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

ANIC

Руководство по установке

версия 0.6.0

ООО «Веб-Сервер»

янв. 14, 2025

Оглавление

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 внесен 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 развертывается 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	3.21	x86_64, arm64
Alt Linux	10	x86_64, arm64
Debian	11	x86_64, arm64

ГЛАВА 2

Установка

Вы можете установить ANIC с помощью манифестов или с помощью Helm.

Установка с помощью манифестов

Установка с помощью Helm

Установка с помощью манифестов подходит для более простых сценариев и когда требуется полный контроль над каждым аспектом конфигурации. Она подразумевает создание объектов Kubernetes с помощью YAML-файлов, которые вы применяете вручную. Эти манифесты должны быть заранее подготовлены, и все настройки делаются прямо в файлах.

Установка с помощью Helm подходит для более сложных проектов, требующих гибкости, легкости обновления и масштабирования. Helm — это пакетный менеджер для Kubernetes, который упрощает установку и управление приложениями. При использовании Helm вы запускаете установку с готовыми конфигурациями и переменными через шаблоны, которые могут быть легко модифицированы.

2.1 Поддерживаемые дистрибутивы

Название	Версии	Архитектуры
Alpine Linux	`3.21`	x86_64, ARM64
Debian	`11` "Bullseye"	x86_64, ARM64
ALT Linux	`10`	x86_64, ARM64

2.1.1 Установка с помощью Helm

Вступление

Эта диаграмма производит развертывание Angie Ingress Controller (ANIC) в кластере Kubernetes.

Предварительные требования

> **Примечание**
>
> Вся документация должна использоваться только с последней стабильной версией ANIC.

- Kubernetes 1.22+
- Helm 3.0+
- Скачайте образ ANIC и перенесите его в свой личный реестр.
- Обновите поле `controller.image.repository` файла `values.yaml` соответственно.

Пользовательские определения ресурсов

По умолчанию для ANIC требуется несколько пользовательских определений ресурсов (CRD), установленных в кластере. Клиент Helm установит эти определения. Если они не установлены, поды ANIC не будут `готовы`.

Скачивание диаграммы

Установить диаграммы для ANIC можно из нашего репозитория. За доступом обращайтесь на info@wbsrv.ru.

Если вы не используете пользовательские ресурсы, для которых требуются эти определения (что соответствует параметру `controller.enableCustomResources`, установленному как `false`), установку определений можно пропустить, указав `--skip-crds` в команде `helm install`.

Обновление определений

Чтобы обновить определения, скачайте исходные файлы диаграммы, как описано в разделе *Скачивание диаграммы*, а затем запустите:

```
kubectl apply -f crds/
```

> **Примечание**
>
> Возможно следующее предупреждение, которое можно игнорировать:
>
> ```
> Warning: kubectl apply should be used on resource created by either
> kubectl create --save-config or kubectl apply
>
> (Предупреждение: kubectl apply следует использовать для ресурса,
> созданного с помощью kubectl create --save-config или kubectl apply).
> ```

Удаление определений

Чтобы удалить определения, скачайте исходные файлы диаграммы, как описано в разделе *Скачивание диаграммы*, а затем запустите:

```
kubectl delete -f crds/
```

> **Примечание**
>
> Эта команда удалит все соответствующие пользовательские ресурсы в вашем кластере во всех пространствах имен. Убедитесь, что в кластере нет пользовательских ресурсов, которые вы хотите сохранить, и не запущены другие выпуски ANIC.

Управление диаграммой с помощью реестра

Установка диаграммы

Чтобы установить диаграмму с названием выпуска *my-release* (*my-release* - это название, которое вы выбираете сами, *myregistry.host.ru/angie-ingress* - необходимо изменить на путь в личном реестре):

```
helm repo add anic https://git.angie.software/api/packages/web-server/helm

helm install my-release anic/anic --set controller.image.repository=myregistry.host.
↪ru/angie-ingress
```

Это приведет к установке последней `пограничной` версии ANIC из реестра контейнеров.

Обновление диаграммы

Helm не обновляет определения во время обновления выпуска. Прежде чем обновлять выпуск, ознакомьтесь с разделом *Обновление определений*.

Чтобы обновить выпуск `my-release`:

```
helm upgrade my-release anic/anic -version 0.6.0
```

Удаление диаграммы

Чтобы удалить выпуск `my-release`:

```
helm uninstall my-release
```

Команда удаляет все компоненты Kubernetes, связанные с выпуском, и сам выпуск.

Удаление выпуска не приводит к удалению определений. Чтобы удалить определения, см. раздел *Удаление определений*.

Конфигурация

В следующей таблице перечислены настраиваемые параметры диаграммы Ingress Controller и их значения по умолчанию.

Параметр	Описание	По умолчанию
controller	Имя набора демонов или развертывания ANIC.	Создается автоматически
controller	Тип установки ANIC - deployment или daemonset (развертывание или набор демонов).	deployment
controller	Позволяет устанавливать *аннотации* для развертывания или набора демонов.	{}
controller	Развертывает ANIC для Angie PRO.	false
controller	Время ожидания в миллисекундах, в течение которого ANIC будет ожидать успешной перезагрузки Angie после изменения или при начальном запуске.	60000
controller	Позволяет подам ANIC использовать сетевое пространство имен хоста.	false
controller	Политика DNS для подов ANIC.	ClusterFirst
controller	Включает отладку для Angie. Требуется задать значение *error-log-level: debug* в ConfigMap через *controller.config.entries*.	false
controller	Уровень ведения журнала ANIC.	1
controller	Дайджест образа ANIC.	Нет
controller	Репозиторий образов ANIC.	myregistry.host.ru/angie-ingress
controller	Тег образа ANIC.	0.1.2
controller	Политика скачивания образа ANIC.	IfNotPresent
controller	Жизненный цикл подов ANIC.	{}
controller	Имя пользовательской ConfigMap, используемой ANIC. Если имя задано, то конфигурация по умолчанию игнорируется.	""
controller	Имя ConfigMap, используемой ANIC.	Создается автоматически
controller	Аннотации к ConfigMap в ANIC.	{}
controller	Записи в ConfigMap для настройки конфигурации Angie.	{}
controller	Список пользовательских портов, которые должны быть доступны в поде ANIC. Следует обычному синтаксису yaml Kubernetes для контейнерных портов.	[]
controller	Сертификат TLS в кодировке base64 для сервера HTTPS по умолчанию. **Примечание** Рекомендуется указать свой собственный сертификат. Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.	Нет

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
controller	Ключ TLS в кодировке base64 для сервера HTTPS по умолчанию. **Примечание** Рекомендуется указать свой собственный ключ. Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.	Нет
controller	Секрет с сертификатом TLS и ключом для сервера HTTPS по умолчанию. Значение должно соответствовать следующему формату: *<пространство имен>/<имя>*. Используется в качестве альтернативы указанию сертификата и ключа с помощью параметров *controller.defaultTLS.cert* и *controller.defaultTLS.key*. **Примечание** Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.	Нет
controller	Сертификат TLS в кодировке base64 для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Если параметр не задан, Angie прервет любую попытку установить TLS-соединение для таких узлов Ingress или VirtualServer.	Нет
controller	Ключ TLS в кодировке base64 для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Если параметр не задан, Angie прервет любую попытку установить TLS-соединение для таких узлов Ingress или VirtualServer.	Нет
controller	Секрет с сертификатом TLS и ключом для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Значение должно соответствовать следующему формату: *<пространство имен>/<имя>*. Используется в качестве альтернативы указанию сертификата и ключа с помощью параметров *controller.wildcardTLS.cert* и *controller.wildcardTLS.key*.	Нет
controller	Селектор узлов для назначения подов ANIC.	{}
controller	Период плавного завершения работы пода ANIC.	30
controller	Допуски подов ANIC.	[]
controller	Привязка подов ANIC.	{}
controller	Ограничения на распространение топологии подов ANIC.	{}
controller	Дополнительные переменные окружения, которые должны быть установлены на подах ANIC.	[]
controller	Тома подов ANIC.	[]
controller	Точки подключения томов подов ANIC.	[]
controller	Значение initContainers для подов ANIC.	[]
controller	Дополнительные контейнеры (например, сайдкар) для подов Ingress Controller.	[]
controller	Ресурсы подов ANIC.	requests: cpu=100m,memory=128M
controller	Количество реплик развертывания ANIC.	1
controller	Класс ANIC. Должен быть развернут ресурс IngressClass с именем, тождественным этому классу. В противном случае ANIC не запустится. ANIC обрабатывает только те ресурсы, которые принадлежат его классу, т. е. их ресурс поля "ingressClassName" тождествен классу. ANIC обрабатывает все ресурсы VirtualServer, VirtualServerRoute и TransportServer, которые не имеют поля "ingressClassName", во всех версиях Kubernetes.	angie

