

# ANIC

Angie Ingress Controller

*версия 0.5.0*

Руководство по эксплуатации

сент. 27, 2024

---

## Оглавление

---

<b>1</b>	<b>Аннотация</b>	<b>1</b>
1.1	Общие сведения . . . . .	1
1.2	Системные требования . . . . .	2
<b>2</b>	<b>Настройка</b>	<b>3</b>
2.1	Параметры Ingress Controller . . . . .	3
2.2	Общие параметры . . . . .	3
2.3	Параметры ведения журнала . . . . .	5
2.4	Управление URI и заголовками в запросах . . . . .	5
2.5	Авторизация и SSL/TLS . . . . .	6
2.6	Протоколы . . . . .	6
2.7	Апстримы . . . . .	6
2.8	Настраиваемые шаблоны . . . . .	7
<b>3</b>	<b>ConfigMap</b>	<b>8</b>
3.1	Использование ConfigMap . . . . .	8
3.2	ConfigMap и аннотации Ingress . . . . .	9
3.3	Переопределение ConfigMap для конкретного ресурса Ingress с помощью аннотации . . . . .	9
3.4	ConfigMap и ресурсы VirtualServer, VirtualServerRoute . . . . .	10
3.5	Краткое описание ключей ConfigMap . . . . .	10
<b>4</b>	<b>GlobalConfiguration</b>	<b>18</b>
4.1	Предварительные требования . . . . .	18
4.2	Спецификация GlobalConfiguration . . . . .	18
4.3	Использование GlobalConfiguration . . . . .	19
<b>5</b>	<b>Policy</b>	<b>22</b>
5.1	Предварительные требования . . . . .	22
5.2	Спецификация Policy . . . . .	22
<b>6</b>	<b>TransportServer</b>	<b>38</b>
6.1	Предварительные требования . . . . .	38
6.2	Спецификация TransportServer . . . . .	38
6.3	Использование TransportServer . . . . .	44
<b>7</b>	<b>VirtualServer, VirtualServerRoute</b>	<b>48</b>
7.1	Спецификация VirtualServer . . . . .	48
7.2	Спецификация VirtualServerRoute . . . . .	58
7.3	Общие части VirtualServer и VirtualServerRoute . . . . .	61
7.4	Использование VirtualServer и VirtualServerRoute . . . . .	76
7.5	Настройка с помощью ConfigMap . . . . .	79

<b>8</b>	<b>Отчеты о состоянии ресурсов</b>	<b>80</b>
8.1	Ресурсы Ingress . . . . .	80
8.2	Ресурсы VirtualServer и VirtualServerRoute . . . . .	81
8.3	Ресурсы Policy . . . . .	82
8.4	Ресурсы TransportServer . . . . .	83
<b>9</b>	<b>Просмотр состояния сервера</b>	<b>84</b>
9.1	Доступ к Stub Status . . . . .	84
9.2	Доступ к веб-панели мониторинга Console Light . . . . .	85
<b>10</b>	<b>Ведение журнала</b>	<b>86</b>
10.1	Логи процесса Ingress Controller . . . . .	86
10.2	Логи Angie . . . . .	86
<b>11</b>	<b>Пользовательские шаблоны</b>	<b>87</b>

# ГЛАВА 1

---

## Аннотация

---

Angie Ingress Controller (ANIC) — приложение, которое запускается в кластере и управляет балансировщиком нагрузки.

ANIC использует в своей работе Angie PRO — эффективный, мощный и масштабируемый веб-сервер, который позволяет балансировать нагрузку между серверами как по протоколам TCP/UDP, так и по HTTP.

### Примечание

Angie PRO внесён в Единый реестр российских программ для электронных вычислительных машин и баз данных (запись № 17604).

## 1.1 Общие сведения

Angie Ingress Controller (ANIC) - это решение для управления трафиком контейнеризированных приложений в Kubernetes.

ANIC разворачивается и работает в кластере, управляя функциями Ingress с возможностью настройки правил обработки трафика. Продукт базируется на Angie PRO, что позволяет строить безопасные масштабируемые высокопроизводительные окружения, используя российское решение с профессиональными сервисами миграции и технической поддержки на русском языке.

ANIC использует широкий набор функций Ingress:

- *Балансировка нагрузки TCP, UDP, TLS, HTTP, gRPC*: Гибкое распределение трафика и его плавного переноса при обновлениях приложений
- *Терминирование сессий TLS*: Подтверждения подлинности сервисов и защиты онлайн-транзакций
- *Настройки гибкого логирования*: Управление современными динамическими приложениями
- *Расширенная маршрутизация трафика*: Разделение трафика и расширенная маршрутизация на основе содержимого
- *Ограничение поступающего трафика*: По различным критериям для защиты приложений от DDoS

- *Модификация ответов на запросы:* На уровне балансировщика HTTP

## 1.2 Системные требования

Список поддерживаемых ОС и архитектур:

ОС	Версии	Архитектуры
Alpine Linux	3.17	x86_64, arm64
Alt Linux	10	x86_64, arm64
Debian	11	x86_64, arm64

## ГЛАВА 2

### Настройка

ANIC настраивается путем изменения параметров ConfigMap и Annotation:

Список 1: Пример ConfigMap

```
$ kubectl apply -f - <<EOF
kind: ConfigMap
apiVersion: v1
metadata:
  name: angie-config
  namespace: angie-ingress
data:
  proxy-connect-timeout: "10s"
  proxy-read-timeout: "10s"
  client-max-body-size: "2m"
EOF
```

## 2.1 Параметры Ingress Controller

**external-status-** Задаёт адрес, который выводится в статусе Ingress ресурса. Имеет приоритет над аргументом командной строки **-external-service**.

## 2.2 Общие параметры

### Примечание

Для всех параметров типа **boolean** допустимы пары значений «true»/»false», «t»/»f», «on»/»off» и «1»/»0». Регистр не имеет значения.

Параметр	Описание	Умолчение
proxy-connect-timeout	Задаёт значение proxy_connect_timeout и grpc_connect_timeout.	60s
proxy-read-timeout	Задаёт значение proxy_read_timeout и grpc_read_timeout	60s
proxy-send-timeout	Задаёт значение proxy_send_timeout и grpc_send_timeout	60s
client-max-body-size	Задаёт значение client_max_body_size	1m
proxy-buffering	Включает или отключает буферизацию ответа от проксируемого сервера	True
proxy-buffers	Задаёт значение proxy_buffers	Зависит от платформы
proxy-buffer-size	Задаёт значение proxy_buffer_size и grpc_buffer_size	Зависит от платформы
proxy-max-temp-file-size	Задаёт значение proxy_max_temp_file_size	1024m
set-real-ip-from	Задаёт значение set_real_ip_from	Нет
real-ip-header	Задаёт значение real_ip_header	X-Real-IP
real-ip-recursive	Включает или отключает real_ip_recursive	False
default-server-return	Настраивает ответ в сервере по умолчанию, который перехватывает клиентский запрос, если для запроса не был определен ресурс Ingress или VirtualServer. Можно установить фиксированный ответ или перенаправление запроса.	Страница с ошибкой HTTP 404
server-tokens	Включает или отключает server_tokens	True
worker-processes	Задаёт значение worker_processes	auto
worker-rlimit-nofile	Задаёт значение worker_rlimit_nofile	Нет
worker-connections	Задаёт значение worker_connections	1024
worker-cpu-affinity	Задаёт значение worker_cpu_affinity	Нет
worker-shutdown-timeout	Задаёт значение worker_shutdown_timeout	Нет
server-names-hash-bucket-size	Задаёт значение server_names_hash_bucket_size	256
server-names-hash-max-size	Задаёт значение server_names_hash_max_size	1024
map-hash-bucket-size	Задаёт значение map_hash_bucket_size	256
map-hash-max-size	Задаёт значение map_hash_max_size	2048
resolver-addresses	Задаёт значение DNS resolver'a	Нет
resolver-ipv6	Разрешает или запрещает поиск IPv6-адресов	True
resolver-valid	Позволяет переопределить срок кэширования DNS-записей	Нет
resolver-timeout	Задаёт значение resolver_timeout	30s
keepalive-timeout	Задаёт значение keepalive_timeout	65s
keepalive-requests	Задаёт значение keepalive_requests	100
variables-hash-bucket-size	Задаёт значение variables_hash_bucket_size	256
variables-hash-max-size	Задаёт значение variables_hash_max_size	1024

## 2.3 Параметры ведения журнала

Параметр	Описание	Умолчание
<code>error-log-level</code>	Определяет глобальное значение уровня <code>error_log</code> и может принимать одно из следующих значений: <code>debug</code> , <code>info</code> , <code>notice</code> , <code>warn</code> , <code>error</code> , <code>crit</code> , <code>alert</code> или <code>emerg</code>	<code>notice</code>
<code>access-log-off</code>	Отключает <code>access_log</code>	<code>False</code>
<code>default-server-access-log-off</code>	Отключает <code>access_log</code> для сервиса по умолчанию	<code>False</code>
<code>log-format</code>	Задаёт общий формат журнала. Для удобства можно использовать несколько строк, разделённых <code>\n</code> . В этом случае каждый перевод строки будет заменён на пробел. Все символы <code>'</code> должны быть экранированы	Нет
<code>log-format-escaping</code>	Позволяет задать экранирование символов <code>json</code> или <code>default</code> в переменных; по умолчанию используется <code>default</code> . Значение <code>none</code> отключает экранирование	<code>default</code>
<code>stream-log-format</code>	Задаёт формат журнала <code>stream</code> для сквозного трафика TCP, UDP и TLS. Для удобства можно использовать несколько строк, разделённых <code>\n</code> . В этом случае каждый перевод строки будет заменён на пробел. Все символы <code>'</code> должны быть экранированы	Нет
<code>stream-log-format-escaping</code>	Позволяет задать экранирование символов <code>json</code> или <code>default</code> в переменных; по умолчанию используется <code>default</code> . Значение <code>none</code> отключает экранирование	<code>default</code>

## 2.4 Управление URI и заголовками в запросах

<code>proxy-hide-headers</code>	Значение одного <code>proxy_hide_header</code> или нескольких
<code>proxy-pass-headers</code>	Значение одного <code>proxy_pass_header</code> или нескольких

## 2.5 Авторизация и SSL/TLS

Параметр	Описание	Умолчение
<code>redirect-to-https</code>	Задаёт правило 301 redirect в зависимости от заголовка <code>http_x_forwarded_proto</code>	False
<code>ssl-redirect</code>	Задаёт правило 301 redirect для всего входящего HTTP-трафика, чтобы перевести запросы в HTTPS	True
<code>ssl-protocols</code>	Задаёт значение <code>ssl_protocols</code>	TLSv1 TLSv1.1 TLSv1.2
<code>ssl-prefer-server-ciphers</code>	Включает или отключает <code>ssl_prefer_server_ciphers</code>	False
<code>ssl-ciphers</code>	Задаёт значение <code>ssl_ciphers</code>	HIGH:!aNULL:!MD5
<code>ssl-dhparam-file</code>	Указывает файл с параметрами для ДНЕ-шифров	Нет

## 2.6 Протоколы

Параметр	Описание	Умолчение
<code>http2</code>	Включает поддержку протокола HTTP/2	False
<code>proxy-protocol</code>	Указывает, что все соединения, принимаемые на данном слушающем сокете, должны использовать протокол PROXY	False

## 2.7 Апстримы

Параметр	Описание	Умолчение
<code>max-fails</code>	Задаёт значение <code>max_fails</code> для сервера	1
<code>upstream-zone-size</code>	Задаёт имя и размер зоны разделяемой памяти	Нет
<code>fail-timeout</code>	Задаёт значение <code>fail_timeout</code> для сервера	10s
<code>keepalive</code>	Задействует кэш соединений для группы серверов апстрима	Нет

## 2.8 Настраиваемые шаблоны

<code>main-snippets</code>	Вставляет собственный фрагмент конфигурации в контекст <code>main</code>
<code>http-snippets</code>	Вставляет собственный фрагмент конфигурации в контекст <code>http</code>
<code>location-snippets</code>	Вставляет собственный фрагмент конфигурации в контекст <code>location</code>
<code>server-snippets</code>	Вставляет собственный фрагмент конфигурации в контекст <code>server</code>
<code>stream-snippets</code>	Вставляет собственный фрагмент конфигурации в контекст <code>stream</code>
<code>main-template</code>	Определяет основной шаблон для основных настроек Angie. По умолчанию шаблон считывается из файла в контейнере
<code>ingress-template</code>	Определяет шаблон настроек для ресурса Ingress. По умолчанию шаблон считывается из файла в контейнере
<code>virtualserver-template</code>	Определяет шаблон настроек для ресурса <i>VirtualServer</i> . По умолчанию шаблон считывается из файла в контейнере

## ГЛАВА 3

---

### ConfigMap

---

Ресурсы ConfigMap позволяют вам настраивать поведение Angie. Например, можно задать количество рабочих процессов или настроить формат журнала доступа.

### 3.1 Использование ConfigMap

1. Наши инструкции по установке с манифестами развертывают пустой ConfigMap, в то время как манифесты установки по умолчанию указывают ее в аргументах командной строки ANIC. Однако, если вы настроили манифесты, чтобы использовать ConfigMap, обязательно укажите ресурс ConfigMap для использования с помощью аргументов командной строки ANIC.
2. Создайте файл ConfigMap с именем `angie-config.yaml` и установите значения, которые имеют смысл для вашей среды:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: angie-config
  namespace: angie-ingress
data:
  proxy-connect-timeout: "10s"
  proxy-read-timeout: "10s"
  client-max-body-size: "2m"
```

См. в разделе *Краткое описание ключей ConfigMap* сведения о доступных ключах ConfigMap (таких как `proxy-connect-timeout` в этом примере).

3. Создайте новый (или обновите существующий) ресурс ConfigMap:

```
kubectl apply -f angie-config.yaml
```

Конфигурация Angie будет обновлена.

## 3.2 ConfigMap и аннотации Ingress

Аннотации позволяют настраивать расширенные функции Angie и менять поведение Angie.

ConfigMap применяется глобально, то есть влияет на каждый ресурс Ingress. Напротив, аннотации всегда применяются только к своему ресурсу Ingress. Аннотации позволяют переопределять некоторые ключи ConfigMap. Например, в `angie.software/proxy-connect-timeout` аннотации переопределяют ключ конфигурации `proxy-connect-timeout`.

## 3.3 Переопределение ConfigMap для конкретного ресурса Ingress с помощью аннотации

Вы можете применять разные конфигурации ConfigMap к Ingress-ресурсам в зависимости от того, какое пространство имен указано в конфигурации. Аннотация `angie.software/configmap` позволяет задать конкретный ConfigMap для настройки ресурса Ingress. Заданный ConfigMap будет иметь приоритет над глобальным. В случае, если глобальный и заданный ConfigMap совпадают, применится заданный.

Чтобы применить конкретный ConfigMap к ресурсу Ingress:

1. Создайте ConfigMap с указанием нужного пространства имен.

Например:

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: echoserver-new-config
  namespace: echoserver-new
data:
  server-snippets: |
    location /echoserver-new-snippet {
      return 302 /echo-test-2;
    }
```

2. Укажите аннотацию `angie.software/configmap` в ресурсе Ingress, к которому нужно применить этот ConfigMap.

Например:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  annotations:
    angie.software/configmap: "echoserver-new/echoserver-new-config"
  name: echoserver-new
  namespace: echoserver-new
spec:
  ingressClassName: angie
  rules:
  - host: test-new.example.com
    http:
      paths:
      - backend:
          service:
            name: echoserver-new
          port:
```

