификатора и дальнейшей аутентификации в системе через одноразовый пароль из этого приложения. Например, можно использовать приложения-аутентификаторы Яндекс.Ключ, Google Authenticator или Multifactor.

> **Важно**
>
> После пяти неудачных попыток ввода одноразового кода в течение пяти минут пользователь будет заблокирован на пять минут.

8. Нажмите `Сохранить политику`.

Политика будет сохранена и применена сразу.

#### 6.2.4.2 Сброс истории паролей для пользователей

Администратор может вручную сбросить историю паролей пользователя.

1. Перейдите в раздел `Система` → `Парольная политика`.
2. В блоке `Сброс истории паролей` укажите, для кого вы хотите сбросить историю:
   - `Один пользователь` (необходимо выбрать пользователя из списка).
   - `Все пользователи` (будет сброшена история паролей для всех локальных пользователей Angie ADC).
3. Нажмите `Сбросить историю`.

#### 6.2.4.3 Сброс пароля локального пользователя

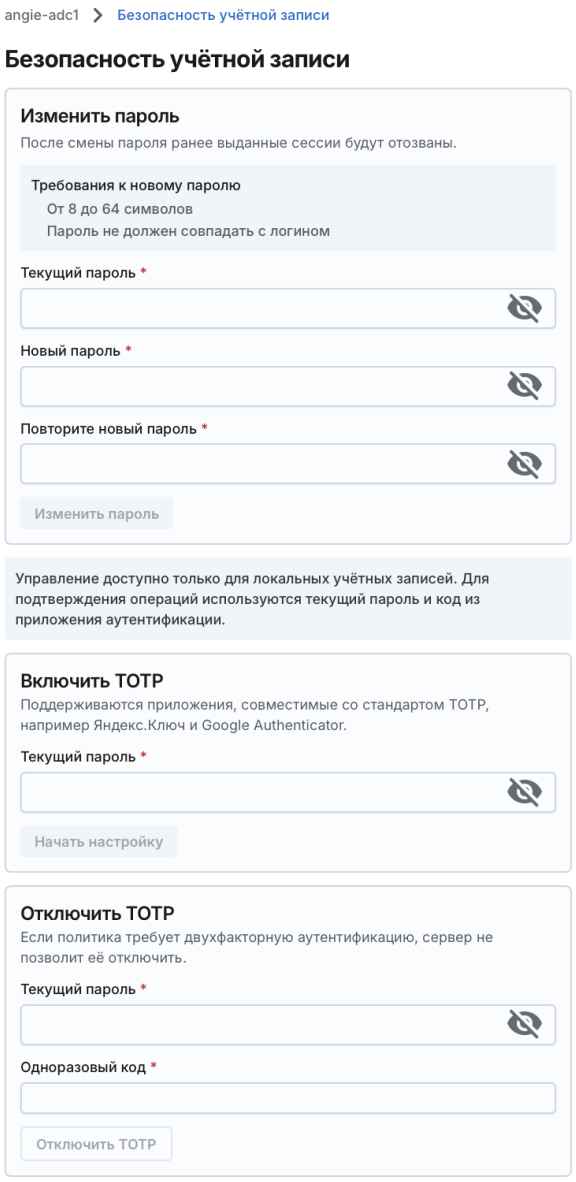
Администратор может *вручную сбросить пароль* для локального пользователя, например, если пользователь забыл пароль. После этого пользователь должен заново войти в систему с новым паролем, назначенным администратором.

#### 6.2.4.4 Сброс настроек двухфакторной аутентификации локального пользователя

Администратор может *вручную сбросить настройки двухфакторной аутентификации* для локального пользователя, например, если пользователь больше не имеет доступа к подключенному приложению для аутентификации. После этого пользователь должен заново войти в систему и настроить двухфакторную аутентификацию, если этого требует *парольная политика*.

### 6.2.5 Настройка учетной записи

Angie ADC позволяет настраивать безопасность учетной записи в Angie ADC. Настройки текущей учетной записи находятся в меню в верхнем правом углу веб-интерфейса (`Безопасность учетной записи`).




**Важно**

Настройка учетной записи доступна только локальным пользователям Angie ADC.

Поддерживается:

- Смена пароля учетной записи.
- Включение и выключение двухфакторной аутентификации для учетной записи.

**Примечание**

Выключение двухфакторной аутентификации возможно, если она не требуется *по парольной политике*.

#### 6.2.5.1 Смена пароля учетной записи

1. В верхнем правом углу веб-интерфейса нажмите на значок с символом логина текущей учетной записи и выберите `Безопасность учетной записи`.

   Откроется окно настройки безопасности для текущей учетной записи.

2. Введите текущий пароль, новый пароль, повторите новый пароль и нажмите `Изменить пароль`.

   Новый пароль не должен повторять текущий пароль, не должен повторять логин без учета регистра и должен соответствовать *требованиям парольной политики*. В случае прохождения всех проверок пароль будет изменен и текущая пользовательская сессия будет завершена.

3. Войдите в систему с новым паролем.

**Совет**

Администратор может *вручную сбросить пароль* для локального пользователя, например, если пользователь забыл пароль. После этого пользователь должен заново войти в систему с новым паролем, назначенным администратором.

#### 6.2.5.2 Включение двухфакторной аутентификации

Для включения двухфакторной аутентификации можно использовать, например, приложения Яндекс.Ключ, Google Authenticator или Multifactor.