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
controller	Новым Ingress без указанного поля *"ingressClassName"* будет присвоен класс, указанный в *controller.ingressClass*.	false
controller	Разделенный запятыми список пространств имен, за ресурсами которых должен следить ANIC. По умолчанию ANIC отслеживает все пространства имен. Взаимоисключающие с *controller.watchNamespaceLabel*. Обратите внимание, что при настройке нескольких пространств имен с использованием опции Helm cli *--set* строка должна быть заключена в двойные кавычки, а запятые экранированы с помощью обратной косой черты - например, *--set controller.watchNamespace="default,anic"*.	""
controller	Настраивает в ANIC просмотр только пространств имен с меткой foo=bar. По умолчанию ANIC отслеживает все пространства имен. Взаимоисключающая с *controller.watchNamespace* настройка.	""
controller	Разделенный запятыми список пространств имен, за которыми Ingress Controller должен следить в поисках ресурсов типа Secret. Если этот параметр не настроен, ANIC отслеживает одни и те же пространства имен в поисках всех ресурсов. См. также *controller.watchNamespace* и *controller.watchNamespaceLabel*. Обратите внимание, что при настройке нескольких пространств имен с использованием опции Helm cli *--set* строка должна быть заключена в двойные кавычки, а запятые экранированы с помощью обратной косой черты - например, *--set controller.watchSecretNamespace="default,angie-ingress"*.	""
controller	Включает пользовательские ресурсы.	true
controller	Включает передачу данных по протоколу TLS на порту 443. Требуется *controller.enableCustomResources*.	false
controller	Включает автоматическое управление сертификатами x509 для ресурсов виртуального сервера с помощью cert-manager (cert-manager.io). Требуется *controller.enableCustomResources*.	false
controller	Включает интеграцию с ExternalDNS для настройки общедоступных записей DNS у ресурсов VirtualServer с использованием ExternalDNS. Требуется *controller.enableCustomResources*.	false
controller	Создает пользовательский ресурс GlobalConfiguration. Требуется *controller.enableCustomResources*.	false
controller	Спецификация GlobalConfiguration для определения параметров глобальной конфигурации ANIC.	{}
controller	Включает пользовательские фрагменты конфигурации Angie в ресурсах Ingress, VirtualServer, VirtualServerRoute и TransportServer.	false
controller	Добавляет местоположение "/angie-health" на сервер по умолчанию. Местоположение отвечает кодом статуса 200 на любой запрос. Это полезно для внешней проверки работоспособности ANIC.	false
controller	Задает URI местоположения состояния работоспособности на сервере по умолчанию. Требуется *controller.HealthStatus*.	"/angie-health"
controller	Включает в Angie API.	true
controller	Задает порт, на котором доступен Angie API.	8080
controller	Добавляет блоки IP или CIDR в список разрешенных для Angie API. Несколько IP или CIDR разделяются запятыми.	127.0.0.1,::1
controller	Класс приоритета подов ANIC.	Нет
controller	Создает сервис для предоставления доступа к подам ANIC.	true
controller	Тип сервиса, который необходимо создать для ANIC.	LoadBalancer
controller	Внешняя политика трафика сервиса. Значение Local сохраняет исходный IP-адрес клиента.	Local
controller	Аннотации сервиса ANIC.	{}
controller	Экстра-метки сервиса.	{}

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
controller	Статический IP-адрес для балансировщика нагрузки. Для *controller.service.type* должно быть установлено значение *LoadBalancer*. Поставщик облачных услуг должен поддерживать эту функцию.	""
controller	Список внешних IP-адресов для сервиса ANIC.	[]
controller	Диапазоны IP-адресов (CIDR), которым разрешен доступ к балансировщику нагрузки. Для *controller.service.type* должно быть установлено значение *LoadBalancer*. Поставщик облачных услуг должен поддерживать эту функцию.	[]
controller	Имя сервиса.	Создается автоматически
controller	Список пользовательских портов, которые будут доступны через сервис ANIC. Следует обычному синтаксису yaml Kubernetes для портов сервиса.	[]
controller	Включает HTTP-порт для сервиса ANIC.	true
controller	HTTP-порт сервиса ANIC.	80
controller	Пользовательский NodePort для HTTP-порта. Для *controller.service.type* должно быть установлено значение *NodePort*.	""
controller	Целевое значение HTTP-порта сервиса ANIC.	80
controller	Включает порт HTTPS для сервиса ANIC.	true
controller	HTTPS-порт сервиса ANIC.	443
controller	Пользовательский NodePort для HTTPS-порта. Для *controller.service.type* должно быть установлено значение *NodePort*.	""
controller	Целевой порт HTTPS-порта сервиса ANIC.	443
controller	Аннотации учетной записи сервиса ANIC.	{}
controller	Имя учетной записи сервиса подов ANIC. Используется для RBAC.	Создается автоматически
controller	Имя секретного файла, содержащего учетные данные реестра Docker. Секрет должен находиться в том же пространстве имен, что и выпуск Helm.	""
controller	Имя serviceMonitor.	Создается автоматически
controller	Создает пользовательский ресурс ServiceMonitor.	false
controller	Метки объектов Kubernetes для применения к объекту serviceMonitor.	""
controller	Набор меток, позволяющих выбирать конечные точки для serviceMonitor.	""
controller	Список конечных точек, разрешенных в составе этого serviceMonitor.	""
controller	Добавляет в поле адреса в статусе ресурсов Ingress внешний адрес Ingress Controller. Нужно также указать источник внешнего адреса через внешнюю службу через *controller.reportIngressStatus.ExternalService*, либо через *controller.reportIngressStatus.ingressLink*, либо через запись *external-status-address* в ConfigMap через *controller.config.entries*. **Примечание** Значение *controller.config.entries.external-status-address* имеет приоритет над остальными.	true

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
controller	Указывает имя сервиса с типом LoadBalancer, через который Ingress Controller будет доступен извне. Внешний адрес сервиса используется для отчетов о состоянии ресурсов Ingress, VirtualServer и VirtualServerRoute. Значение *controller.reportIngressStatus.enable* должно быть задано как *true*. Значение по умолчанию создается автоматически и включается, когда *controller.service.create* имеет значение *true*, а *controller.service.type* - значение *LoadBalancer*.	Создается автоматически
controller	Указывает имя ресурса IngressLink, который предоставляет доступ к подам ANIC через систему BIG-IP. IP-адрес системы BIG-IP используется для отчетов о состоянии ресурсов Ingress, VirtualServer и VirtualServerRoute. Значение *controller.reportIngressStatus.enable* должно быть задано как *true*.	""
controller	Включает выбор лидера, чтобы избежать ситуации, когда несколько реплик контроллера сообщают о состоянии ресурсов Ingress. Значение *controller.reportIngressStatus.enable* должно быть задано как *true*.	`true`
controller	Указывает имя ConfigMap в том же пространстве имен, что и контроллер, которое используется для блокировки выбора лидера. Значение controller.reportIngressStatus.enableLeaderElection должно быть задано как `true`.	Создается автоматически
controller	Аннотации к конфигурационной карте выборов лидера.	{}
controller	Аннотации пода ANIC.	{}
controller	Дополнительные экстра-метки для пода ANIC.	{}
controller	Включает конечную точку готовности `"/angie-ready"`. Конечная точка возвращает код успешного завершения, если Angie загрузил всю конфигурацию после запуска. Этим также настраивается проверка готовности для подов ANIC, которая использует конечную точку готовности.	`true`
controller	HTTP-порт для конечной точки готовности.	8081
controller	Число секунд с запуска пода ANIC до инициирования проверки готовности.	0
controller	Включает сбор метрик задержки для апстримов. Требуется `prometheus.create`.	`false`
controller	Задает минимальное количество секунд, в течение которых вновь созданный под должен прийти в готовое состояние без сбоя какого-либо из контейнеров, чтобы считаться доступным; документацию см. здесь.	0
controller	Включает HorizontalPodAutoscaling.	`false`
controller	Аннотации HorizontalPodAutoscaler для ANIC.	{}
controller	Минимальное число реплик для HPA.	1
controller	Максимальное число реплик для HPA.	3
controller	Целевой процент загрузки ЦП.	50
controller	Целевой процент использования памяти.	50
controller	Включает PodDisruptionBudget.	`false`
controller	Аннотации к бюджету сбоев пода ANIC.	{}
controller	Количество подов ANIC, которые должны быть доступны. Взаимоисключающая с `maxUnavailable`` настройка.	0
controller	Количество подов ANIC, которые могут быть недоступны. Взаимоисключающая с `minAvailable` настройка.	0
controller	Задает стратегию замены старых подов новыми. Документация по стратегии обновления развертывания и стратегии обновления набора демонов	{}
controller	В явной форме отключает прослушиватели IPV6 для узлов, которые не поддерживают стек IPV6.	`false`
controller	Настраивает корневую файловую систему как доступную только для чтения и добавляет тома для временных данных.	`false`
rbac.crea	Настраивает RBAC.	`true`
prometheu	Публикует метрики Angie в формате Prometheus.	`true`