```
number: 8077
pathType: ImplementationSpecific
```

В этом примере аннотация `angie.software/configmap` указывает на использование конфигурации из ConfigMap `echoserver-new-config`. Это означает, что директивы, описанные в `server-snippets` из этого ConfigMap, будут применяться к запросам, обрабатываемым этим Ingress.

См. также документацию по расширенной конфигурации с помощью аннотаций.

### 3.4 ConfigMap и ресурсы VirtualServer, VirtualServerRoute

ConfigMap влияет на все ресурсы VirtualServer и VirtualServerRoute. Однако поля этих ресурсов позволяют переопределять некоторые ключи ConfigMap. Например, поле `connect-timeout` сервера апстрима имеет приоритет над ключом ConfigMap `proxy-connect-timeout`.

См. документацию по *ресурсам VirtualServer и VirtualServerRoute*.

### 3.5 Краткое описание ключей ConfigMap

**Примечание**

Для всех параметров типа `boolean` допустимы пары значений `«true»/»false»`, `«t»/»f»`, `«on»/»off»` и `«1»/»0»`. Регистр не имеет значения.

#### 3.5.1 ANIC (не связанные с конфигурацией Angie)

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>external-status-address</code>	Задаёт адрес, который будет отображаться в статусе ресурсов Ingress. Требуется аргумент командной строки <code>-report-status</code> . Имеет приоритет над аргументом <code>-external-service</code> .	Н/Д	<i>Отчет о состоянии Ingress</i>



### 3.5.2 Общая настройка

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>proxy-connect-timeout</code>	Задаёт значение директив <code>proxy_connect_timeout</code> и <code>grpc_connect_timeout</code> .	60s	
<code>proxy-read-timeout</code>	Задаёт значение директив <code>proxy_read_timeout</code> и <code>grpc_read_timeout</code> .	60s	
<code>proxy-send-timeout</code>	Задаёт значение директив <code>proxy_send_timeout</code> и <code>grpc_send_timeout</code> .	60s	
<code>client-max-body-size</code>	Задаёт значение директивы <code>client_max_body_size</code> .	1m	
<code>proxy-buffering</code>	Включает или отключает буферизацию ответов от проксируемого сервера.	True	
<code>proxy-buffers</code>	Задаёт значение директивы <code>proxy_buffers</code> .	Зависит от платформы.	
<code>proxy-buffer-size</code>	Задаёт значение директив <code>proxy_buffer_size</code> и <code>grpc_buffer_size</code> .	Зависит от платформы.	
<code>proxy-max-temp-file-size</code>	Задаёт значение директивы <code>proxy_max_temp_file_size</code> .	1024m	
<code>set-real-ip-from</code>	Задаёт значение директивы <code>set_real_ip_from</code> .	Н/Д	
<code>real-ip-header</code>	Задаёт значение директивы <code>real_ip_header</code> .	X-Real-IP	
<code>real-ip-recursive</code>	Включает или отключает директиву <code>real_ip_recursive</code> .	False	
<code>default-server-return</code>	Настраивает директиву <code>return</code> на сервере по умолчанию, которая обрабатывает клиентский запрос, если ни один из узлов ресурсов Ingress или VirtualServer не совпадает. Значение по умолчанию настраивает в Angie возврат страницы с ошибкой 404. Вы можете настроить фиксированный ответ или перенаправление. Например, значение <code>default-server-return: 302 https://mysite.ru</code> перенаправит клиент на <code>https://mysite.ru</code> .	404	
<code>server-tokens</code>	Включает или отключает директиву <code>server_tokens</code> .	True	
<code>worker-processes</code>	Задаёт значение директивы <code>worker_processes</code> .	auto	
<code>worker-rlimit-nofile</code>	Задаёт значение директивы <code>worker_rlimit_nofile</code> .	Н/Д	
<code>worker-connections</code>	Задаёт значение директивы <code>worker_connections</code> .	1024	
<code>worker-cpu-affinity</code>	Задаёт значение директивы <code>worker_cpu_affinity</code> .	Н/Д	
<code>worker-shutdown-timeout</code>	Задаёт значение директивы <code>worker_shutdown_timeout</code> .	Н/Д	
<code>server-names-hash-bucket-size</code>	Задаёт значение директивы <code>server_names_hash_bucket_size</code> .	256	
<code>server-names-hash-max-size</code>	Задаёт значение директивы <code>server_names_hash_max_size</code> .	1024	
<code>map-hash-bucket-size</code>	Задаёт значение директивы <code>map_hash_bucket_size</code> .	256	
<code>map-hash-max-size</code>	Задаёт значение директивы <code>map_hash_max_size</code> .	2048	
<code>resolver-addresses</code>	Задаёт значение адресов <code>resolver</code> .	Н/Д	
<code>resolver-ipv6</code>	Включает разрешение IPv6 в распознавателе.	True	
<code>resolver-timeout</code>	Задаёт значение <code>resolver_timeout</code> для разрешения имен.	30s	
<code>keepalive-timeout</code>	Задаёт значение директивы <code>keepalive_timeout</code> .	65s	
<code>keepalive-requests</code>	Задаёт значение директивы <code>keepalive_requests</code> .	100	
<code>variables-hash-bucket-size</code>	Задаёт значение директивы	256	

### 3.5.3 Ведение журнала

Ключ ConfigMap	Описание	По умолчанию	При-мер
<code>error-log-level</code>	Задаёт глобальный уровень журнала ошибок для Angie.	<code>notice</code>	
<code>access-log-off</code>	Отключает журнал доступа.	<code>False</code>	
<code>default-server-access-</code>	Отключает журнал доступа для сервера по умолчанию. Если журнал доступа отключен глобально ( <code>access-log-off: "True"</code> ), то журнал доступа к серверу по умолчанию всегда отключен.	<code>False</code>	
<code>log-format</code>	Задаёт настраиваемый формат журнала для HTTP- и HTTPS-трафика. Для удобства можно определить формат журнала в нескольких строках (строки разделяются символом <code>\n</code> ). В этом случае ANIC заменит каждый символ <code>\n</code> символом пробела. Все символы <code>'</code> должны быть экранированы.		
<code>log-format-escaping</code>	Задаёт экранирующие символы для переменных формата журнала. Поддерживаемые значения: <code>json</code> (экранирование JSON), <code>default</code> (экранирование по умолчанию), <code>none</code> (отключает экранирование).	<code>default</code>	
<code>stream-log-format</code>	Задаёт настраиваемый формат журнала для сквозного трафика TCP, UDP и TLS. Для удобства можно определить формат журнала в нескольких строках (строки разделяются символом <code>\n</code> ). В этом случае ANIC заменит каждый символ <code>\n</code> символом пробела. Все символы <code>'</code> должны быть экранированы.		
<code>stream-log-format-escape</code>	Задаёт экранирующие символы для переменных формата журнала потока. Поддерживаемые значения: <code>json</code> (экранирование JSON), <code>default</code> (экранирование по умолчанию), <code>none</code> (отключает экранирование).	<code>default</code>	

### 3.5.4 Манипулирование URI и заголовками запроса

Ключ ConfigMap	Описание	По умолчанию	При-мер
<code>proxy-hide-headers</code>	Задаёт значение одной директивы <code>proxy_hide_header</code> или нескольких.	Н/Д	<code>"angie. software/ proxy-hide-headers": "header-a, header-b"</code>
<code>proxy-pass-headers</code>	Задаёт значение одной директивы <code>proxy_pass_header</code> или нескольких.	Н/Д	<code>"angie. software/ proxy-pass-headers": "header-a, header-b"</code>



### 3.5.5 Аутентификация, SSL, TLS

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>redirect-to-https</code>	Задаёт правило перенаправления 301 на основе значения заголовка <code>http_x_forwarded_proto</code> в серверном блоке, требуя, чтобы входящий трафик шел по протоколу HTTPS. Полезно при терминировании SSL в системе балансировки нагрузки перед ANIC.	False	
<code>ssl-redirect</code>	Задаёт безусловное правило перенаправления 301 для всего входящего HTTP-трафика, требуя, чтобы входящий трафик шел по протоколу HTTPS.	True	
<code>hsts</code>	Включает режим HTTP Strict Transport Security (HSTS): заголовок HSTS добавляется к ответам от проксируемых серверов. Директива <code>preload</code> будет включена в заголовок.	False	
<code>hsts-max-age</code>	Задаёт значение директивы <code>max-age</code> заголовка HSTS.	2592000 (1 месяц)	
<code>hsts-include-subdomains</code>	Добавляет директиву <code>includeSubDomains</code> в заголовки HSTS.	False	
<code>hsts-behind-proxy</code>	Включает HSTS на основе значения заголовка запроса <code>http_x_forwarded_proto</code> . Следует использовать только в том случае, если в балансировщике нагрузки (прокси-сервере) перед ANIC настроено терминирование TLS.	False	
<div style="border: 1px solid #ccc; border-radius: 5px; padding: 10px; background-color: #e6f2ff;"> <p><b>Примечание</b></p> <p>Чтобы управлять перенаправлением с HTTP на HTTPS, настройте аннотацию <code>angie.software/redirect-to-https</code>.</p> </div>			
<code>ssl-protocols</code>	Задаёт значение директивы <code>ssl_protocols</code> .	TLSv1 TLSv1.1 TLSv1.2	
<code>ssl-prefer-server-ciphers</code>	Включает или отключает директиву <code>ssl_prefer_server_ciphers</code> .	On	
<code>ssl-ciphers</code>	Задаёт значение директивы <code>ssl_ciphers</code> .	HIGH:! aNULL:! MD5	
<code>ssl-dhparam-file</code>	Задаёт содержимое файла <code>dhparam</code> . Контроллер создаст файл и установит значение директивы <code>ssl_dhparam</code> с указанием пути к файлу.	Н/Д	
<code>ssl_session_timeout</code>	Задаёт время, в течение которого клиент может повторно использовать параметры сессии. См. также директиву <code>ssl_session_timeout</code> в документации Angie.	10m	
<code>ssl_session_cache</code>	Задаёт тип и размеры кэшей для хранения параметров сессий. См. также директиву <code>ssl_session_cache</code> в документации Angie.	shared:SSL:	
<code>ssl_session_tickets</code>	Разрешает или запрещает возобновление сессий при помощи TLS session tickets. См. также директиву <code>ssl_session_tickets</code> в документации Angie.	Off	
<code>ssl_stapling</code>	Разрешает или запрещает прикрепление OCSP-ответов сервером. См. также директиву <code>ssl_stapling</code> в документации Angie.	On	
<code>ssl_stapling_verify</code>	Разрешает или запрещает проверку сервером ответов OCSP. См. также директиву <code>ssl_stapling_verify</code> в документации Angie.	On	15

### 3.5.6 Прослушиватели

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>http2</code>	Включает HTTP/2 на серверах с включенным SSL.	<code>False</code>	
<code>proxy-protocol</code>	Включает прокси-протокол для входящих соединений.	<code>False</code>	

### 3.5.7 Бэкэнд-сервисы (апстримы)

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>lb-method</code>	Задаёт метод балансировки нагрузки. Чтобы использовать циклический метод, укажите <code>"round_robin"</code> .	<code>"random two least_conn"</code>	
<code>max-fails</code>	Задаёт значение параметра <code>max_fails</code> директивы <code>server</code> .	<code>1</code>	
<code>upstream-zone-size</code>	Задаёт размер зоны разделяемой памяти для апстримов.		
<code>fail-timeout</code>	Задаёт значение параметра <code>fail_timeout</code> директивы <code>server</code> .	<code>10s</code>	
<code>keepalive</code>	Задаёт значение директивы <code>keepalive</code> . Обратите внимание: если значение больше 0, в сгенерированную конфигурацию добавляется <code>proxy_set_header Connection ""</code> ;	<code>0</code>	

### 3.5.8 Фрагменты и пользовательские шаблоны

Ключ ConfigMap	Описание	По умолчанию	Пример
<code>main-snippets</code>	Задаёт пользовательский фрагмент в основном контексте.	Н/Д	
<code>http-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>http</code> .	Н/Д	
<code>location-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>location</code> .	Н/Д	
<code>server-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>server</code> .	Н/Д	
<code>stream-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>stream</code> .	Н/Д	
<code>main-template</code>	Задаёт основной шаблон конфигурации Angie.	По умолчанию шаблон считывается из файла в контейнере.	
<code>ingress-template</code>	Задаёт шаблон конфигурации Angie для ресурса <code>Ingress</code> .	По умолчанию шаблон считывается из файла в контейнере.	
<code>virtualserver-template</code>	Задаёт шаблон конфигурации Angie для ресурса <code>VirtualServer</code> .	По умолчанию шаблон считывается из файла в контейнере.	

## ГЛАВА 4

---

### GlobalConfiguration

---

Ресурс GlobalConfiguration позволяет вам определить глобальные параметры конфигурации ANIC. Он реализован как пользовательский ресурс.

Ресурс поддерживает настройку прослушивателей для балансировки нагрузки TCP и UDP. Прослушиватели требуются *ресурсам* *TransportServer*.

#### 4.1 Предварительные требования

При установке ANIC с манифестами необходимо указать ссылку на ресурс GlobalConfiguration в аргументе командной строки `-global-configuration`. Для ANIC требуется только один ресурс GlobalConfiguration.

#### 4.2 Спецификация GlobalConfiguration

Ресурс GlobalConfiguration определяет глобальные параметры конфигурации ANIC. Ниже приведен пример:

```
apiVersion: k8s.angie.software/v1alpha1
kind: GlobalConfiguration
metadata:
  name: angie-configuration
  namespace: angie-ingress
spec:
  listeners:
  - name: dns-udp
    port: 5353
    protocol: UDP
  - name: dns-tcp
    port: 5353
    protocol: TCP
```

Поле	Описание	Тип	Обязательно
<code>listeners</code>	Список прослушивателей.	<code>[[listener</code>	Нет

### 4.2.1 Прослушиватель

Прослушиватель определяет комбинацию протокола и порта, которые Angie будет использовать при приеме трафика для *TransportServer*:

```
name: dns-tcp
port: 5353
protocol: TCP
```

Поле	Описание	Тип	Обязательно
<code>name</code>	Имя прослушивателя. Это должна быть допустимая метка DNS, как определено в RFC 1035. Например, допустимы значения <code>hello</code> и <code>listener-123</code> . Имя должно быть уникальным среди всех прослушивателей. Имя <code>tls-passthrough</code> зарезервировано для встроенного прослушивателя TLS Passthrough и не может быть использовано.	<code>string</code>	Да
<code>port</code>	Порт прослушивателя. Порт должен находиться в диапазоне <code>1..65535</code> со следующими исключениями: <code>80</code> , <code>443</code> , порт <code>[статуса](/angie-ingress-controller/logging-and-monitoring/status-page)</code> . Комбинация порта и протокола должна быть уникальна среди всех прослушивателей.	<code>int</code>	Да
<code>protocol</code>	Протокол прослушивателя. Поддерживаемые значения: <code>TCP</code> и <code>UDP</code> .	<code>string</code>	Да

## 4.3 Использование GlobalConfiguration

Вы можете использовать обычные команды `kubectl` для работы с ресурсом `GlobalConfiguration`.

Например, следующая команда создает ресурс `GlobalConfiguration`, определенный в `global-configuration.yaml` с именем `angie-configuration`:

```
$ kubectl apply -f global-configuration.yaml

globalconfiguration.k8s.angie.software/angie-configuration created
```

Предполагая, что пространство имен ресурса называется `angie-ingress`, вы можете получить ресурс, запустив:

```
$ kubectl get globalconfiguration angie-configuration -n angie-ingress
```

NAME	AGE
angie-configuration	13s

В `kubectl get` и подобных командах также можно использовать короткое имя `gc` вместо `globalconfiguration`.

### 4.3.1 Валидация

Для ресурса `GlobalConfiguration` доступны два типа валидации:

- *Структурная валидация* с помощью `kubectl` и сервера Kubernetes API.
- *Всесторонняя валидация* с помощью ANIC.

#### Структурная валидация

Пользовательское определение ресурса для `GlobalConfiguration` включает структурную схему OpenAPI, которая описывает тип каждого поля ресурса.

Если вы попытаетесь создать (или обновить) ресурс с нарушением структурной схемы (например, используете строковое значение для поля порта прослушивателя), `kubectl` и сервер Kubernetes API отклонят такой ресурс:

- Пример проверки `kubectl`:

```
$ kubectl apply -f global-configuration.yaml

error: error validating "global-configuration.yaml": error validating
data: ValidationError(GlobalConfiguration.spec.listeners[0].port):
invalid type for
software.angie.k8s.v1alpha1.GlobalConfiguration.spec.listeners.port:
got "string", expected "integer"; if you choose to ignore these
errors, turn validation off with --validate=false
```

- Пример проверки сервера API Kubernetes:

```
$ kubectl apply -f global-configuration.yaml --validate=false

The GlobalConfiguration "angie-configuration" is invalid: []: Invalid
value: map[string]interface {}{ ... }: validation failure list:
spec.listeners.port in body must be of type integer: "string"
```

Если ресурс не отклонен (то есть не нарушает структурную схему), ANIC проверит его дополнительно.

#### Всесторонняя валидация

ANIC проверяет поля ресурса `GlobalConfiguration`. Если ресурс недопустим, ANIC не будет его использовать. Рассмотрим следующие два случая:

1. Если при запуске пода ANIC ресурс `GlobalConfiguration` недопустим, ANIC не сможет запуститься и завершит работу с ошибкой.
2. Если ресурс `GlobalConfiguration` становится недействительным, когда ANIC запущен, то ANIC проигнорирует новую версию. Он сообщит об ошибке и продолжит использовать предыдущую версию. Когда ресурс снова станет действительным, ANIC начнет его использовать.

**i Примечание**

Если ресурс GlobalConfiguration был удален во время работы ANIC, тот продолжит использовать предыдущую версию ресурса.

Вы можете проверить, успешно ли ANIC применил конфигурацию для GlobalConfiguration. Для нашего ресурса GlobalConfiguration `angie-configuration` мы можем запустить:

```
$ kubectl describe gc angie-configuration -n angie-ingress

. . .
Events:
  Type          Reason          Age   From                      Message
  ---          -
  Normal        Updated         11s   angie-ingress-controller  GlobalConfiguration
  angie-ingress/angie-configuration was updated
```

Обратите внимание, что раздел «События» (Events) включает событие Normal с причиной Updated, которое информирует нас о том, что конфигурация была успешно применена.

Если вы создадите недопустимый ресурс, ANIC отклонит его и выдаст событие Rejected. Например, если вы создадите ресурс GlobalConfiguration `angie-configuration` с несколькими прослушивателями, для которых задан один и тот же протокол UDP и порт 53, вы получите:

```
$ kubectl describe gc angie-configuration -n angie-ingress

. . .
Events:
  Type          Reason          Age   From                      Message
  ---          -
  Normal        Updated         55s   angie-ingress-controller  GlobalConfiguration
  angie-ingress/angie-configuration was updated

  Warning       Rejected         6s    angie-ingress-controller  GlobalConfiguration
  angie-ingress/angie-configuration is invalid and was rejected:
  spec.listeners: Duplicate value: "Duplicated port/protocol combination
  53/UDP"
```

Обратите внимание, что раздел "События" (Events) включает предупреждающее событие с указанием причины отклонения.

## ГЛАВА 5

---

### Policy

---

Ресурс Policy позволяет настраивать такие функции, как контроль доступа и ограничение скорости; их можно добавить к вашим ресурсам `VirtualServer` и `VirtualServerRoute`.

Он реализован как пользовательский ресурс.

Это справочная документация по ресурсу Policy.

### 5.1 Предварительные требования

Политики работают совместно с ресурсами `VirtualServer` и `VirtualServerRoute`, которые необходимо создавать отдельно.

### 5.2 Спецификация Policy

Ниже приведен пример политики, которая разрешает доступ клиентам из подсети `10.0.0.0/8` и запрещает доступ любым другим:

```
apiVersion: k8s.angie.software/v1
kind: Policy
metadata:
  name: allow-localhost
spec:
  accessControl:
    allow:
      - 10.0.0.0/8
```

Поле	Описание	Тип	Обязательно
<code>AccessControl</code>	Политика контроля доступа, основанная на IP-адресе клиента.	<i>AccessControl</i>	Нет
<code>ingressClassName</code>	Указывает, какой экземпляр ANIC должен обрабатывать ресурс Policy.	<code>string</code>	Нет
<code>rateLimit</code>	Политика ограничения скорости управляет скоростью обработки запросов по определенному ключу.	<i>RateLimit</i>	Нет
<code>basicAuth</code>	Политика базовой аутентификации настраивает в Angie аутентификацию клиентских запросов с использованием базовой аутентификации HTTP по учетным данным.	<i>BasicAuth</i>	Нет
<code>ingressMTLS</code>	Политика IngressMTLS настраивает проверку сертификата клиента.	<i>IngressMTLS</i>	Нет
<code>egressMTLS</code>	Политика EgressMTLS настраивает аутентификацию и проверку сертификата апстрима.	<i>EgressMTLS</i>	Нет
<code>OIDC</code>	Политика OIDC настраивает аутентификацию через провайдера OIDC.	<i>OIDC</i>	Нет
<code>JWT</code>	Политика JWT настраивает Angie для аутентификации запросов клиентов с использованием JSON Web Tokens.	<i>JWT</i>	Нет

#### Примечание

Политика должна включать в себя ровно одно значение.

### 5.2.1 AccessControl

Политика контроля доступа настраивает в Angie отклонение или принятие запросов от клиентов с указанными IP-адресами и подсетями.

Например, следующая политика разрешает доступ клиентам из подсети 10.0.0.0/8 и запрещает доступ любым другим:

```
accessControl:
  allow:
  - 10.0.0.0/8
```

Напротив, приведенная ниже политика делает обратное: запрещает доступ клиентам с 10.0.0.0/8 и разрешает доступ любым другим клиентам:

```
accessControl:
  deny:
  - 10.0.0.0/8
```

#### Примечание

Функция реализована с использованием модуля Angie `http_access`. Политика контроля доступа ANIC поддерживает либо разрешающие, либо запрещающие правила, но не оба вида сразу (в отличие от модуля).

По-ле	Описание	Тип	Обяза-тельно
allow	Разрешает доступ для указанных сетей или адресов. Например, 192.168.1.1 или 10.1.1.0/16.	[]string	Нет
deny	Запрещает доступ для указанных сетей или адресов. Например, 192.168.1.1 или 10.1.1.0/16.	[]string	Нет

AccessControl должен включать либо `allow`, либо `deny`.

### Поведение слияния AccessControl

Ресурс `VirtualServer` или `VirtualServerRoute` может ссылаться на несколько политик контроля доступа. Например, здесь мы ссылаемся на две политики, в каждой из которых настроен список разрешений:

```
policies:
- name: allow-policy-one
- name: allow-policy-two
```

Когда вы ссылаетесь на несколько политик контроля доступа, ANIC объединит их содержимое в один список разрешений или запретов.

Ссылки как на разрешающие, так и на запрещающие политики, как показано в примере ниже, не поддерживаются. Если указаны ссылки как на разрешающие, так и на запрещающие списки, ANIC использует только политики разрешающих списков.