**Важно**

После пяти неудачных попыток ввода одноразового кода в течение пяти минут пользователь будет заблокирован на пять минут. Коды восстановления не поддерживаются.

1. В верхнем правом углу веб-интерфейса нажмите на значок с символом логина текущей учетной записи и выберите `Безопасность учетной записи`.

   Откроется окно настройки безопасности для текущей учетной записи.

2. В блоке `Включить TOTP` введите пароль от текущей учетной записи и нажмите `Начать настройку`.

   Отобразятся QR-код и секрет для подключения приложения-аутентификатора.

3. Отсканируйте QR-код в приложении-аутентификаторе на своем мобильном телефоне. Если вы работаете на компьютере, то вместо QR-кода скопируйте и введите в приложении секрет.

4. Вернитесь в Angie ADC и введите одноразовый код аутентификации из приложения-аутентификатора.

5. Нажмите `Подтвердить и включить`.

   Двухфакторная аутентификация будет включена для текущей учетной записи. Текущая пользовательская сессия будет завершена.

6. Войдите в систему заново, введя свой пароль и одноразовый код.

**Совет**

Администратор может *вручную сбросить настройки двухфакторной аутентификации* для локального пользователя, например, если пользователь больше не имеет доступа к подключенному приложению для аутентификации. После этого пользователь должен заново войти в систему и настроить двухфакторную аутентификацию, если этого требует *парольная политика*.

#### 6.2.5.3 Отключение двухфакторной аутентификации

1. В верхнем правом углу веб-интерфейса нажмите на значок с символом логина текущей учетной записи и выберите `Безопасность учетной записи`.

   Откроется окно настройки безопасности для текущей учетной записи.

2. В блоке `Отключить TOTP` введите пароль от текущей учетной записи, одноразовый код из подключенного приложения-аутентификатора и нажмите `Отключить TOTP`.

   Двухфакторная аутентификация для текущей учетной записи будет отключена. Текущая пользовательская сессия будет завершена.

3. Войдите в систему заново, введя свой пароль.

### 6.2.6 Аудит действий пользователей

Действия пользователей, выполняемые в веб-интерфейсе, записываются в журнал событий Angie ADC и имеют помету `"msg":"audit"`.

> **Примечание**
>
> На данный момент логируются только действия, выполняемые через API. В дальнейших релизах будет реализовано логирование действий пользователей, выполняемых через CLI.

Просмотреть журнал событий можно через CLI. Для этого необходимо ввести команды `logs` → `adc-server`.

При *отправке на syslog-сервер* события аудита помечаются с помощью пользовательской категории `facility=local4` (в соответствии с RFC 3164). Вы можете настроить фильтрацию по `facility=local4` на своем syslog-сервере для обработки только аудит-логов.

В сообщениях аудита выводятся следующие поля:

- `time` — время фиксации события.
- `level` — уровень критичности события: всегда `INFO` (информационное событие).
- `msg` — тип события: всегда `audit` (аудит действий пользователя).
- `app` — устройство, на котором произошло событие.
- `node_id` — имя устройства, на котором произошло событие.
- `actor_id` — идентификатор пользователя, инициировавшего событие.
- `actor_name` — имя пользователя, инициировавшего событие.
- `actor_auth_source` — тип аутентификации пользователя, например локальная или RADIUS.
- `resource` — категория объекта, к которому относится операция.
- `operation` — выполняемая операция, например `read`, `update`, `delete`.
- `mutating` — изменяет ли выполняемая операция систему (`true`, `false`).
- `resource_id` — идентификатор конкретного экземпляра объекта, если применимо.
- `actor_host` — IP-адрес клиента.
- `client_channel` — канал взаимодействия, например веб-интерфейс, CLI или межсервисный вызов.
- `duration_ms` — время выполнения операции в миллисекундах.
- `result` — результат выполнения (`OK`, `InvalidArgument`).
- `stage` — стадия выполнения операции: `pre` — намерение выполнить действие, `post` — результат выполнения.

- `request_id` — идентификатор запроса, связывающий события одной операции.
- `trace_id` — идентификатор цепочки выполнения запроса.
- `span_id` — идентификатор конкретной операции внутри цепочки выполнения запроса.