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
prometheu	Настраивает порт для получения метрик.	`9113`
prometheu	Настраивает схему HTTP, используемую для подключений к конечной точке Prometheus.	`http`
prometheu	Пространство имен или имя TLS-секрета Kubernetes. Если секрет указан, он используется для защиты конечной точки Prometheus с помощью TLS-соединений.	`""`

2.1.2 Установка с помощью манифестов

Предварительные требования

Необходим доступ к Docker-образу в нашем репозитории:

```
anic.docker.angie.software/
```

Для текущей версии доступны следующие образы:

```
anic.docker.angie.software/anic:0.6.0-alpine
anic.docker.angie.software/anic:0.6.0-debian
anic.docker.angie.software/anic:0.6.0-altlinux
```

За доступом обращайтесь на info@wbsrv.ru.

Настройка RBAC

1. Создайте пространство имен и сервисный аккаунт для ANIC:

```
$ kubectl apply -f - <<EOF
apiVersion: v1
kind: Namespace
metadata:
 name: angie-ingress
---
apiVersion: v1
kind: ServiceAccount
metadata:
 name: angie-ingress
 namespace: angie-ingress
EOF
```

2. Создайте `ClusterRole` и `ClusterRoleBinding`:

Пример

```
$ kubectl apply -f - <<EOF
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: angie-ingress
rules:
- apiGroups:
  - discovery.k8s.io
  resources:
  - endpointslices
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - services
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - secrets
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - get
  - list
  - watch
  - update
  - create
- apiGroups:
  - ""
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
  - update
- apiGroups:
  - ""
  resources:
  - namespaces
  verbs:
  - get
  - list
```

```
  - watch
- apiGroups:
  - ""
  resources:
  - events
  verbs:
  - create
  - patch
  - list
- apiGroups:
  - coordination.k8s.io
  resources:
  - leases
  verbs:
  - get
  - list
  - watch
  - update
  - create
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - list
  - watch
  - get
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses/status
  verbs:
  - update
- apiGroups:
  - k8s.angie.software
  resources:
  - virtualservers
  - virtualserverroutes
  - globalconfigurations
  - transportservers
  - policies
  verbs:
  - list
  - watch
  - get
- apiGroups:
  - k8s.angie.software
  resources:
  - virtualservers/status
  - virtualserverroutes/status
  - policies/status
  - transportservers/status
  - dnsendpoints/status
  verbs:
  - update
- apiGroups:
  - networking.k8s.io
```

```
  resources:
  - ingressclasses
  verbs:
  - get
- apiGroups:
    - cis.f5.com
  resources:
    - ingresslinks
  verbs:
    - list
    - watch
    - get
- apiGroups:
    - cert-manager.io
  resources:
    - certificates
  verbs:
    - list
    - watch
    - get
    - update
    - create
    - delete
- apiGroups:
    - externaldns.angie.software
  resources:
    - dnsendpoints
  verbs:
    - list
    - watch
    - get
    - update
    - create
    - delete
- apiGroups:
  - externaldns.angie.software
  resources:
  - dnsendpoints/status
  verbs:
  - update
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: angie-ingress
subjects:
- kind: ServiceAccount
  name: angie-ingress
  namespace: angie-ingress
roleRef:
  kind: ClusterRole
  name: angie-ingress
  apiGroup: rbac.authorization.k8s.io
EOF
```

Создание ресурсов

3. Добавьте TLS-сертификат в настройки:

```
$ kubectl apply -f - <<EOF
apiVersion: v1
kind: Secret
metadata:
 name: default-server-secret
 namespace: angie-ingress
type: kubernetes.io/tls
data:
 tls.crt: Place TLS Certificate here in base64 format
 tls.key: Place TLS Key here in base64 format
EOF
```

4. Добавьте `ConfigMap` с настройками для Angie PRO:

```
$ kubectl apply -f - <<EOF
kind: ConfigMap
apiVersion: v1
metadata:
 name: angie-config
 namespace: angie-ingress
data:
EOF
```

5. Создайте `IngressClass`:

```
$ kubectl apply -f - <<EOF
apiVersion: networking.k8s.io/v1
kind: IngressClass
metadata:
 name: angie
spec:
 controller: angie/ingress-controller
EOF
```

6. Создайте пользовательские ресурсы `VirtualServer`, `VirtualServerRoute`, `TransportServer` и `Policy`:

Пример Virtual Server

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: virtualservers.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: VirtualServer
    listKind: VirtualServerList
    plural: virtualservers
    shortNames:
```

```
      - vs
    singular: virtualserver
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
        - description: Current state of the VirtualServer. If the resource has a␣
↪valid status, it means it has been validated and accepted by ANIC.
          jsonPath: .status.state
          name: State
          type: string
        - jsonPath: .spec.host
          name: Host
          type: string
        - jsonPath: .status.externalEndpoints[*].ip
          name: IP
          type: string
        - jsonPath: .status.externalEndpoints[*].hostname
          name: ExternalHostname
          priority: 1
          type: string
        - jsonPath: .status.externalEndpoints[*].ports
          name: Ports
          type: string
        - jsonPath: .metadata.creationTimestamp
          name: Age
          type: date
      name: v1
      schema:
        openAPIV3Schema:
          description: VirtualServer defines the VirtualServer resource.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: VirtualServerSpec is the spec of the VirtualServer␣
↪resource.
              type: object
              properties:
                dos:
                  type: string
                externalDNS:
                  description: ExternalDNS defines externaldns sub-resource of a␣
```

```
↪virtual server.
                  type: object
                  properties:
                    enable:
                      type: boolean
                    labels:
                      description: Labels stores labels defined for the Endpoint
                      type: object
                      additionalProperties:
                        type: string
                    providerSpecific:
                      description: ProviderSpecific stores provider specific␣
↪config
                      type: array
                      items:
                        description: ProviderSpecificProperty defines specific␣
↪property for using with ExternalDNS sub-resource.
                        type: object
                        properties:
                          name:
                            description: Name of the property
                            type: string
                          value:
                            description: Value of the property
                            type: string
                    recordTTL:
                      description: TTL for the record
                      type: integer
                      format: int64
                    recordType:
                      type: string
                host:
                  type: string
                http-snippets:
                  type: string
                ingressClassName:
                  type: string
                policies:
                  type: array
                  items:
                    description: PolicyReference references a policy by name and␣
↪an optional namespace.
                    type: object
                    properties:
                      name:
                        type: string
                      namespace:
                        type: string
                routes:
                  type: array
                  items:
                    description: Route defines a route.
                    type: object
                    properties:
                      action:
                        description: Action defines an action.
                        type: object
```

```
                        properties:
                          pass:
                            type: string
                          proxy:
                            description: ActionProxy defines a proxy in an␣
↪Action.
                            type: object
                            properties:
                              requestHeaders:
                                description: ProxyRequestHeaders defines the␣
↪request headers manipulation in an ActionProxy.
                                type: object
                                properties:
                                  pass:
                                    type: boolean
                                  set:
                                    type: array
                                    items:
                                      description: Header defines an HTTP Header.
                                      type: object
                                      properties:
                                        name:
                                          type: string
                                        value:
                                          type: string
                              responseHeaders:
                                description: ProxyResponseHeaders defines the␣
↪response headers manipulation in an ActionProxy.
                                type: object
                                properties:
                                  add:
                                    type: array
                                    items:
                                      description: AddHeader defines an HTTP␣
↪Header with an optional Always field to use with the add_header directive.
                                      type: object
                                      properties:
                                        always:
                                          type: boolean
                                        name:
                                          type: string
                                        value:
                                          type: string
                                  hide:
                                    type: array
                                    items:
                                      type: string
                                  ignore:
                                    type: array
                                    items:
                                      type: string
                                  pass:
                                    type: array
                                    items:
                                      type: string
                              rewritePath:
                                type: string
```