```
policies:
- name: deny-policy
- name: allow-policy-one
- name: allow-policy-two
```

### RateLimit

Политика ограничения скорости настраивает в Angie ограничение скорости обработки запросов.

Например, следующая политика ограничит все последующие запросы, поступающие с одного IP-адреса, при превышении скорости в 10 запросов в секунду:

```
rateLimit:
rate: 10r/s
zoneSize: 10M
key: ${binary_remote_addr}
```

#### Примечание

Функция реализована с использованием модуля Angie `http_limit_req_module`.

Поле	Описание	Тип	Обязательно
<code>rate</code>	Допустимая скорость запросов. Скорость указывается в запросах в секунду (r/s) или запросах в минуту (r/m).	<code>string</code>	Да
<code>key</code>	Ключ, к которому применяется ограничение скорости. Может содержать текст, переменные или их комбинацию. Переменные должны заключены в <code>\${}</code> . Например: <code>binary_remote_addr</code> . Допустимые переменные: <code>binary_remote_addr</code> , <code>request_uri</code> , <code>url</code> , <code>http_</code> , <code>args</code> , <code>arg_</code> , <code>cookie_</code> .	<code>string</code>	Да
<code>zoneSize</code>	Размер зоны разделяемой памяти. Допускаются только положительные значения. Допустимые суффиксы - <code>k</code> или <code>m</code> ; если суффикс не задан, предполагается <code>k</code> .	<code>string</code>	Да
<code>delay</code>	Указывает предел, при достижении которого избыточные запросы становятся отложенными. Если этот параметр не задан, задерживаются все избыточные запросы.	<code>int</code>	Нет
<code>noDelay</code>	Отключает задержку избыточных запросов при ограничении количества запросов. Имеет приоритет над <code>delay</code> , если заданы оба параметра.	<code>bool</code>	Нет
<code>burst</code>	Избыточные запросы задерживаются до тех пор, пока их количество не превысит размер <code>burst</code> , после чего запрос завершается с ошибкой.	<code>int</code>	Нет
<code>dryRun</code>	Включает режим сухого прогона. В этом режиме ограничение скорости фактически не применяется, но количество избыточных запросов учитывается, как обычно, в зоне разделяемой памяти.	<code>bool</code>	Нет
<code>logLevel</code>	Устанавливает желаемый уровень ведения журнала для случаев, когда сервер отказывается обрабатывать запросы из-за превышения скорости или задерживает обработку запросов. Допустимые значения: <code>info</code> , <code>notice</code> , <code>warn</code> или <code>error</code> . Значение по умолчанию - <code>error</code> .	<code>string</code>	Нет
<code>rejectCode</code>	Задаёт код состояния, возвращаемый в ответ на отклоненные запросы. Значение должно попадать в диапазон <code>400..599</code> . Значение по умолчанию - <code>503</code> .	<code>int</code>	Нет

Для каждой политики, на которую ссылается `VirtualServer` или его `VirtualServerRoute`, ANIC сгенерирует единую зону ограничения скорости, определенную директивой `http_limit_req`. Если два ресурса `VirtualServer` ссылаются на одну и ту же политику, ANIC сгенерирует две разные зоны ограничения скорости, по одной на каждый `VirtualServer`.

## Поведение слияния RateLimit

Ресурс VirtualServer или VirtualServerRoute может ссылаться на несколько политик ограничения скорости. Например, здесь мы ссылаемся на две политики:

```

policies:
- name: rate-limit-policy-one
- name: rate-limit-policy-two
    
```

Когда вы ссылаетесь на несколько политик ограничения скорости, ANIC настроит в Angie использование всех указанных ограничений скорости. Если определено несколько политик, каждая дополнительная политика наследует параметры `dryRun`, `LogLevel` и `rejectCode` из первой политики, на которую идет ссылка (`rate-limit-policy-one` в примере выше).

### 5.2.2 BasicAuth

Настраивает в Angie аутентификацию клиентских запросов при помощи базовой схемы аутентификации HTTP.

Например, следующая политика будет отклонять все запросы, которые не содержат действительную комбинацию имени пользователя и пароля в заголовке HTTP Authentication

```

basicAuth:
  secret: htpasswd-secret
  realm: "My API"
    
```

#### Примечание

Функция реализована с использованием модуля Angie `http_auth_basic`.

Поле	Описание	Тип	Обязательно
<code>secret</code>	Имя секрета Kubernetes, в котором хранится конфигурация Htpasswd. Он должен находиться в том же пространстве имен, что и ресурс Policy. Секрет должен иметь тип <code>angie.software/htpasswd</code> , а конфигурация должна храниться в секрете по ключу <code>htpasswd</code> ; в противном случае секрет будет отклонен как недействительный.	<code>string</code>	Да
<code>realm</code>	Область для базовой аутентификации.	<code>string</code>	Нет

## Поведение слияния BasicAuth

Ресурс VirtualServer или VirtualServerRoute может ссылаться на несколько политик базовой аутентификации. При этом будет применяться только одна из них. Все последующие ссылки будут проигнорированы. Например, здесь мы ссылаемся на две политики:

```
policies:
- name: basic-auth-policy-one
- name: basic-auth-policy-two
```

В этом примере ANIC будет использовать конфигурацию из первой ссылки на политику «basic-auth-policy-one» и игнорирует «basic-auth-policy-two».

### 5.2.3 IngressMTLS

Политика IngressMTLS настраивает проверку сертификата клиента.

Например, следующая политика будет проверять сертификат клиента, используя сертификат Центра Сертификации, указанный в `ingress-mtls-secret`:

```
ingressMTLS:
  clientCertSecret: ingress-mtls-secret
  verifyClient: "on"
  verifyDepth: 1
```

Ниже приведен пример `ingress-mtls-secret` типа `angie.software/ca`

```
kind: Secret
metadata:
  name: ingress-mtls-secret
apiVersion: v1
type: angie.software/ca
data:
  ca.crt: <base64encoded-certificate>
```

У ресурса VirtualServer, который ссылается на политику IngressMTLS, должно быть следующие настройки:

- включено завершение TLS.
- ссылка на политику в спецификации VirtualServer. Не разрешается ссылаться на политику IngressMTLS в маршруте или вложенном маршруте VirtualServerRoute.

Если эти условия нарушены, Angie будет отправлять клиентам код состояния 500.

Вы можете передавать сведения о сертификате клиента, включая сам сертификат, серверам апстрима. Например:

```
action:
  proxy:
    upstream: webapp
    requestHeaders:
      set:
        - name: client-cert-subj-dn
          value: ${ssl_client_s_dn} # subject DN
        - name: client-cert
          value: ${ssl_client_escaped_cert} # клиентский сертификат в формате PEM
→ (urlencoded)
```

Мы используем параметр `requestHeaders` в `Action.Proxy` для задания значений двух заголовков, которые Angie будет передавать серверам апстрима. См. список встроенных переменных,

поддерживаемых модулем `http_ssl`, которые вы можете использовать для передачи сведений о сертификате клиента.

#### Примечание

Функция реализована с использованием модуля Angie `http_ssl`.

## Использование списка отзыва сертификатов

Политика IngressMTLS поддерживает настройку списка CRL для политики. Это можно сделать одним из двух способов.

#### Примечание

Одновременно можно использовать только один из этих параметров конфигурации.

1. Добавление в тип секрета `angie.software/ca` поля `ca.crl`, которое содержит список отзыва сертификатов в кодировке base64. Пример YAML:

```
kind: Secret
metadata:
  name: ingress-mtls-secret
apiVersion: v1
type: AngieSoftware/ca
data:
  ca.crt: <base64encoded-certificate>
  ca.crl: <base64encoded-crl>
```

2. Добавление поля `crlFileName` с именем CRL-файла в спецификацию политики IngressMTLS.

#### Примечание

Этот параметр конфигурации следует использовать только при наличии CRL-файла размером более 1 МБ; в противном случае рекомендуется использовать для управления CRL тип секрета `angie.software/ca`.

Пример YAML:

```
apiVersion: k8s.AngieSoftware/v1
kind: Policy
metadata:
  name: ingress-mtls-policy
spec:
  ingressMTLS:
    clientCertSecret: ingress-mtls-secret
    crlFileName: webapp.crl
    verifyClient: "on"
    verifyDepth: 1
```

#### Предупреждение

При настройке CRL с помощью поля `ingressMTLS.crlFileName` следует учитывать дополнительный контекст:

1. ANIC ожидает, что CRL, в данном случае `webapp.crl`, будет находиться в каталоге `/etc/angie/secrets`. Для развертывания ANIC необходимо будет добавить точку подключения тома. Добавьте свой CRL в каталог `/etc/angie/secrets`.
2. При обновлении содержимого списка CRL (например, был отозван новый сертификат) Angie необходимо перезагрузить, чтобы отразились последние изменения. В зависимости от вашей среды для этого может потребоваться обновить имя списка CRL и применить это обновление к политике `ingress-mtls.yaml`, чтобы Angie получил последнюю версию CRL.

Обратитесь к документации Kubernetes по [томам](#), чтобы найти наилучшую реализацию для вашей среды.

Поле	Описание	Тип	Обязательно
<code>clientCertSecret</code>	Имя секрета Kubernetes, в котором хранится сертификат центра сертификации. Он должен находиться в том же пространстве имен, что и ресурс Policy. Секрет должен иметь тип <code>angie.software/ca</code> , а конфигурация должна храниться в секрете по ключу <code>ca.crt</code> ; в противном случае секрет будет отклонен как недействительный.	<code>string</code>	Да
<code>verifyClient</code>	Верификация для клиента. Допустимые значения: <code>"on"</code> , <code>"off"</code> , <code>"optional"</code> , <code>"optional_no_ca"</code> . Значение по умолчанию - <code>"on"</code> .	<code>string</code>	Нет
<code>verifyDepth</code>	Устанавливает глубину проверки в цепочке клиентских сертификатов. Значение по умолчанию равно 1.	<code>int</code>	Нет
<code>crlFileName</code>	Имя файла списка отзыва сертификатов. ANIC будет искать этот файл в каталоге <code>/etc/angie/secrets</code>	<code>string</code>	Нет

## Поведение слияния IngressMTLS

Ресурс `VirtualServer` может ссылаться только на одну политику `IngressMTLS`. Все последующие ссылки будут проигнорированы. Например, здесь мы ссылаемся на две политики:

```
policies:
- name: ingress-mtls-policy-one
- name: ingress-mtls-policy-two
```

В этом примере ANIC будет использовать конфигурацию из первой ссылки на политику `ingress-mtls-policy-one` и игнорирует `ingress-mtls-policy-two`.

## 5.2.4 EgressMTLS

EgressMTLS настраивает аутентификацию и проверку сертификатов для апстримов.

Например, следующая политика будет использовать `egress-mtls-secret` для аутентификации в приложении апстрима и `egress-trusted-ca-secret` для проверки сертификата приложения:

```
egressMTLS:
  tlsSecret: egress-mtls-secret
  trustedCertSecret: egress-trusted-ca-secret
  verifyServer: on
  verifyDepth: 2
```

### Примечание

Функция реализована с использованием модуля Angie `http_proxy`.

Поле	Описание	Тип	Обязательно
<code>tlsSecret</code>	Имя секрета файла Kubernetes, в котором хранятся сертификат и ключ TLS. Он должен находиться в том же пространстве имен, что и ресурс Policy. Секрет должен иметь тип <code>kubernetes.io/tls</code> , сертификат - храниться в секрете под ключом <code>tls.crt</code> , а ключ - как <code>tls.key</code> ; в противном случае секрет будет отклонен как недействительный.	<code>string</code>	Нет
<code>trustedCertSecret</code>	Имя секрета Kubernetes, в котором хранится сертификат центра сертификации. Он должен находиться в том же пространстве имен, что и ресурс Policy. Секрет должен иметь тип <code>angie.software/ca</code> , а конфигурация должна храниться в секрете по ключу <code>ca.crt</code> ; в противном случае секрет будет отклонен как недействительный.	<code>string</code>	Нет
<code>verifyServer</code>	Включает проверку сертификата HTTPS-сервера апстрима.	<code>bool</code>	Нет
<code>verifyDepth</code>	Устанавливает глубину проверки в цепочке сертификатов проксируемого HTTPS-сервера. Значение по умолчанию равно 1.	<code>int</code>	Нет
<code>sessionReuse</code>	Позволяет повторно использовать SSL-сеансы к апстримам. Значение по умолчанию равно <code>true</code> .	<code>bool</code>	Нет
<code>serverName</code>	Позволяет передавать имя сервера через расширение SNI.	<code>bool</code>	Нет
<code>sslName</code>	Позволяет переопределить имя сервера, используемое для проверки сертификата HTTPS-сервера апстрима.	<code>string</code>	Нет
<code>ciphers</code>	Указывает разрешенные шифры для запросов к HTTPS-серверу апстрима. Значение по умолчанию - <code>DEFAULT</code> .	<code>string</code>	Нет
<code>protocols</code>	Задаёт протоколы для запросов к HTTPS-серверу апстрима. Значение по умолчанию - <code>TLSv1, TLSv1.1, TLSv1.2</code> .	<code>string</code>	Нет

## Поведение слияния EgressMTLS

Ресурс VirtualServer или VirtualServerRoute может ссылаться на несколько политик EgressMTLS. При этом будет применяться только одна из них. Все последующие ссылки будут проигнорированы. Например, здесь мы ссылаемся на две политики:

```
policies:
- name: egress-mtls-policy-one
- name: egress-mtls-policy-two
```

В этом примере ANIC будет использовать конфигурацию из первой ссылки на политику egress-mtls-policy-one и игнорирует egress-mtls-policy-two.

### 5.2.5 OIDC

OIDC (OpenID Connect) обеспечивает удобную аутентификацию пользователей через внешнего провайдера, используя безопасные токены для управления доступом в системе.

#### Примечание

Эта функция отключена по умолчанию. Чтобы ее включить, задайте аргумент командной строки `enable-oidc=true`.

Политика OIDC настраивает ANIC как клиент (relying party) для аутентификации через OpenID Connect:

```
policies:
- name: oidc-policy
```

Например, следующая конфигурация использует clientID `myclient` и clientSecret `oidc-secret` для аутентификации через провайдера OpenID Connect `https://idp.example.com`:

```
oidc:
clientID: myclient
clientSecret: oidc-secret
authEndpoint: https://idp.example.com/openid-connect/auth
jwksURI: https://idp.example.com/openid-connect/certs
tokenEndpoint: https://idp.example.com/openid-connect/token
scope: openid+profile+email
accessTokenEnable: true
```

Обязательные условия:

- В спецификации VirtualServer задайте обязательные переменные `$jwt_claim_iat`, `$jwt_claim_iss`, `$jwt_claim_sub`, `$jwt_claim_aud` в `maps`. Переменные обеспечивают валидацию токенов в процессе аутентификации OIDC.
- Добавьте секрет с ключом клиента. Ключ должен быть закодирован в Base64:

```
apiVersion: v1
kind: Secret
metadata:
  name: oidc-secret
type: angie.software/oidc
data:
  client-secret: <client_secret>
```

Поле	Описание	Тип	Обязательно
<code>clientID</code>	Идентификатор клиента, предоставленный провайдером OIDC. Идентифицирует приложение, которое обращается за авторизацией.	<code>string</code>	да
<code>clientSecret</code>	Имя секрета, где хранится ключ клиента. Должен находиться в том же пространстве имен, что и <code>Policy</code> .	<code>string</code>	да
<code>authEndpoint</code>	URL конечной точки авторизации, предоставленной провайдером OIDC. Это адрес, по которому будут отправляться запросы для аутентификации пользователя.	<code>string</code>	да
<code>jwtKeyURI</code>	URI, по которому провайдер OIDC предоставляет сертификаты (JSON Web Key Set) для проверки выданных сервером JWT-токенов (JSON Web Token).	<code>string</code>	да
<code>tokenEndpoint</code>	URL для получения токенов аутентификации и обновления от провайдера OIDC.	<code>string</code>	да
<code>scope</code>	Список OIDC-областей, которые нужно запросить у провайдера. Значение по умолчанию - <code>openid</code> . Это значение является обязательным. Вы также можете добавлять другие области, используя знак <code>+</code> , например: <code>openid+profile+email</code> .	<code>string</code>	да
<code>accessToken</code>	Включает использование Bearer-токена для авторизации доступа к защищенным ресурсам на проксируемом сервере.	<code>bool</code>	нет

#### Примечание

Одновременно можно ссылаться только на одну политику OIDC в ресурсах `VirtualServer`. При этом одна политика может применяться к разным маршрутам ресурсов `VirtualServer`.

### Поведение слияния OIDC

Ресурс `VirtualServer` может ссылаться на несколько политик OIDC. При этом будет применяться только одна из них. Все последующие ссылки будут проигнорированы. Например, здесь мы ссылаемся на две политики:

```
policies:
- name: oidc-policy-one
- name: oidc-policy-two
```

В этом примере ANIC будет использовать конфигурацию из первой ссылки на политику `oidc-policy-one` и игнорирует `oidc-policy-two`.

### 5.2.6 JWT

Политика JWT настраивает Angie для аутентификации запросов клиентов с использованием JSON Web Tokens.