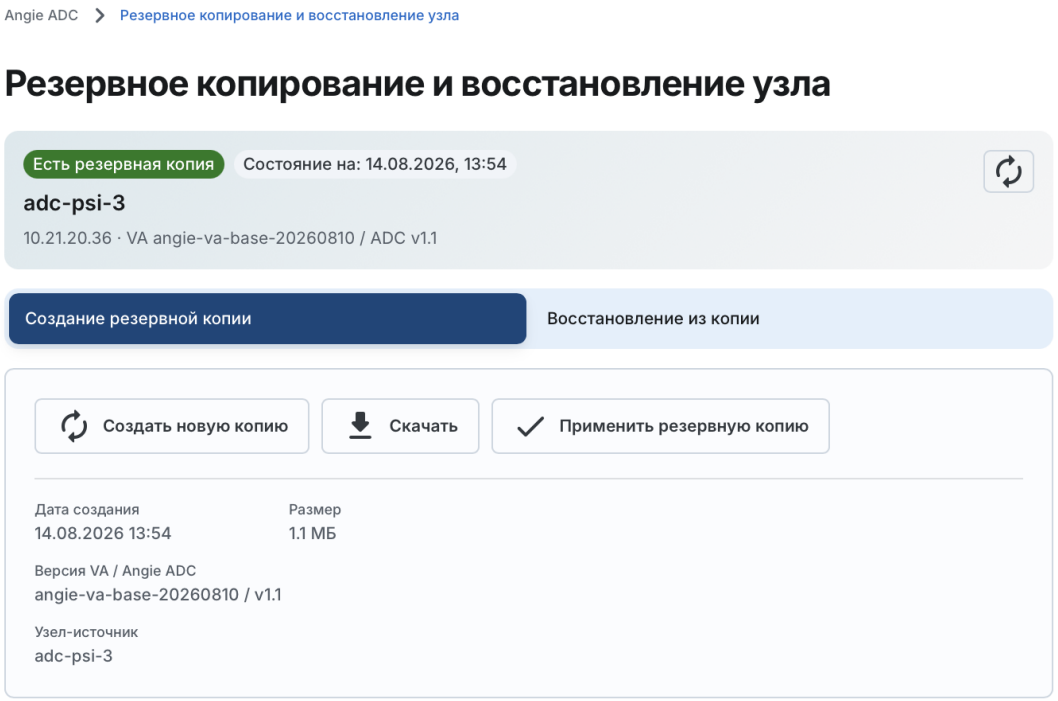
## 6.3 Резервное копирование и восстановление системы

Для защиты от сбоев и возврата к стабильному состоянию в Angie ADC предусмотрены механизмы резервного копирования и восстановления узлов. Они позволяют выполнить следующие задачи:

- Резервное копирование: создание копии текущего узла.
- Откат изменений: восстановление работоспособности текущего узла из сохраненной копии в случае сбоя.
- Миграция узла: перенос резервной копии на новое оборудование.

**Важно**

Рекомендуется создавать или обновлять резервную копию узла перед внесением в систему значительных изменений.




При переносе резервной копии на новый узел он станет полной копией исходного узла:

- сохранятся IP-адреса и сетевая конфигурация исходного узла;
- будут перенесены все данные и настройки;
- если узел был в *паре высокой доступности*, то пара восстановится;
- если узел был в *кластере*, то соседи снова будут видеть этот узел как исходный.

**Важно**

Если вы проводите восстановление из резервной копии на новом узле, то исходный узел необходимо отключить, иначе возникнет конфликт IP-адресов.

### 6.3.1 Статусы узла

Статусы узла отображают текущее состояние узла или этапы резервного копирования и восстановления.

В блоке `Создание резервной копии` у узла могут быть следующие статусы:

- `Копия еще не создана`;
- `Создание копии...`;
- `Есть резервная копия`;
- `Не удалось создать резервную копию`.

В блоке `Восстановление из копии` у узла могут быть следующие статусы:

- `Файл восстановления еще не загружен`;
- `Готов к восстановлению`;
- `Проверка архива`;
- `Не удалось проверить архив`;
- `Запущено восстановление из резервной копии`;
- `Восстановление завершено`;
- `Восстановление не удалось`;
- `Восстановлен из копии`.

### 6.3.2 Создание резервной копии

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Резервное копирование и восстановление`.
2. Выберите `Создание резервной копии` и нажмите `Создать копию`.

   Будет запущен процесс создания резервной копии текущего узла.

После завершения процесса отобразится информация о созданной копии:

- дата создания;
- размер;
- версия VA и версия Angie ADC в копии;
- узел-источник копии (hostname).

### 6.3.3 Обновление резервной копии

**Примечание**

Поддерживается хранение только одной резервной копии узла, при загрузке новой копии предыдущая удаляется.

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Резервное копирование и восстановление`.

2. Выберите `Создание резервной копии` и нажмите `Создать новую копию`.

   Будет запущен процесс создания новой резервной копии текущего узла. Старая копия будет удалена.

После завершения процесса отобразится информация о новой копии:

- дата создания;
- размер;
- версия VA и версия Angie ADC в копии;
- узел-источник копии (hostname).

### 6.3.4 Скачивание резервной копии

Вы можете скачать резервную копию текущего узла, чтобы затем применить ее на новом узле.

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Резервное копирование и восстановление`.
2. Выберите `Создание резервной копии`.

   Отобразится информация о текущей резервной копии.
3. Если необходимо, обновите копию, нажав `Создать новую копию`.
4. Нажмите `Скачать`.

   Будет загружен файл `.tar`, содержащий текущую резервную копию узла.

### 6.3.5 Восстановление текущего узла из резервной копии

1. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Резервное копирование и восстановление`.
2. Выберите `Создание резервной копии` и нажмите `Применить резервную копию`.
3. В открывшемся окне подтвердите операцию, нажав `Начать восстановление`.

После завершения восстановления система перезагрузится. Веб-интерфейс будет доступен по прежнему адресу.

> **Примечание**
>
> Если восстановление завершилось с ошибкой, то по кнопке `Скачать лог` вы можете загрузить файл `restore.log` и получить диагностическую информацию.
>
> Если в процессе восстановления был потерян доступ к веб-интерфейсу, то файл `restore.log` можно просмотреть *через мастер настройки*.