```
                              upstream:
                                type: string
                            redirect:
                              description: ActionRedirect defines a redirect in an␣
↪Action.
                              type: object
                              properties:
                                code:
                                  type: integer
                                url:
                                  type: string
                            return:
                              description: ActionReturn defines a return in an␣
↪Action.
                              type: object
                              properties:
                                body:
                                  type: string
                                code:
                                  type: integer
                                type:
                                  type: string
                        dos:
                          type: string
                        errorPages:
                          type: array
                          items:
                            description: ErrorPage defines an ErrorPage in a Route.
                            type: object
                            properties:
                              codes:
                                type: array
                                items:
                                  type: integer
                              redirect:
                                description: ErrorPageRedirect defines a redirect␣
↪for an ErrorPage.
                                type: object
                                properties:
                                  code:
                                    type: integer
                                  url:
                                    type: string
                              return:
                                description: ErrorPageReturn defines a return for␣
↪an ErrorPage.
                                type: object
                                properties:
                                  body:
                                    type: string
                                  code:
                                    type: integer
                                  headers:
                                    type: array
                                    items:
                                      description: Header defines an HTTP Header.
                                      type: object
```

```
                                    properties:
                                      name:
                                        type: string
                                      value:
                                        type: string
                                type:
                                  type: string
                        location-snippets:
                          type: string
                        matches:
                          type: array
                          items:
                            description: Match defines a match.
                            type: object
                            properties:
                              action:
                                description: Action defines an action.
                                type: object
                                properties:
                                  pass:
                                    type: string
                                  proxy:
                                    description: ActionProxy defines a proxy in an␣
↪Action.
                                    type: object
                                    properties:
                                      requestHeaders:
                                        description: ProxyRequestHeaders defines␣
↪the request headers manipulation in an ActionProxy.
                                        type: object
                                        properties:
                                          pass:
                                            type: boolean
                                          set:
                                            type: array
                                            items:
                                              description: Header defines an HTTP␣
↪Header.
                                              type: object
                                              properties:
                                                name:
                                                  type: string
                                                value:
                                                  type: string
                                      responseHeaders:
                                        description: ProxyResponseHeaders defines␣
↪the response headers manipulation in an ActionProxy.
                                        type: object
                                        properties:
                                          add:
                                            type: array
                                            items:
                                              description: AddHeader defines an␣
↪HTTP Header with an optional Always field to use with the add_header directive.
                                              type: object
                                              properties:
                                                always:
```

```
                                        type: boolean
                                    name:
                                      type: string
                                    value:
                                      type: string
                                hide:
                                  type: array
                                  items:
                                    type: string
                                ignore:
                                  type: array
                                  items:
                                    type: string
                                pass:
                                  type: array
                                  items:
                                    type: string
                            rewritePath:
                              type: string
                            upstream:
                              type: string
                        redirect:
                          description: ActionRedirect defines a redirect␣
↪in an Action.
                          type: object
                          properties:
                            code:
                              type: integer
                            url:
                              type: string
                        return:
                          description: ActionReturn defines a return in␣
↪an Action.
                          type: object
                          properties:
                            body:
                              type: string
                            code:
                              type: integer
                            type:
                              type: string
                    conditions:
                      type: array
                      items:
                        description: Condition defines a condition in a␣
↪MatchRule.
                        type: object
                        properties:
                          argument:
                            type: string
                          cookie:
                            type: string
                          header:
                            type: string
                          value:
                            type: string
                          variable:
```

```
                                    type: string
                            splits:
                              type: array
                              items:
                                description: Split defines a split.
                                type: object
                                properties:
                                  action:
                                    description: Action defines an action.
                                    type: object
                                    properties:
                                      pass:
                                        type: string
                                      proxy:
                                        description: ActionProxy defines a proxy␣
↪in an Action.
                                        type: object
                                        properties:
                                          requestHeaders:
                                            description: ProxyRequestHeaders␣
↪defines the request headers manipulation in an ActionProxy.
                                            type: object
                                            properties:
                                              pass:
                                                type: boolean
                                              set:
                                                type: array
                                                items:
                                                  description: Header defines an␣
↪HTTP Header.
                                                  type: object
                                                  properties:
                                                    name:
                                                      type: string
                                                    value:
                                                      type: string
                                          responseHeaders:
                                            description: ProxyResponseHeaders␣
↪defines the response headers manipulation in an ActionProxy.
                                            type: object
                                            properties:
                                              add:
                                                type: array
                                                items:
                                                  description: AddHeader defines␣
↪an HTTP Header with an optional Always field to use with the add_header␣
↪directive.
                                                  type: object
                                                  properties:
                                                    always:
                                                      type: boolean
                                                    name:
                                                      type: string
                                                    value:
                                                      type: string
                                              hide:
                                                type: array
```

```
                                        items:
                                          type: string
                                      ignore:
                                        type: array
                                        items:
                                          type: string
                                      pass:
                                        type: array
                                        items:
                                          type: string
                                  rewritePath:
                                    type: string
                                  upstream:
                                    type: string
                              redirect:
                                description: ActionRedirect defines a␣
↪redirect in an Action.
                                type: object
                                properties:
                                  code:
                                    type: integer
                                  url:
                                    type: string
                              return:
                                description: ActionReturn defines a␣
↪return in an Action.
                                type: object
                                properties:
                                  body:
                                    type: string
                                  code:
                                    type: integer
                                  type:
                                    type: string
                          weight:
                            type: integer
                path:
                  type: string
                policies:
                  type: array
                  items:
                    description: PolicyReference references a policy by␣
↪name and an optional namespace.
                    type: object
                    properties:
                      name:
                        type: string
                      namespace:
                        type: string
                route:
                  type: string
                splits:
                  type: array
                  items:
                    description: Split defines a split.
                    type: object
                    properties:
```

```
                            action:
                              description: Action defines an action.
                              type: object
                              properties:
                                pass:
                                  type: string
                                proxy:
                                  description: ActionProxy defines a proxy in an␣
↪Action.
                                  type: object
                                  properties:
                                    requestHeaders:
                                      description: ProxyRequestHeaders defines␣
↪the request headers manipulation in an ActionProxy.
                                      type: object
                                      properties:
                                        pass:
                                          type: boolean
                                        set:
                                          type: array
                                          items:
                                            description: Header defines an HTTP␣
↪Header.
                                            type: object
                                            properties:
                                              name:
                                                type: string
                                              value:
                                                type: string
                                    responseHeaders:
                                      description: ProxyResponseHeaders defines␣
↪the response headers manipulation in an ActionProxy.
                                      type: object
                                      properties:
                                        add:
                                          type: array
                                          items:
                                            description: AddHeader defines an␣
↪HTTP Header with an optional Always field to use with the add_header directive.
                                            type: object
                                            properties:
                                              always:
                                                type: boolean
                                              name:
                                                type: string
                                              value:
                                                type: string
                                        hide:
                                          type: array
                                          items:
                                            type: string
                                        ignore:
                                          type: array
                                          items:
                                            type: string
                                        pass:
                                          type: array
```

```
                                        items:
                                          type: string
                                    rewritePath:
                                      type: string
                                    upstream:
                                      type: string
                                redirect:
                                  description: ActionRedirect defines a redirect␣
↪in an Action.
                                  type: object
                                  properties:
                                    code:
                                      type: integer
                                    url:
                                      type: string
                                return:
                                  description: ActionReturn defines a return in␣
↪an Action.
                                  type: object
                                  properties:
                                    body:
                                      type: string
                                    code:
                                      type: integer
                                    type:
                                      type: string
                            weight:
                              type: integer
                server-snippets:
                  type: string
                tls:
                  description: TLS defines TLS configuration for a VirtualServer.
                  type: object
                  properties:
                    cert-manager:
                      description: CertManager defines a cert manager config for␣
↪a TLS.
                      type: object
                      properties:
                        cluster-issuer:
                          type: string
                        common-name:
                          type: string
                        duration:
                          type: string
                        issuer:
                          type: string
                        issuer-group:
                          type: string
                        issuer-kind:
                          type: string
                        renew-before:
                          type: string
                        usages:
                          type: string
                    redirect:
                      description: TLSRedirect defines a redirect for a TLS.
```