```
policies:
- name: jwt-policy
```

#### Примечание

Эта функция отключена по умолчанию. Чтобы ее включить, задайте аргумент командной строки `-enable-jwt=true`.

Следующая политика будет отклонять все запросы, которые не содержат действительный JWT в токене заголовка HTTP:

```
jwt:
  realm: MyProductAPI
  secret: jwk-secret
  token: $http_token
```

Обязательные условия:

- Добавьте секрет с ключом клиента. Ключ должен быть закодирован в Base64:

```
apiVersion: v1
kind: Secret
metadata:
  name: jwk-secret
type: angie.software/jwk
data:
  jwk: <client_secret>
```

Поле	Описание	Тип	Обязательно
realm	Область, которую клиент увидит при запросе аутентификации, например, строка, описывающая защищенный ресурс.	string	да
secret	Имя секрета Kubernetes, который хранит JSON Web Key (JWK), используемый для проверки подписей токенов JWT. Angie будет использовать ключи из этого секрета для валидации подписей JWT и проверки их подлинности. Секрет должен находиться в том же пространстве имен, что и Policy.	string	да
token	Указывает, откуда нужно извлечь JWT. Например, переменная \$http_token ссылается на значение заголовка HTTP token, который будет передан клиентом.	string	нет

### Поведение слияния JWT

Ресурс VirtualServer может ссылаться на несколько политик JWT. При этом будет применяться только одна из них. Все последующие ссылки будут проигнорированы. Например, здесь мы ссылаемся на две политики:

```
policies:
- name: jwt-policy-one
- name: jwt-policy-two
```

В этом примере ANIC будет использовать конфигурацию из первой ссылки на политику jwt-policy-one и игнорирует jwt-policy-two.

## 5.2.7 Применение политик

Политики можно применять как к ресурсам VirtualServer, так и к VirtualServerRoute. Например:

```
- VirtualServer:
  apiVersion: k8s.angie.software/v1
  kind: VirtualServer
  metadata:
    name: cafe
    namespace: cafe
  spec:
    host: cafe.example.com
    tls:
      secret: cafe-secret
    policies: # spec policies
      - name: policy1
  upstreams:
    - name: coffee
      service: coffee-svc
      port: 80
  routes:
    - path: /tea
      policies: # route policies
      - name: policy2
        namespace: cafe
        route: tea/tea
    - path: /coffee
      policies: # route policies
      - name: policy3
        namespace: cafe
    action:
      pass: coffee
```

В случае VirtualServer политику можно применить:

- для всех маршрутов (политики спецификации)
- к определенному маршруту (политики маршрутов)

Политики маршрутов имеют приоритет над политиками спецификации *того же типа*. Если в примере выше тип политик policy-1 и policy-3 - AccessControl, то для запросов к cafe.example.com/coffee Angie применит policy-3.

Переопределение обеспечивается Angie: политики спецификации реализуются в контексте конфигурации server, а политики маршрутов реализуются в контексте location. В результате приоритет в рамках одного типа имеют политики маршрутов.

- Ресурс VirtualServerRoute, на который ссылается указанный выше VirtualServer:

```
apiVersion: k8s.angie.software/v1
kind: VirtualServerRoute
metadata:
  name: tea
  namespace: tea
spec:
  host: cafe.example.com
  upstreams:
    - name: tea
      service: tea-svc
      port: 80
```

```
subroutes: # subroute policies
- path: /tea
  policies:
- name: policy4
  namespace: tea
  action:
  pass: tea
```

В `VirtualServerRoute` можно применить политику к вложенному маршруту (политики вложенных маршрутов).

Политики вложенных маршрутов имеют приоритет над политиками спецификации того же типа. В приведенном выше примере, если тип политик `policy-1` (в `VirtualServer`) и `policy-4` - `AccessControl`, то для запросов к `cafe.example.com/tea` Angie будет применять `policy-4`. Как и в случае с `VirtualServer`, переопределение обеспечивается средствами Angie.

Политики вложенных маршрутов всегда имеют приоритет над политиками маршрутов независимо от типа. Например, политика `policy-2` в маршруте `VirtualServer` будет проигнорирована на вложенном маршруте `/tea`, поскольку у того есть свои собственные политики (в нашем случае это только `policy4`). Если бы у вложенного маршрута не было политик, то была бы применена `policy-2`. Это переопределение выполняет ANIC - контекст `location` для вложенного маршрута будет содержать либо политики маршрута, либо политики вложенного маршрута, но не то и другое вместе.

## 5.2.8 Недопустимые политики

Angie будет рассматривать политику как недействительную, если выполняется одно из следующих условий:

- Политика не проходит *всестороннюю валидацию*.
- Политика отсутствует в кластере.
- Политика не соответствует требованиям, предъявляемым к ее конкретному типу. Например, политика `ingressMTLS` требует, чтобы в `VirtualServer` было включено завершение TLS.

В случае недопустимой политики Angie возвращает код состояния 500 для клиентских запросов со следующими правилами:

- Если на политику ссылается маршрут `VirtualServer` или вложенный маршрут `VirtualServerRoute`, Angie будет возвращать код состояния 500 для запросов к URI того маршрута.
- Если ссылка на политику задана в спецификации `VirtualServer`, Angie будет возвращать код состояния 500 для запросов ко всем URI этого `VirtualServer`.

Если политика недействительна, `VirtualServer` или `VirtualServerRoute` будет иметь *статус с предупреждением* о состоянии и сообщением, объясняющим, почему политика не была признана недействительной.

## 5.2.9 Валидация

Для ресурса Policy доступны два типа валидации:

- *Структурная валидация* с помощью `kubectl` и сервера Kubernetes API.
- *Всесторонняя валидация* с помощью ANIC.

### Структурная валидация

Пользовательское определение ресурса для Policy включает структурную схему OpenAPI, которая описывает тип каждого поля ресурса.

Если вы попытаетесь создать (или обновить) ресурс, который нарушает структурную схему (например, использует строковое значение вместо массива строк в поле `allow`), то `kubectl` и сервер Kubernetes API отклонят ресурс.

- Пример проверки `kubectl`:

```
kubectl apply -f access-control-policy-allow.yaml

error: error validating "access-control-policy-allow.yaml": error validating
↳data: ValidationError(Policy.spec.accessControl.allow): invalid type for
↳software.angie.k8s.v1.Policy.spec.accessControl.allow: got "string", expected
↳"array"; if you choose to ignore these errors, turn validation off with --
↳validate=false
```

- Пример проверки сервера Kubernetes API:

```
kubectl apply -f access-control-policy-allow.yaml --validate=false

The Policy "webapp-policy" is invalid: spec.accessControl.allow: Invalid value:
↳"string": spec.accessControl.allow in body must be of type array: "string"
```

Если ресурс прошел структурную валидацию, выполняется всесторонняя валидация ANIC.

### Всесторонняя валидация

ANIC проверяет поля ресурса Policy. Если ресурс недопустим, ANIC отклонит его. Ресурс останется в кластере, но ANIC будет игнорировать его.

Можно использовать `kubectl`, чтобы проверить, успешно ли ANIC применил конфигурацию Policy. Для политики `mypolicy` мы можем запустить:

```
kubectl describe pol mypolicy

. . .
Events:
  Type          Reason          Age   From                      Message
  ----
Normal         AddedOrUpdated  11s   angie-ingress-controller  Policy default/mypolicy was
↳added or updated
```

Обратите внимание, что раздел «События» (Events) включает событие Normal с причиной AddedOrUpdated, которое информирует нас о том, что конфигурация была успешно применена.

Если вы создадите недопустимый ресурс, ANIC отклонит его и выдаст событие Rejected. Например, если вы создадите политику `mypolicy` с недопустимым IP-адресом `10.0.0.` в поле `allow`, то вы получите:

```
kubectl describe policy mypolicy
. . .
Events:
  Type          Reason      Age   From              Message
  ----
Warning        Rejected    7s    angie-ingress-controller Policy default/mypolicy is invalid
↳and was rejected: spec.accessControl.allow[0]: Invalid value: "10.0.0.": must be a
↳CIDR or IP
```

Обратите внимание, что раздел «События» (Events) включает предупреждающее событие с указанием причины отклонения.

Кроме того, эта информация также доступна в поле `status` ресурса Policy. Обратите внимание на раздел «Статус» (Status) политики:

```
kubectl describe pol mypolicy
. . .
Status:
  Message: Policy default/mypolicy is invalid and was rejected: spec.accessControl.
↳allow[0]: Invalid value: "10.0.0.": must be a CIDR or IP
  Reason:   Rejected
  State:    Invalid
```

**Примечание**

Если вы сделаете существующий ресурс недействительным, ANIC отклонит его.

## ГЛАВА 6

---

### TransportServer

---

Ресурс TransportServer позволяет настраивать балансировку нагрузки по протоколам TCP, UDP и TLS Passthrough. Он реализован как пользовательский ресурс.

Это справочная документация по ресурсу TransportServer.

### 6.1 Предварительные требования

- Для TCP и UDP ресурс TransportServer должен использоваться совместно с ресурсом GlobalConfiguration, который должен быть создан отдельно.
- Для TLS Passthrough обязательно включите параметр командной строки `-enable-tls-passthrough` в ANIC.

### 6.2 Спецификация TransportServer

Ресурс TransportServer определяет конфигурацию балансировки нагрузки для трафика TCP, UDP или TLS Passthrough. Ниже приведено несколько примеров:

- Балансировка нагрузки TCP:

```
apiVersion: k8s.angie.software/v1alpha1
kind: TransportServer
metadata:
  name: dns-tcp
spec:
  listener:
    name: dns-tcp
    protocol: TCP
  tls:
    secret: cafe-secret
  upstreams:
  - name: dns-app
    service: dns-service
    port: 5353
```

```
action:
pass: dns-app
```

- Балансировка нагрузки UDP:

```
apiVersion: k8s.angie.software/v1alpha1
kind: TransportServer
metadata:
name: dns-udp
spec:
listener:
  name: dns-udp
  protocol: UDP
upstreams:
- name: dns-app
  service: dns-service
  port: 5353
  upstreamParameters:
    udpRequests: 1
    udpResponses: 1
    action:
    pass: dns-app
```

- Балансировка нагрузки TLS Passthrough:

```
apiVersion: k8s.angie.software/v1alpha1
kind: TransportServer
metadata:
name: secure-app
spec:
listener:
  name: tls-passthrough
  protocol: TLS_PASSTHROUGH
host: app.example.com
upstreams:
- name: secure-app
  service: secure-app
  port: 8443
  action:
  pass: secure-app
```

Поле	Описание	Тип	Обязательно
<code>listener</code>	Прослушиватель, через который Angie будет принимать входящие соединения и датаграммы.	<i>Listener</i>	Да
<code>host</code>	Хост (доменное имя) сервера. Это должен быть допустимый поддомен, как определено в RFC 1123, например <code>my-app</code> или <code>hello.example.com</code> . Домены с подстановочными знаками, такие как <code>*.example.com</code> , не допускаются. Требуется для балансировки нагрузки TLS Passthrough.	<i>строка</i>	Нет
<code>tls</code>	Конфигурация завершения TLS. Не поддерживается для балансировки нагрузки TLS Passthrough.	<i>TLS</i>	Нет
<code>upstreams</code>	Список апстримов.	<i>//upstream</i>	Да
<code>upstreamParameters</code>	Параметры апстрима.	<i>UpstreamParameters</i>	Нет
<code>action</code>	Действие, выполняемое для клиентского соединения или датаграммы.	<i>Action</i>	Да
<code>ingressClassName</code>	Указывает, какой экземпляр ANIC должен обрабатывать ресурс TransportServer.	<code>string</code>	Нет
<code>streamSnippets</code>	Задаёт пользовательский фрагмент в контексте <code>stream</code> .	<code>string</code>	Нет
<code>serverSnippets</code>	Задаёт пользовательский фрагмент в контексте <code>server</code> .	<code>string</code>	Нет

### 6.2.1 Listener

Ссылается на прослушиватель, через который Angie будет принимать входящий трафик к TransportServer. Для TCP и UDP прослушиватель должен быть определен в ресурсе GlobalConfiguration. При ссылке на прослушиватель должны совпадать как имя, так и протокол. Для TLS Passthrough используйте встроенный прослушиватель с именем `tls-passthrough` и протоколом `TLS_PASSTHROUGH`.

Пример:

```
listener:
  name: dns-udp
  protocol: UDP
```

Поле	Описание	Тип	Обязательно
<code>name</code>	Имя прослушивателя.	<code>string</code>	Да
<code>protocol</code>	Протокол прослушивателя.	<code>string</code>	Да

## 6.2.2 TLS

Поле `tls` определяет конфигурацию TLS для `TransportServer`. Обратите внимание, что текущая реализация поддерживает завершение TLS на нескольких портах, где каждому приложению принадлежит выделенный порт. При этом ANIC завершает TLS-соединения на каждом порту, где каждое приложение использует свой собственный сертификат или ключ, и направляет соединения соответствующему приложению (сервису) на основе этого входящего порта (т. е. любое TLS-соединение независимо от настроек SNI на порту будет перенаправлено в приложение, соответствующее этому порту). Пример конфигурации показан ниже:

```
secret: cafe-secret
```

Поле	Описание	Тип	Обязательно
<code>secret</code>	Имя секрета с сертификатом TLS и ключом. Секрет должен принадлежать тому же пространству имен, что и транспортный сервер. Секрет должен иметь тип <code>kubernetes.io/tls</code> и содержать ключи с именами <code>tls.crt</code> и <code>tls.key</code> , содержащие сертификат и закрытый ключ, как описано <a href="#">здесь</a> .	<code>string</code>	Нет

## 6.2.3 Upstream

Определяет конечное место назначения для `TransportServer`. Например:

```
name: secure-app
service: secure-app
port: 8443
maxFails: 3
maxConns: 100
failTimeout: 30s
loadBalancingMethod: least_conn
```

Поле	Описание	Тип	Обязательно
<code>name</code>	Имя апстрима. Это должна быть допустимая метка DNS, как определено в RFC 1035. Например, допустимы значения <code>hello</code> и <code>upstream-123</code> . Имя должно быть уникальным среди всех апстримов ресурса.	<code>string</code>	Да
<code>service</code>	Название сервиса. Сервис должен принадлежать к тому же пространству имен, что и ресурс. Если сервиса не существует, Angie предположит, что у него нет конечных точек, и будет закрывать клиентские соединения и игнорировать датаграммы.	<code>string</code>	Да
<code>port</code>	Порт службы. Если у сервиса этот порт не задан, Angie предположит, что у него нет конечных точек, и будет закрывать клиентские соединения и игнорировать датаграммы. Значение должно находиться в диапазоне <code>1..65535</code> .	<code>int</code>	Да
<code>maxFails</code>	Задаёт число неудачных попыток установить связь с сервером, которые должны произойти в течение времени, заданного параметром <code>failTimeout</code> , чтобы сервер считался недоступным. Значение по умолчанию: <code>1</code> .	<code>int</code>	Нет
<code>maxConns</code>	Задаёт максимальное число подключений к проксируемому серверу. Значение по умолчанию равно нулю, что означает отсутствие ограничений. Значение по умолчанию равно <code>0</code> .	<code>int</code>	Нет
<code>failTimeout</code>	Задаёт время, в течение которого должно произойти указанное количество неудачных попыток установить связь с сервером, чтобы считать сервер недоступным, и период времени, в течение которого сервер будет считаться недоступным. Значение по умолчанию равно <code>10</code> секундам.	<code>string</code>	Нет
<code>loadBalancingMethod</code>	Метод балансировки нагрузки между серверами апстрима. По умолчанию соединения распределяются между серверами по методу взвешенной циклической балансировки. Доступные методы и подробности смотрите в разделе <a href="#">Апстрим</a> .	<code>string</code>	Нет

## 6.2.4 UpstreamParameters

Различные параметры апстрима:

```
upstreamParameters:
  udpRequests: 1
  udpResponses: 1
  connectTimeout: 60s
  nextUpstream: true
  nextUpstreamTimeout: 50s
  nextUpstreamTries: 1
```

Поле	Описание	Тип	Обязательно
<code>udpRequests</code>	Количество датаграмм, после получения которых следующая датаграмма от того же клиента запускает новый сеанс. См. директиву <code>proxy_requests</code> . Значение по умолчанию равно 0.	<code>int</code>	Нет
<code>udpResponses</code>	Количество датаграмм, ожидаемых от проксируемого сервера в ответ на клиентскую датаграмму. См. директиву <code>proxy_responses</code> . По умолчанию количество датаграмм не ограничено.	<code>int</code>	Нет
<code>connectTimeout</code>	Тайм-аут установки соединения с проксируемым сервером. См. директиву <code>proxy_connect_timeout</code> . Значение по умолчанию - 60 секунд.	<code>string</code>	Нет
<code>nextUpstream</code>	Если соединение с проксируемым сервером установить не удастся, определяет, будет ли клиентское соединение передано на следующий сервер. См. директиву <code>proxy_next_upstream</code> . Значение по умолчанию равно <code>true</code> .	<code>bool</code>	Нет
<code>nextUpstreamTries</code>	Количество попыток до передачи соединения к следующему серверу. См. директиву <code>proxy_next_upstream_tries</code> . Значение по умолчанию равно 0.	<code>int</code>	Нет
<code>nextUpstreamTimeout</code>	Время, отведенное для передачи соединения к следующему серверу. См. директиву <code>proxy_next_upstream_timeout</code> . Значение по умолчанию - 0.	<code>string</code>	Нет

### 6.2.5 SessionParameters

Различные параметры для TCP-соединений и UDP-сеансов.