### 6.3.6 Восстановление из резервной копии на новом узле

1. Разверните Angie ADC на новом узле и назначьте временный IP-адрес для веб-интерфейса. Другие настройки не требуются.

   > **Важно**
   >
   > При восстановлении из резервной копии рекомендуется, чтобы версия VA и версия Angie ADC в копии и на новом узле совпадали.

2. В веб-интерфейсе Angie ADC перейдите в раздел `Система` → `Резервное копирование и восстановление`.

3. Выберите `Восстановление из копии` и загрузите файл `.tar`, содержащий резервную копию узла, который вы хотите восстановить.

   Система проверит файл и отобразит сводную таблицу с основными параметрами текущего узла и загруженной копии.

4. Если были выявлены критические несовпадения и необходимо заменить копию на более подходящую, нажмите `Заменить файл` и загрузите новый архив.

   **Важно**

   Не рекомендуется начинать восстановление, если версии VA или Angie ADC не совпадают.

5. Нажмите `Начать восстановление`.

После завершения восстановления система перезагрузится. Веб-интерфейс будет доступен по адресу, который использовался в копии. В статусе узла будет отображаться `Восстановлен из копии`.

**Примечание**

Если восстановление завершилось с ошибкой, то по кнопке `Скачать лог` вы можете загрузить файл `restore.log` и получить диагностическую информацию.

Если в процессе восстановления был потерян доступ к веб-интерфейсу, то файл `restore.log` можно просмотреть *через мастер настройки*.

**Совет**

- Чтобы просмотреть данные копии, которая использовалась для восстановления, перейдите в блок `Восстановление из копии`.
- Чтобы заново запустить восстановление из копии, в блоке `Восстановление из копии` нажмите `Загрузить файл` и загрузите архив еще раз.

ГЛАВА 7

# Мастер настройки

Мастер настройки позволяет управлять конфигурацией Angie ADC без доступа к веб-интерфейсу. Также в мастере можно просмотреть сетевую конфигурацию и проверить доступность узла, выполнив ping-запрос.

**При первом запуске** Angie ADC вам необходимо настроить хотя бы один *сетевой интерфейс* для доступа к веб-интерфейсу Angie ADC.

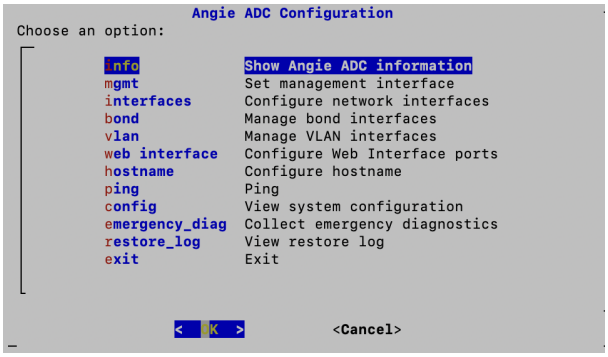
**При расширенной настройке** вы можете дополнительно настроить другие интерфейсы, бонды и VLAN-интерфейсы, назначить выделенный интерфейс управления, изменить имя устройства и порт для веб-интерфейса.

**Для входа в мастер** необходимо перейти в консоль виртуальной машины и в окне входа в мастер ввести предустановленные логин `setup` и пароль `setup`. Чтобы выйти из мастера настройки Angie ADC, нажмите `exit` в основном меню.

В этой статье:

- *Меню мастера*;
- *Настройка сетевых интерфейсов*;
- *Настройка интерфейса управления*;
- *Смена портов для веб-интерфейса Angie ADC*;
- *Создание бондов*;
- *Создание VLAN-интерфейсов*;
- *Просмотр конфигурации Angie ADC и сетевых настроек*;
- *Изменение имени устройства Angie ADC*;
- *Проверка доступности узла*;
- *Аварийная диагностика*;
- *Просмотр журнала восстановления из резервной копии*.

## 7.1 Меню мастера




Основное меню мастера настройки Angie ADC содержит следующие команды:

- `info` — просмотр общей информации об Angie ADC.
- `mgmt` — настройка интерфейса управления Angie ADC.
- `interfaces` — переход к просмотру и настройке сетевых интерфейсов, бондов и VLAN-интерфейсов.
- `bond` — переход к созданию, просмотру и удалению сетевых бондов.
- `vlan` — переход к созданию, просмотру и удалению VLAN-интерфейсов.
- `web interface` — изменение порта для веб-интерфейса Angie ADC.
- `hostname` — смена имени устройства.
- `config` — просмотр конфигурации Angie ADC и сетевых настроек.
- `ping` — проверка доступности узла.
- `emergency_diag` — аварийный сбор логов для передачи в техническую поддержку.
- `restore_log` — просмотр журнала восстановления узла из резервной копии.
- `exit` — выход из мастера.

## 7.2 Настройка сетевых интерфейсов

Чтобы настроить сетевой интерфейс Angie ADC, выполните следующие действия:

1. Выберите в основном меню пункт `interfaces`.

   Откроется список доступных интерфейсов.

2. Выберите интерфейс и перейдите в его меню.

   Откроется список доступных действий:

   - `show` — просмотр информации об интерфейсе.
   - `config` — переход к ручной настройке IP-адреса и шлюза (Gateway).
   - `dhcp` — настройка автоматического получения IP-адреса от DHCP-сервера.
   - `mtu` — переход к настройке MTU.
   - `clear` — сброс всех заданных настроек интерфейса, кроме MTU. Значение MTU, заданное вручную, не сбрасывается.
   - `back` — выход из меню интерфейса.