```
                        type: object
                        properties:
                          basedOn:
                            type: string
                          code:
                            type: integer
                          enable:
                            type: boolean
                      secret:
                        type: string
                  upstreams:
                    type: array
                    items:
                      description: Upstream defines an upstream.
                      type: object
                      properties:
                        buffer-size:
                          type: string
                        buffering:
                          type: boolean
                        buffers:
                          description: UpstreamBuffers defines Buffer␣
↪Configuration for an Upstream.
                          type: object
                          properties:
                            number:
                              type: integer
                            size:
                              type: string
                        client-max-body-size:
                          type: string
                        connect-timeout:
                          type: string
                        fail-timeout:
                          type: string
                        healthCheck:
                          description: HealthCheck defines the parameters for␣
↪active Upstream HealthChecks.
                          type: object
                          properties:
                            connect-timeout:
                              type: string
                            enable:
                              type: boolean
                            fails:
                              type: integer
                            grpcService:
                              type: string
                            grpcStatus:
                              type: integer
                            headers:
                              type: array
                              items:
                                description: Header defines an HTTP Header.
                                type: object
                                properties:
                                  name:
```

```
                                type: string
                              value:
                                type: string
                        interval:
                          type: string
                        jitter:
                          type: string
                        keepalive-time:
                          type: string
                        mandatory:
                          type: boolean
                        passes:
                          type: integer
                        path:
                          type: string
                        persistent:
                          type: boolean
                        port:
                          type: integer
                        read-timeout:
                          type: string
                        send-timeout:
                          type: string
                        statusMatch:
                          type: string
                        tls:
                          description: UpstreamTLS defines a TLS configuration␣
↪for an Upstream.
                          type: object
                          properties:
                            enable:
                              type: boolean
                      keepalive:
                        type: integer
                      lb-method:
                        type: string
                      max-conns:
                        type: integer
                      max-fails:
                        type: integer
                      name:
                        type: string
                      next-upstream:
                        type: string
                      next-upstream-timeout:
                        type: string
                      next-upstream-tries:
                        type: integer
                      ntlm:
                        type: boolean
                      port:
                        type: integer
                      queue:
                        description: UpstreamQueue defines Queue Configuration␣
↪for an Upstream.
                        type: object
                        properties:
```

```
                        size:
                          type: integer
                        timeout:
                          type: string
                    read-timeout:
                      type: string
                    send-timeout:
                      type: string
                    service:
                      type: string
                    sessionCookie:
                      description: SessionCookie defines the parameters for␣
↪session persistence.
                      type: object
                      properties:
                        domain:
                          type: string
                        enable:
                          type: boolean
                        expires:
                          type: string
                        httpOnly:
                          type: boolean
                        name:
                          type: string
                        path:
                          type: string
                        secure:
                          type: boolean
                    slow-start:
                      type: string
                    subselector:
                      type: object
                      additionalProperties:
                        type: string
                    tls:
                      description: UpstreamTLS defines a TLS configuration for␣
↪an Upstream.
                      type: object
                      properties:
                        enable:
                          type: boolean
                    type:
                      type: string
                    use-cluster-ip:
                      type: boolean
            status:
              description: VirtualServerStatus defines the status for the␣
↪VirtualServer resource.
              type: object
              properties:
                externalEndpoints:
                  type: array
                  items:
                    description: ExternalEndpoint defines the IP/ Hostname and␣
↪ports used to connect to this resource.
                    type: object
```