```
sessionParameters:
  timeout: 50s
```

Поле	Описание	Тип	Обязательно
<code>timeout</code>	Тайм-аут между двумя последовательными операциями чтения или записи в соединениях с клиентом или проксируемым сервером. См. директиву <code>proxy_timeout</code> . Значение по умолчанию равно 10m.	<code>string</code>	Нет

## 6.2.6 Action

Действие, которое необходимо выполнить для клиентского соединения или датаграммы.

В приведенном ниже примере клиентские подключения и датаграммы передаются на апстрим в `dns-app`:

```
action:
  pass: dns-app
```

Поле	Описание	Тип	Обязательно
<code>pass</code>	Передает соединения и датаграммы апстриму. Апстрим с таким именем должен быть определен в ресурсе.	<code>string</code>	Да

## 6.3 Использование TransportServer

Для работы с ресурсами `TransportServer` можно использовать обычные команды `kubectl`, аналогично ресурсам `Ingress`.

Например, следующая команда создает ресурс `TransportServer`, определенный в `transport-server-passthrough.yaml`, с именем `secure-app`:

```
kubectl apply -f transport-server-passthrough.yaml

transportserver.k8s.angie.software/secure-app created
```

Вы можете получить ресурс, выполнив:

```
kubectl get transportserver secure-app

NAME          AGE
secure-app    46sm
```

В `kubectl get` и подобных командах также можно использовать короткое имя `ts` вместо `transportserver`.

### 6.3.1 Использование фрагментов

Фрагменты позволяют вставлять элементы конфигурации `Angie` в различные контексты конфигурации `Angie`. В приведенном ниже примере мы используем фрагменты для настройки контроля доступа на `TransportServer`:

```
apiVersion: k8s.angie.software/v1alpha1
kind: TransportServer
metadata:
  name: cafe
spec:
  host: cafe.example.com
  serverSnippets: |
    deny 192.168.1.1;
    allow 192.168.1.0/24;
  upstreams:
  - name: tea
```

```
service: tea-svc
port: 80
```

Фрагменты также можно указать для потока. В приведенном ниже примере мы используем фрагменты для ограничения количества подключений :

```
apiVersion: k8s.angie.software/v1alpha1
kind: TransportServer
metadata:
  name: cafe
spec:
  host: cafe.example.com
  streamSnippets: limit_conn_zone $binary_remote_addr zone=addr:10m;
  serverSnippets: limit_conn addr 1;
  upstreams:
  - name: tea
    service: tea-svc
    port: 80
```

Фрагменты предназначены для продвинутых пользователей Angie, которым требуется больше контроля над генерируемой конфигурацией Angie.

Однако из-за недостатков, описанных ниже, фрагменты по умолчанию отключены. Чтобы использовать фрагменты, задайте аргумент командной строки `enable-snippets`.

Недостатки использования фрагментов:

- *Сложность.* Чтобы использовать фрагменты, требуется:
  - Понимать примитивы конфигурации Angie и реализовать правильную конфигурацию Angie.
  - Понимать, как ANIC генерирует конфигурацию Angie, чтобы фрагмент не мешал другим функциям конфигурации.
- *Сниженная надежность.* Неправильный фрагмент делает конфигурацию Angie недействительной, что приведет к ошибке при перезагрузке. Это помешает применить какие-либо обновления конфигурации, включая обновления для другого ресурса TransportServer, пока фрагмент не будет исправлен.
- *Последствия для безопасности.* Фрагменты предоставляют доступ к примитивам конфигурации Angie, и эти примитивы не проверяются самим ANIC.

#### Примечание

Пока конфигурация Angie содержит недопустимый фрагмент, Angie будет продолжать работать с последней допустимой конфигурацией.

#### Примечание

Чтобы настроить фрагменты в контексте `stream`, используйте ключ `stream-snippets` ConfigMap.

### 6.3.2 Валидация

Для ресурса `TransportServer` доступны два типа валидации:

- *Структурная валидация* с помощью `kubectl` и сервера Kubernetes API.
- *Всесторонняя валидация* с помощью ANIC.

#### Структурная валидация

Пользовательское определение ресурса для `TransportServer` включает структурную схему OpenAPI, которая описывает тип каждого поля ресурса.

Если вы попытаетесь создать (или обновить) ресурс с нарушением структурной схемы (например, используете строковое значение для поля порта апстрима), сервер `kubectl` и Kubernetes API отклонят такой ресурс:

- Пример проверки `kubectl`:

```
kubectl apply -f transport-server-passthrough.yaml

error: error validating "transport-server-passthrough.yaml": error validating
↪data: ValidationError(TransportServer.spec.upstreams[0].port): invalid type
↪for software.angie.k8s.v1alpha1.TransportServer.spec.upstreams.port: got
↪"string", expected "integer"; if you choose to ignore these errors, turn
↪validation off with --validate=false
```

- Пример проверки сервера Kubernetes API:

```
kubectl apply -f transport-server-passthrough.yaml --validate=false

The TransportServer "secure-app" is invalid: []: Invalid value:
↪map[string]interface {}{ ... }: validation failure list:
spec.upstreams.port in body must be of type integer: "string"
```

Если ресурс не отклонен (то есть не нарушает структурную схему), ANIC проверит его дополнительно.

#### Всесторонняя валидация

ANIC проверяет поля ресурса `TransportServer`. Если ресурс недействителен, ANIC отклонит его: ресурс продолжит существовать в кластере, но ANIC будет его игнорировать.

Вы можете проверить, успешно ли ANIC применил конфигурацию `TransportServer`. Для примера `TransportServer secure-app` мы можем запустить:

```
kubectl describe ts secure-app

. . .
Events:
  Type      Reason          Age   From                    Message
  ----
Normal    AddedOrUpdated  3s   angie-ingress-controller Configuration for default/
↪secure-app was added or updated
```

Обратите внимание, что раздел «События» (Events) включает событие `Normal` с причиной `AddedOrUpdated`, которое информирует нас о том, что конфигурация была успешно применена.

Если вы создадите недопустимый ресурс, ANIC отклонит его и выдаст событие `Rejected`. Например, если вы создадите `TransportServer secure-app` с действием `pass`, которое ссылается на несуществующий апстрим, вы получите:

```
kubectl describe ts secure-app
. . .
Events:
  Type          Reason      Age   From              Message
  ----          -
Warning        Rejected    2s    angie-ingress-controller TransportServer default/secure-app
↳ is invalid and was rejected: spec.action.pass: Not found: "some-app"
```

Обратите внимание, что раздел событий включает событие Warning с причиной Rejected.

**i Примечание**

Если вы внесете ошибку в уже существующий ресурс, контроллер входа отклонит его и удалит соответствующую конфигурацию из Angie.

### 6.3.3 Настройка с помощью ConfigMap

Ключи ConfigMap (за исключением stream-snippets, stream-log-format, resolver-addresses, resolver-ipv6, resolver-valid и resolver-timeout) не влияют на ресурсы TransportServer.

## ГЛАВА 7

---

### VirtualServer, VirtualServerRoute

---

Ресурсы VirtualServer и VirtualServerRoute реализуют сценарии использования, не поддерживаемые ресурсом Ingress, такие как разделение трафика и продвинутая маршрутизация на основе содержимого. Они реализованы как пользовательские ресурсы .

Это справочная документация по обоим ресурсам.

#### 7.1 Спецификация VirtualServer

Ресурс VirtualServer определяет конфигурацию балансировки нагрузки для доменного имени, например example.com. Ниже приведен пример такой конфигурации:

```
apiVersion: k8s.angie.software/v1
kind: VirtualServer
metadata:
  name: cafe
spec:
  host: cafe.example.com
  tls:
    secret: cafe-secret
  staticLocations:
  - type: root
    urlPath: /americano
    dirPath: /latte
  gunzip: on
  upstreams:
  - name: tea
    service: tea-svc
    port: 80
  - name: coffee
    service: coffee-svc
    port: 80
  routes:
  - path: /tea
    action:
      pass: tea
```

```

- path: /coffee
  action:
    pass: coffee
- path: ~ ~/decaf/.*\\.jpg$
  action:
    pass: coffee
- path: = /green/tea
  action:
    pass: tea
activeHealthProbes:
- name: activename1
  upstream: tea
  uri: uri
  port: 80
  interval: 3s
  isEssential: true
  isPersistent: true
  maxBody: 10m
  fails: 4
  passes: 5
  mode: onfail
maps:
- variable: $jwt_claim_iat
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: '80'
- variable: $jwt_claim_iss
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'PROVIDER_URL'
- variable: $jwt_claim_sub
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'myclient'
- variable: $jwt_claim_aud
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'myclient'

```

Поле	Описание	Тип	Обязательно
<code>host</code>	Хост (доменное имя) сервера. Это должен быть допустимый поддомен, как определено в RFC 1123, например <code>my-app</code> или <code>hello.example.com</code> . При использовании домена с подстановочным знаком, например <code>*.example.com</code> , домен должен быть заключен в двойные кавычки. Значение <code>host</code> должно быть уникальным среди всех ресурсов Ingress и VirtualServer.	<code>string</code>	Да
<code>tls</code>	Конфигурация завершения TLS.	<code>tls</code>	Нет
<code>staticLocations</code>	Список каталогов для раздачи статических файлов.	<code>//staticLocations</code>	Нет
<code>gunzip</code>	Включает или отключает распаковку архивированных ответов для клиентов. Допустимые пары значений: “on” и “off”, “true” и “false” или “yes” и “no”. Если значение <code>gunzip</code> не установлено, то по умолчанию оно равно <code>off</code> .	<code>boolean</code>	Нет
<code>ExternalDNS</code>	Конфигурация ExternalDNS для VirtualServer.	<code>ExternalDNS</code>	Нет
<code>dos</code>	Ссылка на <code>DosProtectedResource</code> ; установка этого параметра включает защиту VirtualServer от DOS-атак.	<code>string</code>	Нет
<code>policies</code>	Список политик.	<code>//policy</code>	Нет
<code>upstreams</code>	Список апстримов.	<code>//upstream</code>	Нет
<code>routes</code>	Список маршрутов.	<code>//route</code>	Нет
<code>activeHealthProbes</code>	Список активных проверок работоспособности.	<code>//activeHealth</code>	Нет
<code>maps</code>	Список переменных, необходимых для валидации токенов в процессе аутентификации <i>OIDC</i> .	<code>//maps</code>	Нет
<code>ingressClassName</code>	Указывает, какой экземпляр ANIC должен обрабатывать ресурс VirtualServer.	<code>string</code>	Нет
<code>internalRoute</code>	Указывает, является ли ресурс VirtualServer внутренним маршрутом.	<code>boolean</code>	Нет
<code>http-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>http</code> .	<code>string</code>	Нет
<code>server-snippets</code>	Задаёт пользовательский фрагмент в контексте <code>.</code> . Имеет приоритет над ключом <code>ConfigMap server-snippets</code> .	<code>string</code>	Нет

### 7.1.1 VirtualServer.TLS

Поле `tls` определяет конфигурацию TLS для ресурса VirtualServer. Например:

```
secret: cafe-secret
redirect:
  enable: true
```

Поле	Описание	Тип	Обязательно
<code>secret</code>	Имя секрета с сертификатом TLS и ключом. Секрет должен принадлежать тому же пространству имен, что и VirtualServer. Секрет должен иметь тип <code>kubernetes.io/tls</code> и содержать ключи с именами <code>tls.crt</code> и <code>tls.key</code> , содержащие сертификат и закрытый ключ, как описано <a href="#">здесь</a> . Если секрет не существует или недействителен, Angie прервет любую попытку установить TLS-соединение с хостом VirtualServer. Если секрет не указан, но настроен секрет TLS с подстановочным знаком, Angie будет использовать секрет со знаком для завершения TLS.	<code>string</code>	Нет
<code>redirect</code>	Конфигурация перенаправления TLS для VirtualServer.	<code>tls.redirect</code>	Нет
<code>cert-manager</code>	Конфигурация TLS cert-manager для VirtualServer.	<code>tls.cert-manager</code>	Нет

### 7.1.2 VirtualServer.TLS.Redirect

Поле перенаправления настраивает перенаправление TLS для VirtualServer:

```
enable: true
code: 301
basedOn: scheme
```

Поле	Описание	Тип	Обязательно
<code>enable</code>	Включает перенаправление TLS для VirtualServer. Значение по умолчанию равно <code>false</code> .	<code>boolean</code>	Нет
<code>код</code>	Код состояния перенаправления. Допустимые значения: 301, 302, 307, 308. Значение по умолчанию - 301.	<code>int</code>	Нет
<code>basedOn</code>	Атрибут запроса, который Angie будет оценивать для отправки перенаправления. Допустимыми значениями являются <code>scheme</code> (схема запроса) или <code>x-forwarded-proto</code> (заголовок X-Forwarded-Proto запроса). Значение по умолчанию - <code>scheme</code> .	<code>string</code>	Нет

### 7.1.3 VirtualServer.TLS.CertManager

Поле `cert-manager` настраивает автоматическое управление сертификатами x509 для ресурсов `VirtualServer` с помощью `cert-manager` ([cert-manager.io](https://cert-manager.io)). Ознакомьтесь с [документацией по конфигурации cert-manager](#) для получения дополнительной информации о развертывании и настройке эмитентов (Issuer). Пример:

```
cert-manager:  
  cluster-issuer: "my-issuer-name"
```

Поле	Описание	Тип	Обязательно
<code>issuer</code>	Имя эмитента. Эмитент - это ресурс <code>cert-manager</code> , который описывает центр сертификации, способный подписывать сертификаты. Он должен находиться в том же пространстве имен, что и ресурс <code>VirtualServer</code> . Обратите внимание, что требуется задать <code>issuer</code> или <code>cluster-issuer</code> , но эти параметры взаимоисключающие - должен быть задан один и только один.	<code>string</code>	Нет
<code>cluster-issuer</code>	Имя <code>ClusterIssuer</code> . <code>ClusterIssuer</code> - это ресурс <code>cert-manager</code> , который описывает центр сертификации, способный подписывать сертификаты. Не имеет значения, в каком пространстве имен находится ваш <code>VirtualServer</code> , поскольку <code>ClusterIssuer</code> - это ресурсы, не относящиеся к пространствам имен. Обратите внимание, что требуется задать <code>issuer</code> или <code>cluster-issuer</code> , но эти параметры взаимоисключающие - должен быть задан один и только один.	<code>string</code>	Нет
<code>issuer-kind</code>	Тип внешнего ресурса-эмитента, например <code>AWSPCAIssuer</code> . Это необходимо только для сторонних эмитентов. Его нельзя задавать, если также задан <code>cluster-issuer</code> .	<code>string</code>	Нет
<code>issuer-group</code>	Группа API внешнего контроллера-эмитента, например <code>awspca.cert-manager.io</code> . Это необходимо только для сторонних эмитентов. Его нельзя задавать, если также задан <code>cluster-issuer</code> .	<code>string</code>	Нет
<code>common-name</code>	Это поле позволяет настроить <code>spec.commonName</code> для создаваемого сертификата. Эта конфигурация добавляет CN к сертификату x509.	<code>string</code>	Нет
<code>duration</code>	Это поле позволяет настроить поле <code>spec.duration</code> для генерируемого сертификата. Оно должно быть задано с использованием формата <code>time.Duration</code> в Go, который не допускает суффикса <code>d</code> (дни). Указывайте такие значения, используя вместо них суффиксы <code>s</code> , <code>m</code> и <code>h</code> .	<code>string</code>	Нет
<code>renew-before</code>	Эта аннотация позволяет настроить поле <code>spec.renewBefore</code> для генерируемого сертификата. Оно должно быть задано с использованием формата <code>time.Duration</code> в Go, который не допускает суффикса <code>d</code> (дни). Указывайте такие значения, используя вместо них суффиксы <code>s</code> , <code>m</code> и <code>h</code> .	<code>string</code>	Нет
<code>usages</code>	Позволяет настроить поле <code>spec.usages</code> для генерируемого сертификата. Задайте строку со значениями, разделенными запятыми, т. е. соглашение о ключе, цифровая подпись, серверная аутентификация. Исчерпывающий список поддерживаемых способов использования ключей можно найти в документации <a href="#">API cert-manager</a> .	<code>string</code>	Нет

### 7.1.4 VirtualServer.ExternalDNS

Поле ExternalDNS настраивает динамическое управление записями DNS для ресурсов VirtualServer с использованием ExternalDNS. Ознакомьтесь с документацией по конфигурации ExternalDNS для получения дополнительной информации о развертывании и настройке ExternalDNS и поставщиков. Пример:

```
enable: true
```

Поле	Описание	Тип	Обязательно
enable	Включает интеграцию ExternalDNS для ресурса VirtualServer. Значение по умолчанию равно false.	string	Нет
labels	Настраивает метки, применяемые к ресурсам конечной точки, которые будут использоваться ExternalDNS.	map[string]	Нет
providerSpecific	Настраивает свойства, относящиеся к конкретному поставщику, которые содержат имя и значение конфигурации, специфичной для отдельных поставщиков DNS.	<i>///ProviderSp</i>	Нет
recordTTL	TTL для записи DNS. По умолчанию это значение равно 0. См. документацию ExternalDNS TTL для определения значений по умолчанию для конкретного поставщика	int64	Нет
recordType	Тип создаваемой записи, например «A», «AAAA», «CNAME». Если значение не задано, оно автоматически вычисляется на основе внешних конечных точек.	string	Нет

### 7.1.5 VirtualServer.ExternalDNS.ProviderSpecific

Поле providerSpecific блока ExternalDNS позволяет указать свойства, специфичные для поставщика, которые представляют собой список пар «ключ-значение» для конфигураций, специфичных для отдельных поставщиков DNS. Пример:

```
- name: my-name
  value: my-value
- name: my-name2
  value: my-value2
```

Поле	Описание	Тип	Обязательно
name	Имя в паре «ключ-значение».	string	Да
value	Значение в паре «ключ-значение».	string	Да

### 7.1.6 VirtualServer.Policy

Ссылается на ресурс [Policy](#) по имени и необязательному пространству имен. Например:

```
name: access-control
```

Поле	Описание	Тип	Обязательно
name	Имя политики. Если политика не существует или недействительна, Angie выдаст сообщение об ошибке с кодом состояния 500.	string	Да
namespace	Пространство имен политики. Если не указано, используется пространство имен ресурса VirtualServer.	string	Нет

### 7.1.7 VirtualServer.Route

Маршрут определяет правила для сопоставления клиентских запросов с такими действиями, как передача запроса апстриму. Например:

```
path: /tea
action:
  pass: tea
```

Поле	Описание	Тип	Обязательно
<code>path</code>	Путь маршрута. Angie сопоставит его с URI запроса. Возможные значения: префикс ( <code>/</code> , <code>/path</code> ), точное совпадение ( <code>=/exact/match</code> ), регулярное выражение без учета регистра ( <code>~*/Var.*\.</code> <code>jpg</code> ) или регулярное выражение с учетом регистра ( <code>~/foo.*\.</code> <code>jpg</code> ). В случае префикса (должен начинаться с <code>/</code> ) или точного совпадения (должно начинаться с <code>=</code> ) путь не должен содержать никаких пробельных символов, <code>{</code> , <code>}</code> или <code>;</code> . В случае регулярных выражений все двойные кавычки <code>"</code> должны быть экранированы, при этом совпадение не может заканчиваться неэкранированной обратной косой чертой <code>\</code> . Путь должен быть уникальным среди путей всех маршрутов <code>VirtualServer</code> . Дополнительные сведения см. в описании директивы <code>location</code> .	<code>string</code>	Да
<code>policies</code>	Список политик. Эти политики имеют приоритет над политиками того же типа, определенными в спецификации <code>VirtualServer</code> . Более подробную информацию смотрите в <i>Применение политик</i> .	<code>//policy</code>	Нет
<code>action</code>	Действие по умолчанию, выполняемое для запроса.	<code>Action</code>	Нет
<code>dos</code>	Ссылка на <code>DosProtectedResource</code> ; установка этого параметра включает защиту маршрута <code>VirtualServer</code> от DOS-атак.	<code>string</code>	Нет
<code>splits</code>	Конфигурация разделения трафика по умолчанию. Должно быть не менее 2 разделений.	<code>Split</code>	Нет
<code>matches</code>	Правила сопоставления для продвинутой маршрутизации на основе содержимого. Требуется задать <code>action</code> или <code>splits</code> по умолчанию. Несопоставленные запросы будут обрабатываться <code>action</code> или <code>splits</code> по умолчанию.	<code>matches</code>	Нет
<code>route</code>	Имя ресурса <code>VirtualServerRoute</code> , который определяет этот маршрут. Если <code>VirtualServerRoute</code> не принадлежит к тому же пространству имен, что и <code>VirtualServer</code> , необходимо включить пространство имен. Например: <code>tea-namespace/tea</code> .	<code>string</code>	Нет
<code>errorPages</code>	Настраиваемые ответы на коды ошибок. Angie будет использовать эти ответы вместо того, чтобы возвращать ответы об ошибках с серверов апстрима или ответы по умолчанию, сгенерированные Angie. Настраиваемый ответ может быть перенаправлением или сохраненным ответом. Например, это может быть перенаправление на другой URL-адрес, если вышестоящий сервер ответил кодом состояния 404.	<code>//errorPage</code>	Нет
<code>location-snippets</code>	Задаёт пользовательский фрагмент в контексте местоположения. Имеет приоритет над ключом <code>ConfigMap location-snippets</code> .	<code>string</code>	Нет

### Примечание

Маршрут должен включать в себя ровно одно из следующих действий: `action`, `splits` или `route`.

## 7.1.8 Maps

Определяет обязательные переменные `$jwt_claim_iat`, `$jwt_claim_iss`, `$jwt_claim_sub` и `$jwt_claim_aud` для валидации токенов в процессе *аутентификации OIDC*.