3. Выберите `config` или `dhcp` в зависимости от ваших целей и задайте необходимые настройки.

## 7.3 Настройка интерфейса управления

Рекомендуется использовать для управления отдельную сеть, в которой нет служебных сервисов, иначе эти сервисы перестанут быть доступными для других компонентов Angie ADC. Если вы хотите сохранить доступ к этим сервисам, после выделения интерфейса управления настройте для них маршруты-исключения.

Чтобы выделить отдельный интерфейс для управления Angie ADC, выполните следующие действия:

1. Выберите в основном меню пункт `mgmt`.

   Откроется список доступных действий:

   - `setup` — переход к настройке интерфейса управления.
   - `status` — просмотр текущего статуса интерфейса управления.
   - `leak` — переход к настройке маршрутов-исключений для сервисов внутри сети управления (пункт отображается, если интерфейс управления настроен).
   - `back` — выход из меню `mgmt`.

2. В открывшемся меню выберите `setup`.

   Откроется список доступных интерфейсов.

3. Выберите нужный интерфейс. Этот интерфейс будет использоваться для управления системой через отдельную сеть и будет иметь собственную таблицу маршрутизации.

4. Подтвердите операцию.

   > **Предупреждение**
   >
   > Служебные сервисы, расположенные в этой сети, перестанут быть доступными для других компонентов Angie ADC.

5. Если необходимо, настройте маршруты-исключения для служебных сервисов. Для этого перейдите в меню `mgmt` → `leak` → `add` и добавьте сеть назначения, из которой будет производиться доступ к сервисам, находящимся в сети управления.

## 7.4 Создание бондов

При создании бондов используется протокол LACP (802.3ad) и режим active (интерфейс инициирует создание агрегированного канала).

Чтобы объединить несколько физических интерфейсов в один логический интерфейс (бонд), выполните следующие действия:

1. Выберите в основном меню пункт `bond`.

   Откроется список доступных действий:

   - `list` — просмотр списка бондов.
   - `create` — переход к созданию нового бонда.
   - `back` — выход из меню `bond`.

2. Выберите `create`, укажите номер бонда (по умолчанию будет предложен первый свободный номер по возрастанию) и нажмите `OK`.

3. Выберите интерфейсы, которые необходимо объединить (максимум 8 интерфейсов), и нажмите `OK`.

> **Примечание**
>
> Рекомендуется объединять в бонд ненастроенные сетевые интерфейсы.

4. В открывшемся окне с конфигурацией подтвердите создание нового бонда.

Просмотр свойств бонда и его удаление доступны через основное меню `bond` → `list` → `<bond_name>`.

Настройка бонда доступна через основное меню настройки интерфейсов: `interfaces` → `<bond_name>`.

## 7.5 Создание VLAN-интерфейсов

Чтобы настроить VLAN-интерфейс, выполните следующие действия:

1. Выберите в основном меню пункт `vlan`.

   Откроется список доступных действий:

   - `list` — просмотр списка VLAN-интерфейсов.
   - `create` — переход к созданию нового VLAN-интерфейса.
   - `back` — выход из меню `vlan`.

2. Выберите `create`, укажите родительский интерфейс, VLAN ID для нового VLAN и нажмите `OK`.

   Будет создан новый VLAN-интерфейс.

Просмотр свойств VLAN и его удаление доступны через основное меню `vlan` → `list` → `<vlan_name>`.

Настройка VLAN доступна через основное меню настройки интерфейсов: `interfaces` → `<vlan_name>`.

## 7.6 Просмотр конфигурации Angie ADC и сетевых настроек

Чтобы просмотреть текущую конфигурацию Angie ADC или конфигурацию systemd-networkd для заданных сетей, выполните следующие действия:

1. Выберите в основном меню пункт `config`.

   Откроется список доступных действий:

   - `system` — просмотр сетевой конфигурации Angie ADC.
   - `network` — просмотр конфигурации systemd-networkd для заданных сетей.
   - `back` — выход из меню `config`.

2. Выберите `system`, чтобы просмотреть текущую конфигурацию Angie ADC.
3. Выберите `network` → `<имя сети>`, чтобы просмотреть настройки systemd-networkd для этой сети.

## 7.7 Смена портов для веб-интерфейса Angie ADC

Вы можете назначить для веб-интерфейса Angie ADC любой незанятый порт. Чтобы изменить порты по умолчанию, выполните следующие действия:

1. Выберите в основном меню пункт `web interface`.

   Откроется список доступных действий:

   - `set` — назначение новых портов.

- `reset` — восстановление настроек по умолчанию.
- `logs` — просмотр журнала.
- `back` — выход из меню `web interface`.

2. Выберите `set`, укажите новые значения портов и примените настройки.

## 7.8 Изменение имени устройства Angie ADC

Чтобы изменить имя устройства Angie ADC, выполните следующие действия:

1. Выберите в основном меню пункт `hostname`.
2. В открывшемся окне введите новое имя устройства и сохраните изменения.