```
                    properties:
                      hostname:
                        type: string
                      ip:
                        type: string
                      ports:
                        type: string
                message:
                  type: string
                reason:
                  type: string
                state:
                  type: string
      served: true
      storage: true
      subresources:
        status: {}
EOF
```

Пример VirtualServerRoute

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: virtualserverroutes.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: VirtualServerRoute
    listKind: VirtualServerRouteList
    plural: virtualserverroutes
    shortNames:
      - vsr
    singular: virtualserverroute
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
        - description: Current state of the VirtualServerRoute. If the resource␣
↪has a valid status, it means it has been validated and accepted by ANIC.
          jsonPath: .status.state
          name: State
          type: string
        - jsonPath: .spec.host
          name: Host
          type: string
        - jsonPath: .status.externalEndpoints[*].ip
          name: IP
          type: string
        - jsonPath: .status.externalEndpoints[*].hostname
          name: ExternalHostname
          priority: 1
          type: string
```

```
        - jsonPath: .status.externalEndpoints[*].ports
          name: Ports
          type: string
        - jsonPath: .metadata.creationTimestamp
          name: Age
          type: date
      name: v1
      schema:
        openAPIV3Schema:
          description: VirtualServerRoute defines the VirtualServerRoute␣
↪resource.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: VirtualServerRouteSpec is the spec of the␣
↪VirtualServerRoute resource.
              type: object
              properties:
                host:
                  type: string
                ingressClassName:
                  type: string
                subroutes:
                  type: array
                  items:
                    description: Route defines a route.
                    type: object
                    properties:
                      action:
                        description: Action defines an action.
                        type: object
                        properties:
                          pass:
                            type: string
                          proxy:
                            description: ActionProxy defines a proxy in an␣
↪Action.
                            type: object
                            properties:
                              requestHeaders:
                                description: ProxyRequestHeaders defines the␣
```

```
↪request headers manipulation in an ActionProxy.
                                type: object
                                properties:
                                  pass:
                                    type: boolean
                                  set:
                                    type: array
                                    items:
                                      description: Header defines an HTTP Header.
                                      type: object
                                      properties:
                                        name:
                                          type: string
                                        value:
                                          type: string
                              responseHeaders:
                                description: ProxyResponseHeaders defines the␣
↪response headers manipulation in an ActionProxy.
                                type: object
                                properties:
                                  add:
                                    type: array
                                    items:
                                      description: AddHeader defines an HTTP␣
↪Header with an optional Always field to use with the add_header directive.
                                      type: object
                                      properties:
                                        always:
                                          type: boolean
                                        name:
                                          type: string
                                        value:
                                          type: string
                                  hide:
                                    type: array
                                    items:
                                      type: string
                                  ignore:
                                    type: array
                                    items:
                                      type: string
                                  pass:
                                    type: array
                                    items:
                                      type: string
                              rewritePath:
                                type: string
                              upstream:
                                type: string
                          redirect:
                            description: ActionRedirect defines a redirect in an␣
↪Action.
                            type: object
                            properties:
                              code:
                                type: integer
                              url:
```

```
                                type: string
                            return:
                              description: ActionReturn defines a return in an␣
↪Action.
                              type: object
                              properties:
                                body:
                                  type: string
                                code:
                                  type: integer
                                type:
                                  type: string
                        dos:
                          type: string
                        errorPages:
                          type: array
                          items:
                            description: ErrorPage defines an ErrorPage in a Route.
                            type: object
                            properties:
                              codes:
                                type: array
                                items:
                                  type: integer
                              redirect:
                                description: ErrorPageRedirect defines a redirect␣
↪for an ErrorPage.
                                type: object
                                properties:
                                  code:
                                    type: integer
                                  url:
                                    type: string
                              return:
                                description: ErrorPageReturn defines a return for␣
↪an ErrorPage.
                                type: object
                                properties:
                                  body:
                                    type: string
                                  code:
                                    type: integer
                                  headers:
                                    type: array
                                    items:
                                      description: Header defines an HTTP Header.
                                      type: object
                                      properties:
                                        name:
                                          type: string
                                        value:
                                          type: string
                                  type:
                                    type: string
                        location-snippets:
                          type: string
                        matches:
```

```
                        type: array
                        items:
                          description: Match defines a match.
                          type: object
                          properties:
                            action:
                              description: Action defines an action.
                              type: object
                              properties:
                                pass:
                                  type: string
                                proxy:
                                  description: ActionProxy defines a proxy in an␣
↪Action.
                                  type: object
                                  properties:
                                    requestHeaders:
                                      description: ProxyRequestHeaders defines␣
↪the request headers manipulation in an ActionProxy.
                                      type: object
                                      properties:
                                        pass:
                                          type: boolean
                                        set:
                                          type: array
                                          items:
                                            description: Header defines an HTTP␣
↪Header.
                                            type: object
                                            properties:
                                              name:
                                                type: string
                                              value:
                                                type: string
                                    responseHeaders:
                                      description: ProxyResponseHeaders defines␣
↪the response headers manipulation in an ActionProxy.
                                      type: object
                                      properties:
                                        add:
                                          type: array
                                          items:
                                            description: AddHeader defines an␣
↪HTTP Header with an optional Always field to use with the add_header directive.
                                            type: object
                                            properties:
                                              always:
                                                type: boolean
                                              name:
                                                type: string
                                              value:
                                                type: string
                                        hide:
                                          type: array
                                          items:
                                            type: string
                                        ignore:
```

```
                                        type: array
                                        items:
                                          type: string
                                    pass:
                                        type: array
                                        items:
                                          type: string
                                rewritePath:
                                    type: string
                                upstream:
                                    type: string
                            redirect:
                                description: ActionRedirect defines a redirect␣
↪in an Action.
                                type: object
                                properties:
                                    code:
                                        type: integer
                                    url:
                                        type: string
                            return:
                                description: ActionReturn defines a return in␣
↪an Action.
                                type: object
                                properties:
                                    body:
                                        type: string
                                    code:
                                        type: integer
                                    type:
                                        type: string
                        conditions:
                            type: array
                            items:
                                description: Condition defines a condition in a␣
↪MatchRule.
                                type: object
                                properties:
                                    argument:
                                        type: string
                                    cookie:
                                        type: string
                                    header:
                                        type: string
                                    value:
                                        type: string
                                    variable:
                                        type: string
                        splits:
                            type: array
                            items:
                                description: Split defines a split.
                                type: object
                                properties:
                                    action:
                                        description: Action defines an action.
                                        type: object
```

```
                                    properties:
                                      pass:
                                        type: string
                                      proxy:
                                        description: ActionProxy defines a proxy␣
↪in an Action.
                                        type: object
                                        properties:
                                          requestHeaders:
                                            description: ProxyRequestHeaders␣
↪defines the request headers manipulation in an ActionProxy.
                                            type: object
                                            properties:
                                              pass:
                                                type: boolean
                                              set:
                                                type: array
                                                items:
                                                  description: Header defines an␣
↪HTTP Header.
                                                  type: object
                                                  properties:
                                                    name:
                                                      type: string
                                                    value:
                                                      type: string
                                          responseHeaders:
                                            description: ProxyResponseHeaders␣
↪defines the response headers manipulation in an ActionProxy.
                                            type: object
                                            properties:
                                              add:
                                                type: array
                                                items:
                                                  description: AddHeader defines␣
↪an HTTP Header with an optional Always field to use with the add_header␣
↪directive.
                                                  type: object
                                                  properties:
                                                    always:
                                                      type: boolean
                                                    name:
                                                      type: string
                                                    value:
                                                      type: string
                                              hide:
                                                type: array
                                                items:
                                                  type: string
                                              ignore:
                                                type: array
                                                items:
                                                  type: string
                                              pass:
                                                type: array
                                                items:
                                                  type: string
```

```
                                        rewritePath:
                                          type: string
                                        upstream:
                                          type: string
                                    redirect:
                                      description: ActionRedirect defines a␣
↪redirect in an Action.
                                      type: object
                                      properties:
                                        code:
                                          type: integer
                                        url:
                                          type: string
                                    return:
                                      description: ActionReturn defines a␣
↪return in an Action.
                                      type: object
                                      properties:
                                        body:
                                          type: string
                                        code:
                                          type: integer
                                        type:
                                          type: string
                                weight:
                                  type: integer
                    path:
                      type: string
                    policies:
                      type: array
                      items:
                        description: PolicyReference references a policy by␣
↪name and an optional namespace.
                        type: object
                        properties:
                          name:
                            type: string
                          namespace:
                            type: string
                    route:
                      type: string
                    splits:
                      type: array
                      items:
                        description: Split defines a split.
                        type: object
                        properties:
                          action:
                            description: Action defines an action.
                            type: object
                            properties:
                              pass:
                                type: string
                              proxy:
                                description: ActionProxy defines a proxy in an␣
↪Action.
                                type: object
```

```
                                properties:
                                  requestHeaders:
                                    description: ProxyRequestHeaders defines␣
↪the request headers manipulation in an ActionProxy.
                                    type: object
                                    properties:
                                      pass:
                                        type: boolean
                                      set:
                                        type: array
                                        items:
                                          description: Header defines an HTTP␣
↪Header.
                                          type: object
                                          properties:
                                            name:
                                              type: string
                                            value:
                                              type: string
                                  responseHeaders:
                                    description: ProxyResponseHeaders defines␣
↪the response headers manipulation in an ActionProxy.
                                    type: object
                                    properties:
                                      add:
                                        type: array
                                        items:
                                          description: AddHeader defines an␣
↪HTTP Header with an optional Always field to use with the add_header directive.
                                          type: object
                                          properties:
                                            always:
                                              type: boolean
                                            name:
                                              type: string
                                            value:
                                              type: string
                                      hide:
                                        type: array
                                        items:
                                          type: string
                                      ignore:
                                        type: array
                                        items:
                                          type: string
                                      pass:
                                        type: array
                                        items:
                                          type: string
                                  rewritePath:
                                    type: string
                                  upstream:
                                    type: string
                              redirect:
                                description: ActionRedirect defines a redirect␣
↪in an Action.
                                type: object
```

```
                                properties:
                                  code:
                                    type: integer
                                  url:
                                    type: string
                              return:
                                description: ActionReturn defines a return in␣
↪an Action.
                                type: object
                                properties:
                                  body:
                                    type: string
                                  code:
                                    type: integer
                                  type:
                                    type: string
                          weight:
                            type: integer
              upstreams:
                type: array
                items:
                  description: Upstream defines an upstream.
                  type: object
                  properties:
                    buffer-size:
                      type: string
                    buffering:
                      type: boolean
                    buffers:
                      description: UpstreamBuffers defines Buffer␣
↪Configuration for an Upstream.
                      type: object
                      properties:
                        number:
                          type: integer
                        size:
                          type: string
                    client-max-body-size:
                      type: string
                    connect-timeout:
                      type: string
                    fail-timeout:
                      type: string
                    healthCheck:
                      description: HealthCheck defines the parameters for␣
↪active Upstream HealthChecks.
                      type: object
                      properties:
                        connect-timeout:
                          type: string
                        enable:
                          type: boolean
                        fails:
                          type: integer
                        grpcService:
                          type: string
                        grpcStatus:
```

```
                            type: integer
                        headers:
                            type: array
                            items:
                                description: Header defines an HTTP Header.
                                type: object
                                properties:
                                    name:
                                        type: string
                                    value:
                                        type: string
                        interval:
                            type: string
                        jitter:
                            type: string
                        keepalive-time:
                            type: string
                        mandatory:
                            type: boolean
                        passes:
                            type: integer
                        path:
                            type: string
                        persistent:
                            type: boolean
                        port:
                            type: integer
                        read-timeout:
                            type: string
                        send-timeout:
                            type: string
                        statusMatch:
                            type: string
                        tls:
                            description: UpstreamTLS defines a TLS configuration␣
↪for an Upstream.
                            type: object
                            properties:
                                enable:
                                    type: boolean
                    keepalive:
                        type: integer
                    lb-method:
                        type: string
                    max-conns:
                        type: integer
                    max-fails:
                        type: integer
                    name:
                        type: string
                    next-upstream:
                        type: string
                    next-upstream-timeout:
                        type: string
                    next-upstream-tries:
                        type: integer
                    ntlm:
```

```
                        type: boolean
                      port:
                        type: integer
                      queue:
                        description: UpstreamQueue defines Queue Configuration␣
↪for an Upstream.
                        type: object
                        properties:
                          size:
                            type: integer
                          timeout:
                            type: string
                      read-timeout:
                        type: string
                      send-timeout:
                        type: string
                      service:
                        type: string
                      sessionCookie:
                        description: SessionCookie defines the parameters for␣
↪session persistence.
                        type: object
                        properties:
                          domain:
                            type: string
                          enable:
                            type: boolean
                          expires:
                            type: string
                          httpOnly:
                            type: boolean
                          name:
                            type: string
                          path:
                            type: string
                          secure:
                            type: boolean
                      slow-start:
                        type: string
                      subselector:
                        type: object
                        additionalProperties:
                          type: string
                      tls:
                        description: UpstreamTLS defines a TLS configuration for␣
↪an Upstream.
                        type: object
                        properties:
                          enable:
                            type: boolean
                      type:
                        type: string
                      use-cluster-ip:
                        type: boolean
            status:
              description: VirtualServerRouteStatus defines the status for the␣
↪VirtualServerRoute resource.
```

```
              type: object
              properties:
                externalEndpoints:
                  type: array
                  items:
                    description: ExternalEndpoint defines the IP/ Hostname and␣
↪ports used to connect to this resource.
                    type: object
                    properties:
                      hostname:
                        type: string
                      ip:
                        type: string
                      ports:
                        type: string
                message:
                  type: string
                reason:
                  type: string
                referencedBy:
                  type: string
                state:
                  type: string
      served: true
      storage: true
      subresources:
        status: {}
EOF
```