```
- variable: $jwt_claim_iat
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: '80'
- variable: $jwt_claim_iss
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'PROVIDER_URL'
- variable: $jwt_claim_sub
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'myclient'
- variable: $jwt_claim_aud
  source: $oidc_client
  parameters:
    - value: 'myclient'
      result: 'myclient'
```

Пример включения переменных «map» в зависимости от входного значения (`default`, `volatile`, `include`, `hostnames`):

```
maps:
- variable: $result_var
  source: $host
  parameters:
    - value: 'default'
      result: 'default_value'
    - value: 'volatile'
      result: ''
    - value: 'include'
      result: '/dev/stdout'
    - value: 'example.com'
      result: '1'
    - value: '*.example.com'
      result: '1'
```

См. также директиву `map` в документации Angie.

Поле	Описание	Тип	Обязательно
\$jwt_cl:	Настраивает параметр <code>iat</code> (issued at, время выпуска токена) для клиента.	string	Да
\$jwt_cl:	Параметр <code>iss</code> (issuer, издатель токена) сопоставляется с URL-адресом <code>PROVIDER_URL</code> . Этот параметр указывает на сервис, выпустивший токен.	string	Да
\$jwt_cl:	Параметр <code>sub</code> (subject, субъект токена). Идентифицирует пользователя или субъект, для которого был выдан токен.	string	Да
\$jwt_cl:	Параметр <code>aud</code> (audience, аудитория). Идентифицирует клиент, для которого был предназначен токен.	string	Да

## 7.2 Спецификация VirtualServerRoute

Ресурс `VirtualServerRoute` определяет маршрут для `VirtualServer`. Он может состоять из одного вложенного маршрута или нескольких. `VirtualServerRoute` является альтернативой объединяемым типам `Ingress`.

В приведенном ниже примере виртуальный сервер `cafe` из пространства имен `cafe-ns` определяет маршрут с путем `/coffee`, который далее определяется через `VirtualServerRoute coffee` из пространства имен `coffee-ns`.

VirtualServer:

```
apiVersion: k8s.angie.software/v1
kind: VirtualServer
metadata:
  name: cafe
  namespace: cafe-ns
spec:
  host: cafe.example.com
  upstreams:
  - name: tea
    service: tea-svc
    port: 80
  routes:
  - path: /tea
    action:
      pass: tea
  - path: /coffee
    route: coffee-ns/coffee
```

VirtualServerRoute:

```
apiVersion: k8s.angie.software/v1
kind: VirtualServerRoute
metadata:
  name: coffee
  namespace: coffee-ns
spec:
  host: cafe.example.com
  upstreams:
  - name: latte
    service: latte-svc
    port: 80
  - name: espresso
```

```

service: espresso-svc
port: 80
subroutes:
- path: /coffee/latte
  action:
    pass: latte
- path: /coffee/espresso
  action:
    pass: espresso

```

Обратите внимание, что каждый вложенный маршрут должен иметь путь `path`, начинающийся с того же префикса (здесь «/coffee»), что и в маршруте `VirtualServer`. Кроме того, `host` в `VirtualServerRoute` должен совпадать с `host` у `VirtualServer`.

Поле	Описание	Тип	Обязательно
<code>host</code>	Хост (доменное имя) сервера. Это должен быть допустимый поддомен, как определено в RFC 1123, например <code>my-app</code> или <code>hello.example.com</code> . При использовании домена с подстановочным знаком, например <code>*.example.com</code> , домен должен быть заключен в двойные кавычки. Значение должно совпадать с <code>host</code> у <code>VirtualServer</code> , который ссылается на этот ресурс.	<code>string</code>	Да
<code>upstreams</code>	Список апстримов.	<code>//upstream</code>	Нет
<code>subroutes</code>	Список вложенных маршрутов.	<code>//subroute</code>	Нет
<code>ingressClassName</code>	Указывает, какой экземпляр ANIC должен обрабатывать ресурс <code>VirtualServerRoute</code> . Значение должно совпадать с <code>ingressClassName</code> у <code>VirtualServer</code> , который ссылается на этот ресурс.	<code>string</code>	Нет

### 7.2.1 VirtualServerRoute.Subroute

Определяет правила сопоставления клиентских запросов и действий, например передача запроса апстриму. Например:

```

path: /coffee
action:
  pass: coffee

```

Поле	Описание	Тип	Обязательно
<code>path</code>	Путь вложенного маршрута. Angie сопоставит его с URI запроса. Возможные значения: префикс ( <code>/</code> , <code>/path</code> ), точное совпадение ( <code>=/exact/match</code> ), регулярное выражение без учета регистра ( <code>~*/Bar.*\ .jpg</code> ) или регулярное выражение с учетом регистра ( <code>~/foo.*\ .jpg</code> ). В случае префикса путь должен начинаться с того же пути, что и путь маршрута <code>VirtualServer</code> , который ссылается на этот ресурс. В случае точного совпадения или регулярного выражения путь должен совпадать с путем маршрута <code>VirtualServer</code> , который ссылается на этот ресурс. В случае префикса или точного совпадения путь не должен содержать никаких пробельных символов, <code>{</code> , <code>}</code> или <code>;</code> . В случае регулярных выражений все двойные кавычки " должны быть экранированы, при этом совпадение не может заканчиваться неэкранированной обратной косой чертой <code>.</code> Путь должен быть уникальным среди путей всех вложенных маршрутов <code>VirtualServerRoute</code> .	<code>string</code>	Да
<code>policies</code>	Список политик. Эти политики имеют приоритет над <i>всеми</i> политиками, определенными в маршруте <code>VirtualServer</code> , который ссылается на этот ресурс. Они также имеют приоритет над политиками того же типа, определенными в спецификации <code>VirtualServer</code> . Более подробную информацию смотрите в разделе <i>Применение политик</i> .	<code>//policy</code>	Нет
<code>action</code>	Действие по умолчанию, выполняемое для запроса.	<code>Action</code>	Нет
<code>dos</code>	Ссылка на <code>DosProtectedResource</code> ; установка этого параметра включает защиту вложенного маршрута <code>VirtualServerRoute</code> от DOS-атак.	<code>string</code>	Нет
<code>splits</code>	Конфигурация разделения трафика по умолчанию. Должно быть не менее 2 разделений.	<code>//split</code>	Нет
<code>matches</code>	Правила сопоставления для продвинутой маршрутизации на основе содержимого. Требуется задать <code>action</code> или <code>splits</code> по умолчанию. Несопоставленные запросы будут обрабатываться <code>action</code> или <code>splits</code> по умолчанию.	<code>matches</code>	Нет
<code>errorPages</code>	Настраиваемые ответы на коды ошибок. Angie будет использовать эти ответы вместо того, чтобы возвращать ответы об ошибках с серверов апстрима или ответы по умолчанию, сгенерированные Angie. Настраиваемый ответ может быть перенаправлением или сохраненным ответом. Например, это может быть перенаправление на другой URL-адрес, если вышестоящий сервер ответил кодом состояния 404.	<code>//errorPage</code>	Нет
<code>location-snippets</code>	Задаёт пользовательский фрагмент в контексте местоположения. Переопределяет значение <code>location-snippets</code> <code>VirtualServer</code> (если задано) или ключ <code>ConfigMap</code> <code>location-snippets</code> .	<code>string</code>	Нет

#### Примечание

Вложенный маршрут должен включать в себя ровно одно из следующих действий: `action` или `splits`.

## 7.3 Общие части `VirtualServer` и `VirtualServerRoute`

### 7.3.1 Upstream

Апстрим определяет конечное место назначения для конфигурации маршрутизации. Например:

```
name: tea
service: tea-svc
subselector:
  version: canary
port: 80
lb-method: round_robin
fail-timeout: 10s
max-fails: 1
max-conns: 32
keepalive: 32
connect-timeout: 30s
read-timeout: 30s
send-timeout: 30s
next-upstream: "error timeout non_idempotent"
next-upstream-timeout: 5s
next-upstream-tries: 10
client-max-body-size: 2m
tls:
  enable: true
```

#### Примечание

Протокол WebSocket поддерживается без какой-либо дополнительной настройки.

Поле	Описание	Тип	Обязательно
name	Имя апстрима. Это должна быть допустимая метка DNS, как определено в RFC 1035. Например, допустимы значения <code>hello</code> и <code>upstream-123</code> . Имя должно быть уникальным среди всех апстримов ресурса.	string	Да
service	Название сервиса. Сервис должен принадлежать к тому же пространству имен, что и ресурс. Если сервиса не существует, Angie предположит, что у него нет конечных точек, и будет возвращать ответ 502 для запросов к этому апстриму.	string	Да
subselector	Выбирает поды внутри сервиса, используя ключи меток и значения. По умолчанию выбраны все поды сервиса.	map[string]	Нет
<div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p><b>Примечание</b></p> <p>Ожидается, что указанные метки будут присутствовать в подах при их создании. Если метки подов изменяются, ANIC не увидит это изменение до тех пор, пока не будет изменено количество подов.</p> </div>			
use-cluster-ip	Позволяет использовать IP-адрес кластера и порт сервиса вместо использования IP-адреса и порта подов по умолчанию. Когда это поле включено, поля, которые настраивают поведение Angie, относящееся к нескольким серверам апстрима (например, <code>lb-method</code> и <code>next-upstream</code> ), не будут иметь никакого эффекта, поскольку ANIC настроит Angie только с одним сервером апстрима, который будет соответствовать IP-адресу кластера сервиса.	boolean	Нет
port	Порт службы. Если у сервиса не определен этот порт, Angie предположит, что у него нет конечных точек, и будет возвращать ответ 502 для запросов к этому апстриму. Значение должно находиться в диапазоне 1..65535.	uint16	Да
lb-method	Метод балансировки нагрузки. Чтобы использовать циклический метод, укажите <code>round_robin</code> . Значение по умолчанию указано в ключе <code>lb-method</code> ConfigMap.	string	Нет
fail-timeout	Время, в течение которого должно произойти указанное количество неудачных попыток установить связь с сервером апстрима, чтобы он считался недоступным. См. параметр <code>fail_timeout</code> директивы сервера. Значение по умолчанию задано в ключе ConfigMap <code>fail-timeout</code> .	string	Нет
max-fails	Количество неудачных попыток установить связь с сервером апстрима, которые должны произойти в течение времени, заданного в <code>fail-timeout</code> , чтобы считать сервер недоступным. См. параметр <code>max_fails</code> директивы сервера. Значение по умолчанию задано в ключе ConfigMap <code>max-fails</code> .	int	Нет

### 7.3. Общие части VirtualServer и VirtualServerRoute

max-conns	Максимальное количество одновременных активных подключений к серверу апстрима. См. параметр <code>max_conns</code> директивы сервера. По умолчанию ограниченный нет.	int	Нет
-----------	--	-----	-----

### 7.3.2 Upstream Buffers

Настраивает буферы, используемые для чтения ответа от сервера апстрима в рамках одного соединения.

```
number: 4
size: 8K
```

См. директиву `proxy_buffers` для получения дополнительной информации.

Поле	Описание	Тип	Обязательно
<code>number</code>	Задаёт количество буферов. Значение по умолчанию задано в ключе <code>ConfigMap proxy-buffers</code> .	<code>int</code>	Да
<code>size</code>	Задаёт размер буфера. Значение по умолчанию задано в ключе <code>ConfigMap proxy-buffers (8K)</code> .	<code>string</code>	Нет

### 7.3.3 Upstream.TLS

Поле	Описание	Тип	Обязательно
<code>enable</code>	Включает HTTPS для запросов к серверам апстрима. Значение по умолчанию равно <code>False</code> , что означает, что будет использоваться HTTP.	<code>boolean</code>	Нет

**Примечание**

По умолчанию Angie не будет проверять сертификат вышестоящего сервера. Чтобы включить проверку, настройте политику `EgressMTLS`.

### 7.3.4 Upstream.SessionCookie

Поле `SessionCookie` настраивает сохранение сеансов, что позволяет передавать запросы от одного и того же клиента на один и тот же сервер апстрима. Информация о назначенном сервере апстрима передается в сеансовом cookie, сгенерированном Angie.

В приведенном ниже примере мы настраиваем сохранение сеанса с помощью cookie сеанса для апстрима и задаем все доступные параметры:

```
name: tea
service: tea-svc
port: 80
sessionCookie:
  enable: true
  name: srv_id
  path: /
  expires: 1h
  domain: .example.com
  httpOnly: false
```

```
secure: true
samesite: strict
```

См. директиву `sticky` для получения дополнительной информации. Сеансовый cookie соответствует методу `sticky cookie`.

Поле	Описание	Тип	Обязательно
<code>enable</code>	Включает сохранение сеанса с помощью сеансового cookie для сервера апстрима. Значение по умолчанию равно <code>false</code> .	<code>boolean</code>	Нет
<code>name</code>	Имя cookie.	<code>string</code>	Да
<code>path</code>	Путь, для которого установлен cookie.	<code>string</code>	Нет
<code>expires</code>	Время, в течение которого браузер должен сохранять cookie. Может быть установлено специальное значение <code>max</code> ; тогда срок действия cookie истечет 31 декабря 2037 года в 23:55:55 по Гринвичу.	<code>string</code>	Нет
<code>domain</code>	Домен, для которого установлен cookie.	<code>string</code>	Нет
<code>httpOnly</code>	Добавляет атрибут <code>HttpOnly</code> к cookie.	<code>boolean</code>	Нет
<code>secure</code>	Добавляет атрибут <code>Secure</code> к cookie.	<code>boolean</code>	Нет
<code>samesite</code>	Добавляет атрибут <code>SameSite</code> к cookie. Допустимые значения: <code>strict</code> , <code>lax</code> , <code>none</code>	<code>string</code>	Нет

### 7.3.5 Upstream.SessionRoute

Поле `sessionRoute` настраивает сохранение маршрутов, что позволяет передавать запросы от одного и того же клиента на один и тот же сервер апстрима. Информация о назначенном сервере апстрима поддерживается в режиме `route` Angie.

В приведенном ниже примере мы формируем директиву Angie `sticky route $cookie_route $arg_route`;

```
sessionRoute:
  enable: true
  variables:
    \- "$cookie_route"
    \- "$arg_route"
```

См. описание режима `route` директивы `sticky` для получения дополнительной информации.

Параметры:

Поле	Описание	Тип	Обязательно
<code>enable</code>	Включает сохранение сеанса в режиме <code>route</code> для сервера апстрима. Значение по умолчанию равно <code>false</code> .	<code>boolean</code>	Нет
<code>variables</code>	Список переменных, подставляемых в директиву <code>sticky route</code> в порядке следования.	<code>[]string</code>	Нет

### 7.3.6 ActiveHealthProbes

Поле позволяет настроить активную проверку работоспособности (health probe) для *upstream*. Вам необходимо задать имя проверки `name` и `upstream`, к которому она относится, а также другие параметры. Подробное описание параметров для активной проверки работоспособности см. в документации Angie, директива `upstream_probe`.

Пример:

```
activeHealthProbes:
- name: activename1
  upstream: tea-post
  uri: uri
  port: 80
  interval: 3s
  isEssential: true
  isPersistent: true
  maxBody: 10m
  fails: 4
  passes: 5
  mode: onfail
```

### 7.3.7 staticLocations

Поле позволяет задать каталог, из которого будут раздаваться статические файлы. Вы можете указать корневой каталог с помощью директивы `root` или другое расположение с помощью директивы `alias`, см. описания директив в документации Angie.

Пример конфигурации:

```
staticLocations:
- type: root
  urlPath: /static
  dirPath: /var/www/html
```

Поле	Описание	Тип	Обязательно
<code>type</code>	Способ определения пути к каталогу, из которого будут раздаваться статические файлы. Возможные значения: <code>root</code> , <code>alias</code> .	<code>string</code>	Да
<code>urlPath</code>	Префикс URL-адресов запрашиваемых статических файлов, используемый для <code>location</code> ; см. описание директивы в документации Angie.	<code>string</code>	Да
<code>dirPath</code>	Каталог файловой системы, из которого будут обслуживаться запросы к указанному URL-адресу.	<code>string</code>	Да

### 7.3.8 Header

Определяет HTTP-заголовок:

```
name: Host
value: example.com
```

Поле	Описание	Тип	Обязательно
<code>name</code>	Имя заголовка.	<code>string</code>	Да
<code>value</code>	Значение заголовка.	<code>string</code>	Нет

### 7.3.9 Action

Определяет действие, которое необходимо выполнить для запроса.

В приведенном ниже примере клиентские запросы передаются на апстрим `coffee`:

```
path: /coffee
action:
  pass: coffee
```

Поле	Описание	Тип	Обязательно
<code>pass</code>	Передаёт запросы серверу апстрима. Апстрим с таким именем должен быть определен в ресурсе.	<code>string</code>	Нет
<code>redirect</code>	Перенаправляет запросы на указанный URL-адрес.	<code>action.redirect</code>	Нет
<code>return</code>	Возвращает предварительно сконфигурованный ответ.	<code>action.return</code>	Нет
<code>proxy</code>	Передаёт запросы апстриму, добавляет возможность изменять запрос и ответ (например, переписывать URI или изменять заголовки).	<code>action.proxy</code>	Нет

#### Примечание

Действие должно включать в себя ровно одно из следующих значений: `pass`, `redirect`, `return` или `proxy`.

### 7.3.10 Action.Redirect

Определяет перенаправление, возвращаемое для запроса.

В приведенном ниже примере клиентские запросы направляются на URL-адреса `http://myhost.ru`:

```
redirect:
  url: http://myhost.ru
  code: 301
```

Поле	Описание	Тип	Обязательно
url	URL-адрес, на который будет перенаправлен запрос. Поддерживаемые переменные Angie: <code>\$scheme</code> , <code>\$http_x_forwarded_proto</code> , <code>\$request_uri</code> , <code>\$host</code> . Переменные должны быть заключены в фигурные скобки. Например: <code>\${host}\${request_uri}</code> .	string	Да
код	Код состояния перенаправления. Допустимые значения: 301, 302, 307, 308. Значение по умолчанию - 301.	int	Нет

### 7.3.11 Action.Return

Определяет предварительно сконфигурированный ответ на запрос.

В приведенном ниже примере Angie будет отвечать предварительно настроенным ответом на каждый запрос:

```
return:
  code: 200
  type: text/plain
  body: "Hello World\n"
```

Поле	Описание	Тип	Обязательно
код	Код состояния ответа. Допустимые значения: 2XX, 4XX или 5XX. Значение по умолчанию равно 200.	int	Нет
тип	MIME-тип ответа. Значение по умолчанию - <code>text/plain</code> .	string	Нет
body	Основная часть ответа. Поддерживает переменные Angie*. Переменные должны быть заключены в фигурные скобки. Например: Запрос равен <code>\${request_uri}\n</code> .	string	Да

#### Примечание

Поддерживаемые переменные Angie: `$request_uri`, `$request_method`, `$request_body`, `$scheme`, `$http_`, `$args`, `$arg_`, `$cookie_`, `$host`, `$request_time`, `$request_length`, `$angie_version`, `$pid`, `$connection`, `$remote_addr`, `$remote_port`, `$time_iso8601`, `$time_local`, `$server_addr`, `$server_port`, `$server_name`, `$server_protocol`, `$connections_active`, `$connections_reading`, `$connections_writing` и `$connections_waiting`.

### 7.3.12 Action.Proxy

Передаёт запросы апстриму с возможностью изменять запрос и ответ (например, переписывать URI или изменять заголовки).

В приведенном ниже примере URI запроса переписывается на `/`, а заголовки запроса и ответа изменяются:

```
proxy:
  upstream: coffee
  requestHeaders:
    pass: true
    set:
      - name: My-Header
```

```

value: Value
- name: Client-Cert
  value: ${ssl_client_escaped_cert}
responseHeaders:
  add:
    - name: My-Header
      value: Value
    - name: IC-Angie-Version
      value: ${angie_version}
      always: true
  hide:
    - x-internal-version
  ignore:
    - Expires
    - Set-Cookie
  pass:
    - Server
rewritePath: /

```

Поле	Описание	Тип	Обязательно
<code>upstream</code>	Имя апстрима, куда будут проксироваться запросы. Апстрим с таким именем должен быть определен в ресурсе.	<code>string</code>	Да
<code>requestHeaders</code>	Изменения заголовков запросов.	<i>Action.Proxy</i>	Нет
<code>responseHeaders</code>	Изменения в заголовках ответов.	<i>Action.Proxy</i>	Нет
<code>rewritePath</code>	Переписанный URI. Если путь маршрута является регулярным выражением, т. е. начинается с <code>~</code> , то <code>rewritePath</code> может включать группы захвата <code>\$1-9</code> . Например, <code>\$1</code> - первая группа, и так далее.	<code>string</code>	Нет

### 7.3.13 Action.Proxy.RequestHeaders

Поле `requestHeaders` изменяет заголовки запроса к проксируемому серверу апстрима.

Поле	Описание	Тип	Обязательно
<code>pass</code>	Передаёт исходные заголовки запроса на проксируемый сервер апстрима. Дополнительные сведения см. в описании директивы <code>proxy_pass_request_header</code> . Значение по умолчанию - <code>true</code> .	<code>bool</code>	Нет
<code>set</code>	Позволяет переопределять или добавлять поля для представления заголовков запросов, передаваемых на проксируемые серверы апстрима. Дополнительные сведения см. в описании директивы <code>proxy_set_header</code> .	<i>//header</i>	Нет

### 7.3.14 Action.Proxy.RequestHeaders.Set.Header

Определяет HTTP-заголовок:

```
name: My-Header
value: My-Value
```

Можно переопределить значение заголовка `Host` по умолчанию, которое ANIC устанавливает равным `$host`:

```
name: Host
value: example.com
```

Поле	Описание	Тип	Обязательно
<code>name</code>	Имя заголовка.	string	Да
<code>value</code>	Значение заголовка. Поддерживает переменные Angie*. Переменные должны быть заключены в фигурные скобки. Например: <code>\${scheme}</code> .	string	Нет

#### Примечание

Поддерживаемые переменные Angie: `$request_uri`, `$request_method`, `$request_body`, `$scheme`, `$http_`, `$args`, `$arg_`, `$cookie_`, `$host`, `$request_time`, `$request_length`, `$angie_version`, `$pid`, `$connection`, `$remote_addr`, `$remote_port`, `$time_iso8601`, `$time_local`, `$server_addr`, `$server_port`, `$server_name`, `$server_protocol`, `$connections_active`, `$connections_reading`, `$connections_writing`, `$connections_waiting`, `$ssl_cipher`, `$ssl_ciphers`, `$ssl_client_cert`, `$ssl_client_escaped_cert`, `$ssl_client_fingerprint`, `$ssl_client_i_dn`, `$ssl_client_i_dn_legacy`, `$ssl_client_raw_cert`, `$ssl_client_s_dn`, `$ssl_client_s_dn_legacy`, `$ssl_client_serial`, `$ssl_client_v_end`, `$ssl_client_v_remain`, `$ssl_client_v_start`, `$ssl_client_verify`, `$ssl_curves`, `$ssl_early_data`, `$ssl_protocol`, `$ssl_server_name`, `$ssl_session_id`, `$ssl_session_reused`.

### 7.3.15 Action.Proxy.ResponseHeaders

Поле `responseHeaders` изменяет заголовки ответа клиенту.

Поле	Описание	Тип	Обязательно
hide	Заголовки, которые не будут переданы в ответе клиенту с проксируемого сервера апстрима. Дополнительные сведения см. в описании директивы <code>proxy_hide_header</code> .	[]string	Нет
pass	Позволяет передавать скрытые поля заголовка клиенту с проксируемого сервера апстрима. Дополнительные сведения см. в описании директивы <code>proxy_pass_header</code> .	[]string	Нет
ignore	Отключает обработку определенных заголовков** при передаче клиенту ответа с проксируемого сервера апстрима. Дополнительные сведения см. в описании директивы <code>proxy_ignore_headers</code> .	[]string	Нет
add	Добавляет заголовки к ответу для клиента.	<code>//addHeader</code>	Нет

#### Примечание

Скрытые заголовки по умолчанию: `Date`, `Server`, `X-Pad` и `X-Accel-...`

#### Примечание

Следующие поля могут быть проигнорированы: `X-Accel-Redirect`, `X-Accel-Expires`, `X-Accel-Limit-Rate`, `X-Accel-Buffering`, `X-Accel-Charset`, `Expires`, `Cache-Control`, `Set-Cookie` и `Vary`.

### 7.3.16 AddHeader

Определяет HTTP-заголовок с необязательным полем `always`:

```
name: My-Header
value: My-Value
always: true
```

Поле	Описание	Тип	Обязательно
name	Имя заголовка.	string	Да
value	Значение заголовка. Поддерживает переменные Angie*. Переменные должны быть заключены в фигурные скобки. Например: <code>\${scheme}</code> .	string	Нет
always	Если установлено значение <code>true</code> , добавляет заголовков независимо от кода состояния ответа**. Значение по умолчанию - <code>false</code> . Дополнительные сведения см. в описании директивы <code>add_header</code> .	bool	Нет

**i Примечание**

Поддерживаемые переменные Angie: `$request_uri`, `$request_method`, `$request_body`, `$scheme`, `$http_`, `$args`, `$arg_`, `$cookie_`, `$host`, `$request_time`, `$request_length`, `$angie_version`, `$pid`, `$connection`, `$remote_addr`, `$remote_port`, `$time_iso8601`, `$time_local`, `$server_addr`, `$server_port`, `$server_name`, `$server_protocol`, `$connections_active`, `$connections_reading`, `$connections_writing`, `$connections_waiting`, `$ssl_cipher`, `$ssl_ciphers`, `$ssl_client_cert`, `$ssl_client_escaped_cert`, `$ssl_client_fingerprint`, `$ssl_client_i_dn`, `$ssl_client_i_dn_legacy`, `$ssl_client_raw_cert`, `$ssl_client_s_dn`, `$ssl_client_s_dn_legacy`, `$ssl_client_serial`, `$ssl_client_v_end`, `$ssl_client_v_remain`, `$ssl_client_v_start`, `$ssl_client_verify`, `$ssl_curves`, `$ssl_early_data`, `$ssl_protocol`, `$ssl_server_name`, `$ssl_session_id`, `$ssl_session_reused`.

**i Примечание**

Если значение `always` - `false`, заголовок ответа добавляется только в том случае, если код состояния ответа - это 200, 201, 204, 206, 301, 302, 303, 304, 307 или 308.

### 7.3.17 Split

Определяет вес действия в составе конфигурации разделений.

В приведенном ниже примере Angie передает 80% запросов вышестоящему `coffee-v1`, а оставшиеся 20% - `coffee-v2`:

```
splits:
- weight: 80
  action:
    pass: coffee-v1
- weight: 20
  action:
    pass: coffee-v2
```

Поле	Описание	Тип	Обязательно
<code>weight</code>	Вес действия. Значение должно попадать в диапазон 1..99. Сумма весов всех разделений должна быть равна 100.	<code>int</code>	Да
<code>action</code>	Действие, которое необходимо выполнить для запроса.	<code>:ref` :action`</code>	Да

### 7.3.18 Match

Определяет сопоставление между условиями и действием или разделениями.

В приведенном ниже примере Angie направляет запросы с путем «/coffee» в разные апстримы на основе значения cookie user:

- user=john -> coffee-future
- user=bob -> coffee-deprecated
- Если cookie не установлен или не равен ни john, ни bob, Angie перенаправляет запрос в coffee-stable

```
path: /coffee
matches:
- conditions:
  - cookie: user
    value: john
    action:
      pass: coffee-future
- conditions:
  - cookie: user
    value: bob
    action:
      pass: coffee-deprecated
action:
  pass: coffee-stable
```

В следующем примере Angie направляет запросы на основе значения встроенной переменной \$request\_method , которая представляет HTTP-метод запроса:

- все запросы POST -> coffee-post
- все прочие запросы -> coffee

```
path: /coffee
matches:
- conditions:
  - variable: $request_method
    value: POST
    action:
      pass: coffee-post
action:
  pass: coffee
```

Поле	Описание	Тип	Обязательно
conditions	Список условий. Должен включать по крайней мере одно условие.	<i>//condition</i>	Да
action	Действие, которое необходимо выполнить для запроса.	<i>Action</i>	Нет
splits	Конфигурация разбиений для разделения трафика. Должно быть указано не менее двух разделений.	<i>//split</i>	Нет

**Примечание**

Сопоставление должно использовать ровно одно из следующих значений: `action` или `splits`.

### 7.3.19 Condition

Определяет условие в сопоставлении.

Поле	Описание	Тип	Обязательно
<code>header</code>	Имя заголовка. Должно состоять из буквенно-цифровых символов или <code>-</code> .	<code>string</code>	Нет
<code>cookie</code>	Имя cookie. Должно состоять из буквенно-цифровых символов или <code>_</code> .	<code>string</code>	Нет
<code>argument</code>	Имя аргумента. Должно состоять из буквенно-цифровых символов или <code>_</code> .	<code>string</code>	Нет
<code>variable</code>	Имя переменной Angie. Должно начинаться с <code>\$</code> . См. список поддерживаемых переменных после таблицы.	<code>string</code>	Нет
<code>value</code>	Значение, которому должно соответствовать условие. Как определить значение, показано ниже в таблице.	<code>string</code>	Да

**Примечание**

Условие должно включать ровно одно из следующих значений: `header`, `cookie`, `argument` или `variable`.

Поддерживаемые переменные Angie: `$args`, `$http2`, `$https`, `$remote_addr`, `$remote_port`, `$query_string`, `$request`, `$request_body`, `$request_uri`, `$request_method`, `$scheme`.

Значение поддерживает два вида сопоставления:

- *Сравнение строк без учета регистра.* Например:
  - `john` - сопоставление без учета регистра, которое выполняется успешно для таких строк, как `john`, `John`, `JOHN`.
  - `!john` - отрицание соответствия без учета регистра для `john`, которое выполняется успешно для таких строк, как `bob`, `anything`, `"` (пустая строка).
- *Сопоставление с регулярным выражением.* Обратите внимание, что Angie поддерживает регулярные выражения, совместимые с языком программирования Perl (PCRE). Например:
  - `~^yes` - регулярное выражение с учетом регистра, которое соответствует любой строке, начинающейся с `yes`. Например: `yes`, `yes123`.
  - `!~^yes` - отрицание предыдущего регулярного выражения, которое успешно выполняется для строк типа `YES`, `Yes123`, `noyes`. (Механизм отрицания не является частью синтаксиса PCRE).
  - `~*no$` - регулярное выражение без учета регистра, которое соответствует любой строке, заканчивающейся на `no`. Например: `no`, `123no`, `123NO`.

### Примечание

Значение не должно содержать неэкранированных двойных кавычек (") и не должно заканчиваться неэкранированной обратной косой чертой (\). Например, следующие значения недопустимы: `some"value, somevalue\`.

## 7.3.20 ErrorPage

Определяет настраиваемый ответ для маршрута на случай, когда сервер апстрима отвечает кодом состояния ошибки (или его генерирует Angie). В качестве ответа может быть задано перенаправление или сохраненный ответ. Дополнительные сведения см. в описании директивы `error_page`.

```
path: /coffee
errorPages:
  - codes: [502, 503]
    redirect:
      code: 301
      url: https://angie.software
  - codes: [404]
    return:
      code: 200
      body: "Original resource not found, but success!"
```

Поле	Описание	Тип	Обязательно
<code>codes</code>	Список кодов состояния ошибки.	<code>[] int</code>	Да
<code>redirect</code>	Действие перенаправления для заданных кодов состояния.	<code>errorPage.Redirect</code>	Нет
<code>return</code>	Сохраненное ответное действие для заданных кодов состояния.	<code>errorPage.Return</code>	Нет

### Примечание

Страница с ошибкой должна содержать ровно одно из следующих значений: `return` или `redirect`.

## 7.3.21 ErrorPage.Redirect

Определяет перенаправление для `errorPage`.

В приведенном ниже примере Angie отвечает перенаправлением, когда ответ от сервера апстрима имеет код состояния 404.

```
codes: [404]
redirect:
  code: 301
  url: ${scheme}://cafe.example.com/error.html
```

Поле	Описание	Тип	Обязательно
код	Код состояния перенаправления. Допустимые значения: 301, 302, 307, 308. Значение по умолчанию - 301.	int	Нет
url	URL-адрес, на который будет перенаправлен запрос. Поддерживаемые переменные Angie: <code>\$(scheme)</code> и <code>\$(http_x_forwarded_proto)</code> . Переменные должны быть заключены в фигурные скобки. Например: <code>\$(scheme)</code> .	string	Да

### 7.3.22 ErrorPage.Return

Определяет сохраненный ответ для `errorPage`.

В приведенном ниже примере Angie выдает сохраненный ответ, когда ответ от сервера апстрима имеет код состояния 401 или 403.