## 7.9 Проверка доступности узла

Чтобы проверить доступность узла с помощью ping-запроса, выполните следующие действия:

1. Выберите в основном меню пункт `ping`.
2. В открывшемся окне введите имя или IP-адрес узла, доступность которого нужно проверить.

## 7.10 Аварийная диагностика

Для случаев, когда нет доступа к CLI и веб-интерфейсу Angie ADC, в мастере предусмотрена возможность получения аварийных логов. Будет собран архив, содержащий системные метрики, конфигурацию системы и системный журнал.

Чтобы получить архив:

1. В основном меню выберите пункт `emergency_diag`.
2. Дождитесь создания архива. После этого в мастере отобразятся имя архива, порт SFTP и срок, в течение которого архив доступен для скачивания (15 минут). Через 15 минут архив будет удален.

> **Примечание**
>
> При повторном запуске команды `emergency_diag` в течение 15 минут текущий архив будет удален и будет создан новый архив.

3. Подключитесь по SFTP под учетной записью `setup`:

```
$ sftp -P 2223 setup@<IP-адрес>
```

> **Примечание**
>
> Если был настроен интерфейс управления, используйте его IP-адрес.

4. Проверьте наличие архива (команда `ls`) и скачайте его:

```
sftp> get diag_YYYYMMDD_HHMMSS.tar.gz
```

Для штатного сбора диагностической информации используйте команду diag.

## 7.11 Просмотр журнала восстановления из резервной копии

Для случаев, когда после восстановления узла из резервной копии пропал доступ к веб-интерфейсу Angie ADC, в мастере предусмотрена возможность просмотреть журнал восстановления.

Чтобы просмотреть журнал восстановления, выберите в основном меню пункт `restore_log`.

На экране отобразится содержимое файла `restore.log`.

ГЛАВА 8

# Кластеризация

## 8.1 Обзор

Кластер управления Angie ADC позволяет настраивать несколько устройств Angie ADC с любого узла через общую кластерную конфигурацию. В нее входят:

- конфигурация GSLB;
- сертификаты;
- пользователи SNMP;
- конфигурация мониторинга доступности;
- настройки парольной политики.

Данные между узлами кластера синхронизируются и передаются в зашифрованном виде.

**Важно**

Рекомендуется создавать или обновлять *резервную копию узла* перед внесением в систему значительных изменений.

**Примечание**

Конфигурация локального балансировщика не синхронизируется между узлами кластера.

## 8.2 Применение и отмена изменений

После применения изменений на одном из узлов кластера они автоматически синхронизируются на все остальные узлы кластера.

В случае, если применение изменений в кластере по каким-то причинам занимает долгое время, вы можете отменить применение новой конфигурации, нажав кнопку `Отменить изменения` в окне, информирующем об внесении изменений.

**Предупреждение**

Применение изменений, распространяющихся на весь кластер, невозможно, если хотя бы один узел кластера недоступен. Рекомендуется ввести в строй недоступные узлы перед внесением изменений в кластерную конфигурацию.

В этом разделе:

- *Создание и масштабирование кластера управления*
- *Мониторинг состояния кластера управления*

### 8.2.1 Создание и масштабирование кластера управления

Кластер управления Angie ADC состоит из нескольких узлов (устройств Angie ADC). Один из узлов кластера является точкой входа в кластер (к нему подключаются остальные узлы кластера).

Развертывание кластера состоит из следующих этапов:

1. Развертывание устройств Angie ADC в дата-центрах.
2. *Первоначальная настройка узлов*.
3. *Создание точки входа в кластер* на одном из узлов.
4. *Добавление остальных узлов в кластер*.

**Предупреждение**

В качестве IP-адресов узлов кластера нельзя указывать *IP-адрес интерфейса управления*, если он был настроен.

**Важно**

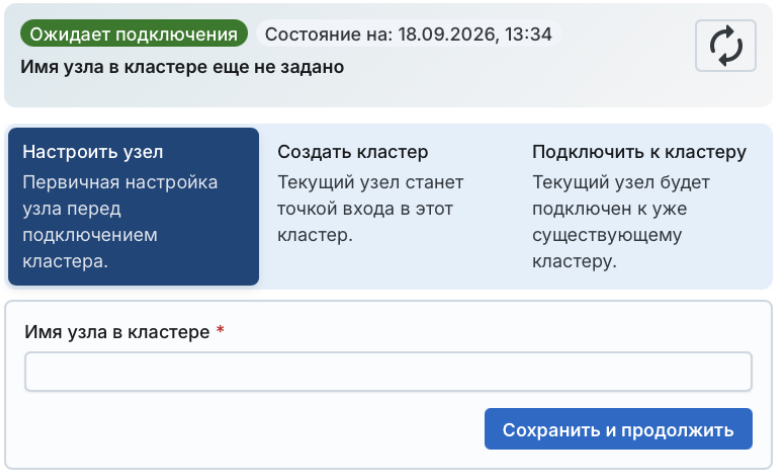
Рекомендуется обязательно выполнять первичную настройку NTP на всех узлах кластера, так как синхронизация времени на узлах с единым внешним источником критически важна для работы кластера.

#### 8.2.1.1 Первоначальная настройка узла

Чтобы выполнить первоначальную настройку узла Angie ADC после установки:

1. Откройте веб-интерфейс Angie ADC на узле.

   Откроется экран с информацией о текущем узле и вариантами его интеграции в кластер.