Пример TransportServer

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: transportservers.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: TransportServer
    listKind: TransportServerList
    plural: transportservers
    shortNames:
      - ts
    singular: transportserver
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
        - description: Current state of the TransportServer. If the resource has␣
↪a valid status, it means it has been validated and accepted by ANIC.
          jsonPath: .status.state
          name: State
          type: string
```

```
        - jsonPath: .status.reason
          name: Reason
          type: string
        - jsonPath: .metadata.creationTimestamp
          name: Age
          type: date
      name: v1alpha1
      schema:
        openAPIV3Schema:
          description: TransportServer defines the TransportServer resource.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: TransportServerSpec is the spec of the␣
↪TransportServer resource.
              type: object
              properties:
                action:
                  description: Action defines an action.
                  type: object
                  properties:
                    pass:
                      type: string
                host:
                  type: string
                ingressClassName:
                  type: string
                listener:
                  description: TransportServerListener defines a listener for a␣
↪TransportServer.
                  type: object
                  properties:
                    name:
                      type: string
                    protocol:
                      type: string
                serverSnippets:
                  type: string
                sessionParameters:
                  description: SessionParameters defines session parameters.
                  type: object
```

```
                  properties:
                    timeout:
                      type: string
                streamSnippets:
                  type: string
                tls:
                  description: TLS defines TLS configuration for a␣
↪TransportServer.
                  type: object
                  properties:
                    secret:
                      type: string
                upstreamParameters:
                  description: UpstreamParameters defines parameters for an␣
↪upstream.
                  type: object
                  properties:
                    connectTimeout:
                      type: string
                    nextUpstream:
                      type: boolean
                    nextUpstreamTimeout:
                      type: string
                    nextUpstreamTries:
                      type: integer
                    udpRequests:
                      type: integer
                    udpResponses:
                      type: integer
                upstreams:
                  type: array
                  items:
                    description: Upstream defines an upstream.
                    type: object
                    properties:
                      failTimeout:
                        type: string
                      healthCheck:
                        description: HealthCheck defines the parameters for␣
↪active Upstream HealthChecks.
                        type: object
                        properties:
                          enable:
                            type: boolean
                          fails:
                            type: integer
                          interval:
                            type: string
                          jitter:
                            type: string
                          match:
                            description: Match defines the parameters of a␣
↪custom health check.
                            type: object
                            properties:
                              expect:
                                type: string
```

```
                                send:
                                  type: string
                            passes:
                              type: integer
                            port:
                              type: integer
                            timeout:
                              type: string
                        loadBalancingMethod:
                          type: string
                        maxConns:
                          type: integer
                        maxFails:
                          type: integer
                        name:
                          type: string
                        port:
                          type: integer
                        service:
                          type: string
              status:
                description: TransportServerStatus defines the status for the␣
↪TransportServer resource.
                type: object
                properties:
                  message:
                    type: string
                  reason:
                    type: string
                  state:
                    type: string
      served: true
      storage: true
      subresources:
        status: {}
  EOF
```

Пример Policy

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: policies.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: Policy
    listKind: PolicyList
    plural: policies
    shortNames:
      - pol
    singular: policy
```

```
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
        - description: Current state of the Policy. If the resource has a valid␣
↪status, it means it has been validated and accepted by ANIC.
          jsonPath: .status.state
          name: State
          type: string
        - jsonPath: .metadata.creationTimestamp
          name: Age
          type: date
      name: v1
      schema:
        openAPIV3Schema:
          description: Policy defines a Policy for VirtualServer and␣
↪VirtualServerRoute resources.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: PolicySpec is the spec of the Policy resource. The␣
↪spec includes multiple fields, where each field represents a different policy.␣
↪Only one policy (field) is allowed.
              type: object
              properties:
                accessControl:
                  description: AccessControl defines an access policy based on␣
↪the source IP of a request.
                  type: object
                  properties:
                    allow:
                      type: array
                      items:
                        type: string
                    deny:
                      type: array
                      items:
                        type: string
                basicAuth:
                  description: 'BasicAuth holds HTTP Basic authentication␣
↪configuration policy status: preview'
                  type: object
```

```
                properties:
                  realm:
                    type: string
                  secret:
                    type: string
              egressMTLS:
                description: EgressMTLS defines an Egress MTLS policy.
                type: object
                properties:
                  ciphers:
                    type: string
                  protocols:
                    type: string
                  serverName:
                    type: boolean
                  sessionReuse:
                    type: boolean
                  sslName:
                    type: string
                  tlsSecret:
                    type: string
                  trustedCertSecret:
                    type: string
                  verifyDepth:
                    type: integer
                  verifyServer:
                    type: boolean
              ingressClassName:
                type: string
              ingressMTLS:
                description: IngressMTLS defines an Ingress MTLS policy.
                type: object
                properties:
                  clientCertSecret:
                    type: string
                  crlFileName:
                    type: string
                  verifyClient:
                    type: string
                  verifyDepth:
                    type: integer
              jwt:
                description: JWT holds JWT authentication configuration.
                realm: string
                secret: string
                token: string
              oidc:
                description: OIDC defines an Open ID Connect policy.
                type: object
                properties:
                  clientID:
                    type: string
                  clientSecret:
                    type: string
                  authEndpoint:
                    type: string
                  jwksURI:
```

```
                    type: string
                  tokenEndpoint:
                    type: string
                  scope:
                    type: string
                  accessTokenEnable:
                    type: boolean
              rateLimit:
                description: RateLimit defines a rate limit policy.
                type: object
                properties:
                  burst:
                    type: integer
                  delay:
                    type: integer
                  dryRun:
                    type: boolean
                  key:
                    type: string
                  logLevel:
                    type: string
                  noDelay:
                    type: boolean
                  rate:
                    type: string
                  rejectCode:
                    type: integer
                  zoneSize:
                    type: string
          status:
            description: PolicyStatus is the status of the policy resource
            type: object
            properties:
              message:
                type: string
              reason:
                type: string
              state:
                type: string
    served: true
    storage: true
    subresources:
      status: {}
  - name: v1alpha1
    schema:
      openAPIV3Schema:
        description: Policy defines a Policy for VirtualServer and␣
↪VirtualServerRoute resources.
        type: object
        properties:
          apiVersion:
            description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
            type: string
```

```
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: PolicySpec is the spec of the Policy resource. The␣
↪spec includes multiple fields, where each field represents a different policy.␣
↪Only one policy (field) is allowed.
              type: object
              properties:
                accessControl:
                  description: AccessControl defines an access policy based on␣
↪the source IP of a request.
                  type: object
                  properties:
                    allow:
                      type: array
                      items:
                        type: string
                    deny:
                      type: array
                      items:
                        type: string
                egressMTLS:
                  description: EgressMTLS defines an Egress MTLS policy.
                  type: object
                  properties:
                    ciphers:
                      type: string
                    protocols:
                      type: string
                    serverName:
                      type: boolean
                    sessionReuse:
                      type: boolean
                    sslName:
                      type: string
                    tlsSecret:
                      type: string
                    trustedCertSecret:
                      type: string
                    verifyDepth:
                      type: integer
                    verifyServer:
                      type: boolean
                ingressMTLS:
                  description: IngressMTLS defines an Ingress MTLS policy.
                  type: object
                  properties:
                    clientCertSecret:
                      type: string
                    verifyClient:
```

```
                    type: string
                  verifyDepth:
                    type: integer
              jwt:
                description: JWT holds JWT authentication configuration.
                realm: string
                secret: string
                token: string
              rateLimit:
                description: RateLimit defines a rate limit policy.
                type: object
                properties:
                  burst:
                    type: integer
                  delay:
                    type: integer
                  dryRun:
                    type: boolean
                  key:
                    type: string
                  logLevel:
                    type: string
                  noDelay:
                    type: boolean
                  rate:
                    type: string
                  rejectCode:
                    type: integer
                  zoneSize:
                    type: string
    served: true
    storage: false
EOF
```