```
codes: [401, 403]
return:
  code: 200
  type: application/json
  body: |
    {"msg": \"You don't have permission to do this\"}
  headers:
    - name: x-debug-original-statuses
      value: ${upstream\_status}
```

Поле	Описание	Тип	Обязательно
code	Код состояния ответа. По умолчанию используется код состояния исходного ответа.	int	Нет
type	МIME-тип ответа. Значение по умолчанию - <code>text/html</code> .	string	Нет
body	Тело ответа. Поддерживаемая переменная Angie: <code>\$(upstream_status)</code> . Переменные должны быть заключены в фигурные скобки. Например: <code>\$(upstream_status)</code> .	string	Да
headers	Настраиваемые заголовки ответа.	<i>errorPage.Return</i>	Нет

### 7.3.23 ErrorPage.Return.Header

Определяет HTTP-заголовок для сохраненного ответа у `errorPage`:

```
name: x-debug-original-statuses
value: ${upstream_status}
```

Поле	Описание	Тип	Обязательно
name	Имя заголовка.	string	Да
value	Значение заголовка. Поддерживаемая переменная Angie: <code>\$upstream_status</code> . Переменные должны быть заключены в фигурные скобки. Например: <code>\${upstream_status}</code> .	string	Нет

## 7.4 Использование VirtualServer и VirtualServerRoute

Для работы с ресурсами VirtualServer и VirtualServerRoute можно использовать обычные команды `kubectl`, аналогично ресурсам Ingress.

Например, следующая команда создает ресурс VirtualServer, определенный в `cafe-virtual-server.yaml` с именем `cafe`:

```
kubectl apply -f cafe-virtual-server.yaml

virtualserver.k8s.angie.software "cafe" created
```

Вы можете получить ресурс, выполнив:

```
kubectl get virtualserver cafe
```

NAME	STATE	HOST	IP	PORTS	AGE
cafe	Valid	cafe.example.com	12.13.23.123	[80,443]	3m

В `kubectl get` и подобных командах также можно использовать короткое имя `vs` вместо `virtualserver`.

Работать с ресурсами VirtualServerRoute можно аналогично. В командах `kubectl` используйте `virtualserverroute` или короткое имя `vsr`.

### 7.4.1 Использование фрагментов

Фрагменты позволяют вставлять элементы конфигурации Angie в различные контексты конфигурации Angie. В приведенном ниже примере мы используем фрагменты кода для настройки нескольких функций Angie на VirtualServer:

```
apiVersion: k8s.angie.software/v1
kind: VirtualServer
metadata:
  name: cafe
  namespace: cafe
spec:
  http-snippets: |
    limit_req_zone $binary_remote_addr zone=mylimit:10m rate=1r/s;
    proxy_cache_path /tmp keys_zone=one:10m;
  host: cafe.example.com
  tls:
    secret: cafe-secret
  server-snippets: |
    limit_req zone=mylimit burst=20;
  upstreams:
  - name: tea
    service: tea-svc
```

```

port: 80
- name: coffee
  service: coffee-svc
  port: 80
routes:
- path: /tea
  location-snippets: |
    proxy\_cache one;
    proxy\_cache\_valid 200 10m;
  action:
    pass: tea
- path: /coffee
  action:
    pass: coffee

```

Фрагменты предназначены для продвинутых пользователей Angie, которым требуется больше контроля над генерируемой конфигурацией Angie.

Однако из-за недостатков, описанных ниже, фрагменты по умолчанию отключены. Чтобы использовать фрагменты, задайте аргумент командной строки `enable-snippets`.

Недостатки использования фрагментов:

- *Сложность.* Чтобы использовать фрагменты, требуется:
  - Понимать примитивы конфигурации Angie и реализовать правильную конфигурацию Angie.
  - Понимать, как ANIC генерирует конфигурацию Angie, чтобы фрагмент не мешал другим функциям конфигурации.
- *Сниженная надежность.* Неправильный фрагмент делает конфигурацию Angie недействительной, что приведет к ошибке при перезагрузке. Это помешает применить какие-либо обновления конфигурации, включая обновления для других ресурсов `VirtualServer` и `VirtualServerRoute`, пока фрагмент не будет исправлен.
- *Последствия для безопасности.* Фрагменты предоставляют доступ к примитивам конфигурации Angie, и эти примитивы не проверяются самим ANIC. Например, через фрагмент можно настроить в Angie произвольную отправку сертификатов TLS и ключей, используемых для завершения TLS у ресурсов Ingress и `VirtualServer`.

Чтобы помочь отлавливать ошибки при использовании фрагментов, ANIC сообщает об ошибках перезагрузки конфигурации в журналах, а также в полях событий и состояния ресурсов `VirtualServer` и `VirtualServerRoute`.

#### **Примечание**

Пока конфигурация Angie содержит недопустимый фрагмент, Angie будет продолжать работать с последней допустимой конфигурацией.

## 7.4.2 Валидация

Для ресурсов `VirtualServer` и `VirtualServerRoute` доступны два типа валидации:

- *Структурная валидация* с помощью `kubectl` и сервера Kubernetes API.
- *Всесторонняя валидация* с помощью ANIC.

### Структурная валидация

Пользовательские определения ресурсов для `VirtualServer` и `VirtualServerRoute` включают структурную схему OpenAPI, которая описывает тип каждого поля этих ресурсов.

Если вы попытаетесь создать (или обновить) ресурс с нарушением структурной схемы (например, используете строковое значение для поля порта апстрима), `kubectl` и сервер Kubernetes API отклонят такой ресурс:

- Пример проверки `kubectl`:

```
kubectl apply -f cafe-virtual-server.yaml

error: error validating "cafe-virtual-server.yaml": error validating
data: ValidationError(VirtualServer.spec.upstreams[0].port): invalid
type for software.angie.k8s.v1.VirtualServer.spec.upstreams.port: got
"string", expected "integer"; if you choose to ignore these errors,
turn validation off with --validate=false
```

- Пример проверки сервера Kubernetes API:

```
kubectl apply -f cafe-virtual-server.yaml --validate=false

The VirtualServer "cafe" is invalid: []: Invalid value:
map[string]interface {}{ ... }: validation failure list:
spec.upstreams.port in body must be of type integer: "string"
```

Если ресурс не отклонен (т. е. не нарушает структурную схему), ANIC проверит его дополнительно.

### Всесторонняя валидация

ANIC проверяет поля ресурсов `VirtualServer` и `VirtualServerRoute`. Если ресурс недействителен, ANIC отклонит его: ресурс продолжит существовать в кластере, но ANIC будет его игнорировать.

Вы можете проверить, успешно ли ANIC применил конфигурацию для `VirtualServer`. Для нашего примера `VirtualServer` `cafe` мы можем запустить:

```
kubectl describe vs cafe

. . .
Events:
  Type      Reason          Age   From              Message
  ----      -
  Normal    AddedOrUpdated  16s   angie-ingress-controller Configuration for default/
↪cafe was added or updated
```

Обратите внимание, что раздел «События» (Events) включает событие `Normal` с причиной `AddedOrUpdated`, которое информирует нас о том, что конфигурация была успешно применена.

Если вы создадите недопустимый ресурс, ANIC отклонит его и выдаст событие `Rejected`. Например, если вы создадите `VirtualServer` `cafe` с двумя серверами апстрима с одинаковым именем `tea`, вы получите:

```
kubectl describe vs cafe
. . .
Events:
  Type          Reason      Age   From              Message
  ----          -
Warning        Rejected    12s   angie-ingress-controller VirtualServer default/cafe is
invalid and was rejected: spec.upstreams[1].name: Duplicate value: "tea"
```

Обратите внимание, что раздел «События» (Events) включает предупреждающее событие с указанием причины отклонения.

Кроме того, эта информация также доступна в поле `status` ресурса `VirtualServer`. Обратите внимание на раздел `Status VirtualServer`:

```
kubectl describe vs cafe
. . .
Status:
  External Endpoints:
    Ip:          12.13.23.123
    Ports:       [80,443]
  Message: VirtualServer default/cafe is invalid and was rejected: spec.upstreams[1].
name: Duplicate value: "tea"
  Reason:       Rejected
  State:        Invalid
```

ANIC проверяет ресурсы `VirtualServerRoute` аналогичным образом.

#### Примечание

Если вы внесете ошибку в существующий ресурс, ANIC отклонит его и удалит соответствующую конфигурацию из Angie.

## 7.5 Настройка с помощью ConfigMap

Вы можете дополнительно настроить конфигурацию Angie для ресурсов `VirtualServer` и `VirtualServerRoutes`, используя `ConfigMap`. Поддерживается большинство ключей `ConfigMap`, за следующими исключениями:

- `proxy-hide-headers`
- `proxy-pass-headers`
- `hsts`
- `hsts-max-age`
- `hsts-include-subdomains`
- `hsts-behind-proxy`
- `redirect-to-https`
- `ssl-redirect`

---

## Отчеты о состоянии ресурсов

---

### 8.1 Ресурсы Ingress

Ресурс Ingress может иметь состояние, куда входит адрес (IP-адрес или DNS-имя), через который становятся общедоступными узлы этого ресурса Ingress. Адрес можно видеть в выходных данных команды `kubectl get ingress` в столбце ADDRESS, как показано ниже:

```
$ kubectl get ingresses
```

NAME	HOSTS	ADDRESS	PORTS	AGE
myapp-ingress	myapp.example.com	12.13.23.123	80, 443	2m

ANIC должен быть сконфигурирован таким образом, чтобы сообщать о состоянии Ingress:

1. Используйте флаг командной строки `-report-ingress-status`.
2. Определите источник для внешнего адреса. Это может быть:
  - Определенный пользователем адрес, указанный в ключе ConfigMap `external-status-address`.
  - Служба типа LoadBalancer, настроенная с внешним IP-адресом или без него и указанная с помощью флага командной строки `-external-service`.

См. документацию по *ключам ConfigMap* и аргументам командной строки.

#### Примечание

При завершении работы ANIC не очищает статус ресурсов Ingress.

## 8.2 Ресурсы VirtualServer и VirtualServerRoute

Ресурс VirtualServer или VirtualServerRoute содержит поле состояния с информацией о состоянии ресурса и IP-адрес, через который становятся общедоступными узлы этого ресурса. Вы можете увидеть состояние в выходных данных команд `kubectl get virtualservers` или `kubectl get virtualserverroutes`, как показано ниже:

```
$ kubectl get virtualservers
```

NAME	STATE	HOST	IP	PORTS	AGE
myapp	Valid	myapp.example.com	12.13.23.123	[80,443]	34s

Чтобы просмотреть внешний адрес имени узла, связанный с ресурсом VirtualServer, используйте параметр `-o wide`:

```
$ kubectl get virtualservers -o wide
```

NAME	STATE	HOST	IP	EXTERNALHOSTNAME	AGE
↔ mysite	Valid	mysite.example.com		ae430f41a1a0042908655abcdefgijkl-	
↔ 12345678.eu-west-2.elb.amazonaws.com			[80,443]	106s	

**Примечание**

При наличии нескольких адресов отображается только первый из них.

Чтобы просмотреть дополнительные адреса или дополнительную информацию о *статусе* ресурса, используйте следующую команду:

```
$ kubectl describe virtualserver <NAME>
```

```

. . .
Status:
  External Endpoints:
    Ip:      12.13.23.123
    Ports:   [80,443]
  Message:  Configuration for myapp/myapp was added or updated
  Reason:   AddedOrUpdated
  State:    Valid

```

### 8.2.1 Спецификация состояния

Следующие поля отображаются как в статусе VirtualServer, так и в статусе VirtualServerRoute:

Поле	Описание	Тип
State	Текущее состояние ресурса. Возможные значения: <code>Valid</code> (допустимо), <code>Warning</code> (внимание) и <code>Invalid</code> (недопустимо). Дополнительные сведения см. в поле <code>message</code> .	string
Reason	Причина последнего обновления.	string
Message	Дополнительная информация о состоянии.	string
ExternalI	Список внешних конечных точек, для которых хосты ресурса являются общедоступными.	<code>ExternalEndpoint</code>

Следующее поле отображается только в состоянии VirtualServerRoute:

Поле	Описание	Тип
Reference	VirtualServer, который ссылается на этот VirtualServerRoute. Формат: пространство имен/имя.	string

### 8.2.2 ExternalEndpoint

Поле	Описание	Тип
IP	Внешний IP-адрес.	string
Hostname	Адрес имени узла внешнего балансировщика LoadBalancer.	string
Ports	Список внешних портов.	string

ANIC должен быть настроен таким образом, чтобы сообщать о состоянии VirtualServer или VirtualServerRoute.

Если вы хотите, чтобы ANIC сообщал о внешних конечных точках, определите источник для внешнего адреса. Это может быть:

- Определенный пользователем адрес, указанный в ключе ConfigMap `external-status-address`.
- Служба типа LoadBalancer, настроенная с внешним IP-адресом или без него и указанная с помощью флага командной строки `-external-service`.

См. документацию по *ключам ConfigMap* и аргументам командной строки.

Остальные поля будут включаться в отчет и без настроенного внешнего адреса.

#### Примечание

При завершении работы ANIC не очищает статус ресурсов VirtualServer и VirtualServerRoute.

## 8.3 Ресурсы Policy

Ресурс Policy включает в себя поле статуса с информацией о состоянии ресурса. Вы можете увидеть статус в выходных данных команды `kubectl get policy`, как показано ниже:

```
$ kubectl get policy
NAME          STATE    AGE
webapp-policy Valid    30s
```

Чтобы просмотреть дополнительные адреса или дополнительную информацию о *статусе* ресурса, используйте следующую команду:

```
$ kubectl describe policy <NAME>
. . .
Status:
  Message: Configuration for default/webapp-policy was added or updated
  Reason:  AddedOrUpdated
  State:   Valid
```

### 8.3.1 Спецификация состояния

В состоянии Policy отображаются следующие поля:

Поле	Описание	Тип
State	Текущее состояние ресурса. Возможные значения: <b>Valid</b> (допустимо) или <b>Invalid</b> (недопустимо). Дополнительные сведения см. в поле <b>message</b> .	string
Reason	Причина последнего обновления.	string
Message	Дополнительная информация о состоянии.	string

## 8.4 Ресурсы TransportServer

Ресурс TransportServer включает в себя поле состояния с информацией о состоянии ресурса. Вы можете увидеть его в выходных данных команды `kubectl get transportserver`, как показано ниже:

```
$ kubectl get transportserver
```

```
NAME      STATE    REASON           AGE
dns-tcp   Valid    AddedOrUpdated   47m
```

Чтобы просмотреть дополнительные адреса или дополнительную информацию о *статусе* ресурса, используйте следующую команду:

```
$ kubectl describe transportserver <NAME>
```

```
. . .
Status:
  Message: Configuration for default/dns-tcp was added or updated
  Reason:  AddedOrUpdated
  State:   Valid
```

### 8.4.1 Спецификация состояния

В состоянии TransportServer отображаются следующие поля:

Поле	Описание	Тип
State	Текущее состояние ресурса. Возможные значения: <b>Valid</b> (допустимо), <b>Warning</b> (внимание) и <b>Invalid</b> (недопустимо). Дополнительные сведения см. в поле <b>message</b> .	string
Reason	Причина последнего обновления.	string
Message	Дополнительная информация о состоянии.	string

---

## Просмотр состояния сервера

---

Angie поставляется со страницей статуса `Stub Status`, которая отображает основные метрики, и с веб-панелью мониторинга `Console Light`, которая отображает ключевые метрики балансировки нагрузки и производительности.

### 9.1 Доступ к `Stub Status`

Необходимые условия:

1. `Stub Status` должен быть включен по умолчанию. Убедитесь, что аргумент командной строки `angie-status` установлен в значении `true`.
2. `Stub Status` доступен на порту 8080 по умолчанию. Порт можно изменить с помощью аргумента командной строки `angie-status-port`. Если ваш порт отличается от 8080, измените команду `kubectl` проху ниже.

Чтобы открыть страницу статуса, выполните следующие действия:

1. Используйте команду `kubectl port-forward`, чтобы перенаправить соединения с порта 8080 на вашем локальном компьютере на порт 8080 пода ANIC (замените `<angie-ingress-pod>` на фактическое имя пода):

```
kubectl port-forward <angie-ingress-pod> 8080:8080 --namespace=angie-ingress
```

2. Откройте браузер по адресу [http://127.0.0.1:8080/stub\\_status](http://127.0.0.1:8080/stub_status).

Чтобы получить доступ к `stub status` извне (без `kubectl port-forward`), выполните следующие действия:

1. Настройте аргумент командной строки `-angie-status-allow-cidrs` с блоками IP/CIDR, для которых вы хотите разрешить доступ к статусу. По умолчанию доступ разрешен для `127.0.0.1,::1`.
2. Используйте IP/порт, через который доступен под ANIC, чтобы подключиться к странице статуса по пути `/stub_status`.

## 9.2 Доступ к веб-панели мониторинга Console Light

Необходимые условия:

1. Веб-панель мониторинга должна быть включена по умолчанию. Убедитесь, что аргумент командной строки `angie-status` установлен в значении `true`.
2. Веб-панель мониторинга доступна на порту 8080 по умолчанию. Порт можно изменить с помощью аргумента командной строки `angie-status-port`. Если ваш порт отличается от 8080, измените команду `kubectl proxy` ниже.

Чтобы открыть панель мониторинга, выполните следующие действия:

1. Используйте команду `kubectl port-forward`, чтобы перенаправить соединения с порта 8080 на вашем локальном компьютере на порт 8080 пода ANIC (замените `<angie-ingress-pod>` на фактическое имя пода):

```
kubectl port-forward <angie-ingress-pod> 8080:8080 --namespace=angie-ingress
```

2. Откройте браузер по адресу <http://127.0.0.1:8080/dashboard.html>.

Чтобы получить доступ к веб-панели мониторинга извне (без `kubectl port-forward`), выполните следующие действия:

1. Настройте аргумент командной строки `-angie-status-allow-cidrs` с блоками IP/CIDR, для которых вы хотите разрешить доступ к веб-панели мониторинга. По умолчанию доступ разрешен для `127.0.0.1,::1`.
2. Используйте IP/порт, через который доступен под ANIC, чтобы подключиться к веб-панели мониторинга по пути `/dashboard.html`.

### Примечание

API, которое используется веб-панелью мониторинга для получения метрик, также доступно по пути `/api`.

## ГЛАВА 10

## Ведение журнала

В ANIC вы можете посмотреть логи процесса Ingress Controller (процесса, который генерирует конфигурацию Angie и перезагружает Angie для ее применения), а также журнал доступа и журнал ошибок Angie. Все логи идут в стандартный вывод и стандартный поток ошибок процесса Ingress Controller. Чтобы посмотреть логи, вы можете выполнить команду `kubectl logs` для пода ANIC.

Например:

```
kubectl logs <angie-ingress-pod> -n angie-ingress
```

## 10.1 Логи процесса Ingress Controller

Логи процесса Ingress Controller можно настроить с помощью аргумента командной строки `-v`, который задает уровень детализации логов. Значение по умолчанию — 1, при этом значении записывается минимальное количество логов. Значение 3 полезно для устранения неполадок: вы сможете увидеть, как Ingress Controller получает обновления от Kubernetes API, генерирует конфигурацию Angie и перезагружает Angie.

## 10.2 Логи Angie

Angie включает два журнала:

- *Журнал доступа.* В этот журнал Angie записывает информацию о запросах клиентов сразу после обработки запроса. Журнал доступа настраивается через *ключи ConfigMap*: `log-format` для HTTP- и HTTPS-трафика и `stream-log-format` для сквозного трафика TCP, UDP и TLS. Дополнительно, вы можете отключить запись логов доступа с помощью ключа `access-log-off`.
- *Журнал ошибок.* В этот журнал Angie записывает информацию о возникших проблемах различного уровня критичности. Этот журнал настраивается через ключ `error-log-level` в *ConfigMap*. Чтобы включить отладочное логирование, установите значение `debug`, а также задайте аргумент командной строки `-angie-debug`. Angie будет запущен с отладочной версией бинарного файла `angie-debug`.

## ГЛАВА 11

---

### Пользовательские шаблоны

---

ANIC использует шаблоны для генерации конфигурации Angie для ресурсов Ingress, *ресурсов VirtualServer* и основного файла конфигурации Angie. Вы можете настроить собственные шаблоны и применить их с помощью *ConfigMap*.