2. Выберите сценарий `Настроить узел`.
3. Задайте имя текущего узла в кластере (произвольное). Допускаются строчные латинские буквы, цифры и одиночные дефисы внутри имени. Это имя может отличаться от `hostname` узла и отображает логическую структуру кластера управления.
4. Нажмите `Сохранить и продолжить`.

   В веб-интерфейсе откроется окно авторизации.
5. Авторизуйтесь в системе. При первом входе система также запросит смену пароля.
6. В окне с запросом на настройку NTP включите переключатель `Синхронизация времени (NTP)`.

> **Важно**
>
> Рекомендуется обязательно выполнять настройку NTP, так как синхронизация времени на узлах с единым внешним источником критически важна для работы кластера.

7. Нажмите `Добавить сервер` и заполните необходимые поля:
   - Адрес NTP-сервера, например `ntp1.vniiftri.ru`.
   - Минимальный и максимальный интервалы для частоты опроса сервера. Частота опроса может меняться динамически: при нестабильном времени сервер опрашивается чаще, при стабильной синхронизации — реже.

     В качестве значения указывается степень двойки, интервал опроса NTP-сервера рассчитывается как 2 в степени n секунд.

     Например:

     - `6` → $2^6 = 64$ секунды;
     - `10` → $2^{10} = 1024$ секунды (~17 минут).
   - Установите флажок `NTS`, если хотите использовать аутентифицированную синхронизацию времени.

> **Примечание**
>
> NTP также можно настроить в веб-интерфейсе в разделе `Система` → `NTP` и через CLI.

8. Нажмите `Сохранить`.

Новые настройки узла будут применены.

Далее можно перейти к созданию кластера или подключению узла к уже существующему кластеру.

#### 8.2.1.2 Создание точки входа в кластер

> **Предупреждение**
>
> После создания точки входа ее нельзя изменить, кластер нельзя расформировать. Внимательно проверьте все параметры перед созданием точки входа.

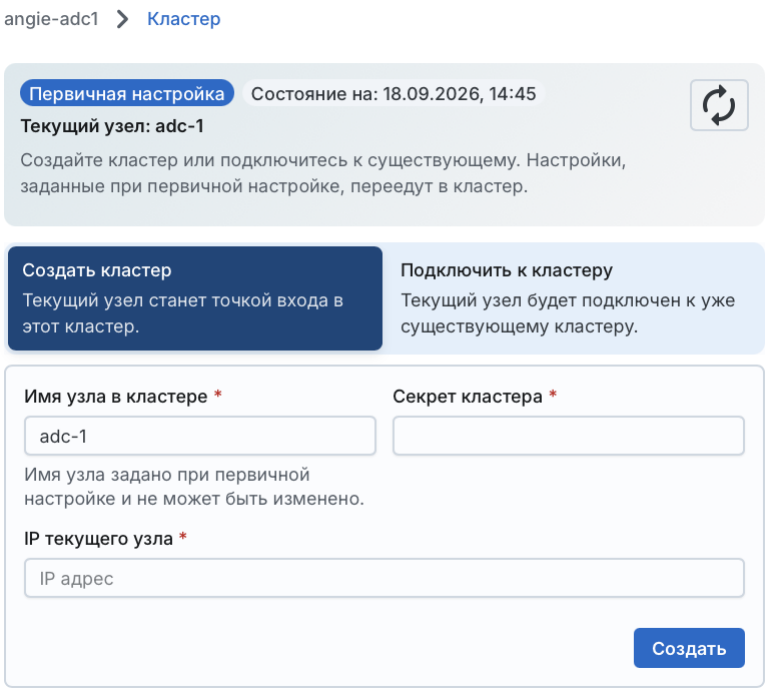
Чтобы создать точку входа в кластер, выполните следующие действия:

1. Откройте веб-интерфейс Angie ADC на узле, который будет точкой входа.

   Откроется экран с информацией о текущем узле и вариантами его интеграции в кластер.

2. Выполните *первоначальную настройку узла* (рекомендуется).

   Откроется экран с информацией о текущем узле и вариантами его интеграции в кластер.




3. Выберите сценарий `Создать кластер` и заполните следующие поля:

   - Секрет для подключения к кластеру.
   - IP-адрес текущего узла (только IPv4).

   > **Важно**
   >
   > В качестве IP-адреса текущего узла нельзя использовать *IP-адрес интерфейса управления*, если он был настроен.

4. Нажмите `Создать`.

   Будет создан кластер. Текущий узел будет назначен точкой входа в кластер.

#### 8.2.1.3 Добавление узлов в кластер

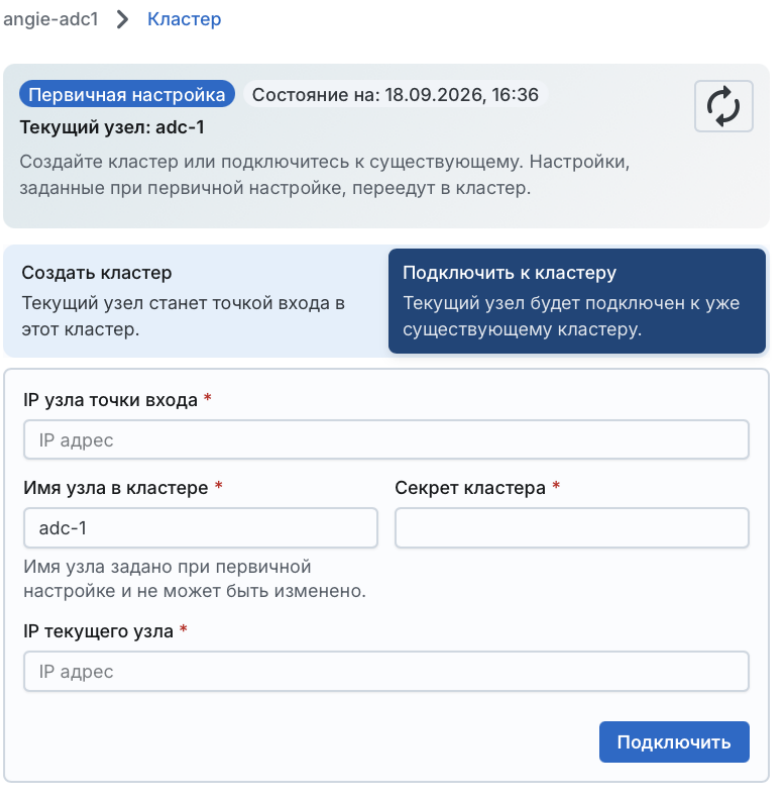
Для каждого последующего узла кластера выполните следующие действия:

1. Перейдите на узел Angie ADC, который вы хотите добавить в кластер.

   Откроется экран с информацией о текущем узле и вариантами его интеграции в кластер.

2. Выполните *первоначальную настройку узла* (рекомендуется).

   Откроется экран с информацией о текущем узле и вариантами его интеграции в кластер.




3. Выберите сценарий `Подключить к кластеру` и заполните следующие поля:

   - IP-адрес узла точки входа (только IPv4);
   - Секрет для подключения к кластеру.
   - IP-адрес текущего узла (только IPv4).

   > **Важно**
   >
   > В качестве IP-адреса текущего узла нельзя использовать *IP-адрес интерфейса управления*, если он был настроен.

4. Нажмите `Подключить`.

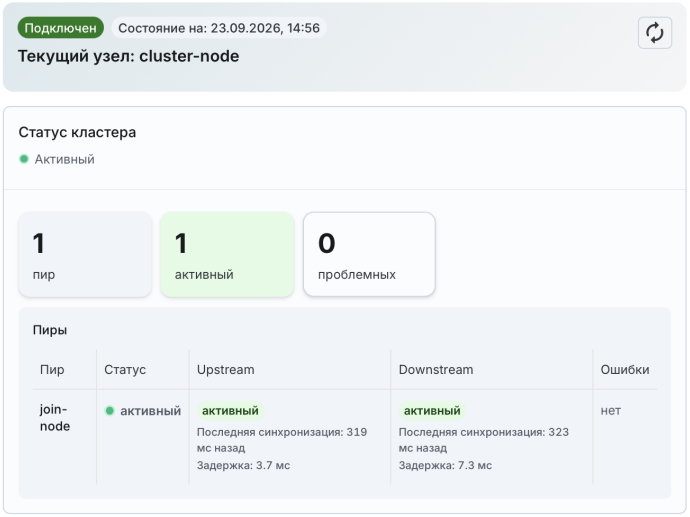
Узел будет подключен к кластеру.

### 8.2.2 Мониторинг состояния кластера управления

После создания кластера управления на странице `Настройки кластера` отображаются:

- Имя текущего узла в кластере.

  Имя узла в кластере управления может отличаться от `hostname` узла и отображает логическую структуру кластера.




- Статус подключения текущего узла к кластеру управления:
  - `Ожидает подключения` – узел не входит в кластер.
  - `Подключен` – узел подключен к кластеру.
  - `Подключение...` – выполняется подключение узла к кластеру.
- Статус кластера управления:
  - `Активный` – кластер запущен.
  - `Подключение` – выполняется запуск кластера.
- Сводная статистика других узлов кластера (пиров текущего узла).
- Таблица с подробной информацией по каждому пиру:

| | |
|---|---|
| `Пир` | Имя узла, входящего в кластер. |
| `Статус` | • `активный` – пир находится в рабочем состоянии, данные синхронизированы.<br>• `сбой` – есть проблемы с синхронизацией данных.<br>• `недоступен` – пир недоступен. |
| `Upstream` | Статус получения данных пиром:<br>• `активный` – upstream доступен, данные синхронизированы.<br>• `подключение` – выполняется подключение к кластеру.<br>• `синхронизация` – выполняется синхронизация (получение данных пиром).<br>• `недоступен` – upstream недоступен, получение данных невозможно. |
| `Downstream` | Статус передачи данных пиром:<br>• `активный` – downstream доступен, данные могут передаваться.<br>• `подключение` – выполняется подключение к кластеру.<br>• `недоступен` – downstream недоступен, передача данных невозможна. |
| `Последняя синхронизация` | Время, прошедшее с последней синхронизации данных. |
| `Задержка` | Задержка синхронизации данных между текущим узлом и пиром. |
| `Ошибки` | Ошибки, возникающие при синхронизации данных для каждого пира. |

ГЛАВА 9

# Права на интеллектуальную собственность

Документация на программный продукт Angie ADC является интеллектуальной собственностью ООО «Веб-Сервер».