7. Если нужно использовать балансировщик нагрузки для TCP- и UDP-соединений, добавьте `GlobalConfiguration`:

Пример

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: globalconfigurations.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: GlobalConfiguration
    listKind: GlobalConfigurationList
    plural: globalconfigurations
    shortNames:
      - gc
    singular: globalconfiguration
```

```
  scope: Namespaced
  versions:
    - name: v1alpha1
      schema:
        openAPIV3Schema:
          description: GlobalConfiguration defines the GlobalConfiguration␣
↪resource.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this␣
↪representation of an object. Servers should convert recognized schemas to the␣
↪latest internal value, and may reject unrecognized values. More info: https://
↪git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST␣
↪resource this object represents. Servers may infer this from the endpoint the␣
↪client submits requests to. Cannot be updated. In CamelCase. More info: https:/
↪/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↪#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: GlobalConfigurationSpec is the spec of the␣
↪GlobalConfiguration resource.
              type: object
              properties:
                listeners:
                  type: array
                  items:
                    description: Listener defines a listener.
                    type: object
                    properties:
                      name:
                        type: string
                      port:
                        type: integer
                      protocol:
                        type: string
      served: true
      storage: true
EOF
```

Развертывание ANIC

8. Поддерживаются два варианта использования ANIC:

- `Deployment`: используйте этот тип развертывания, если планируете динамически изменять количество реплик ANIC.

Пример Deployment

```
$ kubectl apply -f - <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: angie-ingress
  namespace: angie-ingress
spec:
  replicas: 1
  selector:
    matchLabels:
      app: angie-ingress
  template:
    metadata:
      labels:
        app: angie-ingress
        app.kubernetes.io/name: angie-ingress
    #annotations:
      #prometheus.io/scrape: "true"
      #prometheus.io/port: "9113"
      #prometheus.io/scheme: http
    spec:
      serviceAccountName: angie-ingress
      automountServiceAccountToken: true
      securityContext:
        seccompProfile:
          type: RuntimeDefault
#         fsGroup: 101 #angie
        sysctls:
          - name: "net.ipv4.ip_unprivileged_port_start"
            value: "0"
#       volumes:
#       - name: angie-etc
#         emptyDir: {}
#       - name: angie-cache
#         emptyDir: {}
#       - name: angie-lib
#         emptyDir: {}
#       - name: angie-log
#         emptyDir: {}
      containers:
      - image: docker.angie.software/angie-ingress:latest
        imagePullPolicy: IfNotPresent
        name: angie-ingress
        ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
```

```
        - name: readiness-port
          containerPort: 8081
        - name: prometheus
          containerPort: 9113
        readinessProbe:
          httpGet:
            path: /angie-ready
            port: readiness-port
          periodSeconds: 1
        resources:
          requests:
            cpu: "100m"
            memory: "128Mi"
        #limits
        #  cpu: "1"
        #  memory: "1Gi"
        securityContext:
          allowPrivilegeEscalation: false
          runAsUser: 101 #angie
          runAsNonRoot: true
          capabilities:
            drop:
            - ALL
#         volumeMounts:
#         - mountPath: /etc/angie
#           name: angie-etc
#         - mountPath: /var/cache/angie
#           name: angie-cache
#         - mountPath: /var/lib/angie
#           name: angie-lib
#         - mountPath: /var/log/angie
#           name: angie-log
        env:
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        args:
          - -angie-configmaps=$(POD_NAMESPACE)/angie-config
        #- -default-server-tls-secret=$(POD_NAMESPACE)/default-server-secret
        #- -include-year
        #- -enable-cert-manager
        #- -enable-external-dns
        #- -v=3 # Enables extensive logging. Useful for troubleshooting.
        #- -report-ingress-status
        #- -external-service=angie-ingress
        #- -enable-prometheus-metrics
        #- -global-configuration=$(POD_NAMESPACE)/angie-configuration
EOF
```

- `DaemonSet`: используйте этот тип, если планируете развертывать ANIC на каждом узле кластера или подмножестве узлов.

Пример DaemonSet

```
$ kubectl apply -f - <<EOF
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: angie-ingress
  namespace: angie-ingress
spec:
  selector:
    matchLabels:
      app: angie-ingress
  template:
    metadata:
      labels:
        app: angie-ingress
        app.kubernetes.io/name: angie-ingress
    spec:
      serviceAccountName: angie-ingress
      automountServiceAccountToken: true
      securityContext:
        seccompProfile:
          type: RuntimeDefault
        sysctls:
          - name: "net.ipv4.ip_unprivileged_port_start"
            value: "0"
      containers:
      - image: docker.angie.software/angie-ingress:latest
        imagePullPolicy: IfNotPresent
        name: angie-ingress
        ports:
        - name: http
          containerPort: 80
          hostPort: 80
        - name: https
          containerPort: 443
          hostPort: 443
        - name: readiness-port
          containerPort: 8081
        - name: prometheus
          containerPort: 9113
        readinessProbe:
        httpGet:
          path: /angie-ready
          port: readiness-port
        periodSeconds: 1
        resources:
          requests:
            cpu: "100m"
            memory: "128Mi"
        env:
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: POD_NAME
          valueFrom:
            fieldRef:
```

```
              fieldPath: metadata.name
        args:
          - -angie-configmaps=$(POD_NAMESPACE)/angie-config
        #- -default-server-tls-secret=$(POD_NAMESPACE)/default-server-secret
        #- -include-year
        #- -v=3 # Enables extensive logging. Useful for troubleshooting.
        #- -report-ingress-status
        #- -external-service=angie-ingress
        #- -enable-prometheus-metrics
        #- -global-configuration=$(POD_NAMESPACE)/angie-configuration
EOF
```

ГЛАВА 3

Аргументы командной строки

ANIC поддерживает ряд аргументов командной строки. Способ указания этих аргументов зависит от того, как вы устанавливаете ANIC:

- Если вы используете *манифесты Kubernetes* (Deployment или DaemonSet) для установки ANIC, измените эти манифесты соответствующим образом, чтобы задать аргументы командной строки. См. документацию по установке с манифестами.
- Если вы используете *Helm* для установки ANIC, измените параметры диаграммы Helm, соответствующие аргументам командной строки. См. документацию по *установке с помощью Helm*.

Ниже в алфавитном порядке перечислены доступные аргументы командной строки:

3.1 -angie-configmaps <строка>

Ресурс ConfigMap для настройки конфигурации Angie. Если ConfigMap задан, но ANIC не может получить его из API Kubernetes, то ANIC не запустится.

Формат: `<пространство имен>/<имя>`

3.2 -angie-debug

Включает отладку для Angie. Использует бинарник angie-debug. Требуется 'error-log-level: debug' в ConfigMap.

3.3 -angie-reload-timeout <значение>

Время ожидания в миллисекундах, в течение которого ANIC будет ожидать успешной перезагрузки Angie после изменения конфигурации или при начальном запуске.

Значение по умолчанию - 60000.

3.4 -angie-status

Включает Angie stub_status.

По умолчанию `true`.

3.5 -angie-status-allow-cidrs <строка>

Добавляет блоки IP/CIDR в список разрешений для Angie stub_status.

Несколько IP или CIDR разделяются запятыми. (По умолчанию `127.0.0.1,::1`)

3.6 -angie-status-port <int>

Задает порт, на котором доступен Angie stub_status.

Формат: `[1024 - 65535]` (по умолчанию `8080`)

3.7 -angie-status-prometheus <bool>

Включает или отключает выдачу статистики Angie в формате Prometheus.

Формат: `false` или `true` (по умолчанию `true`)

3.8 -angie-status-prometheus-allow-cidrs

Добавляет блоки IP/CIDR в список разрешений для статистики Angie в формате Prometheus.

Несколько IP или CIDR разделяются запятыми. (По умолчанию `127.0.0.1,::1`)

3.9 -angie-status-prometheus-path <строка>

Позволяет менять путь для публикации статистики Angie в формате Prometheus.

По умолчанию используется `/p8s`.

3.10 -angie-status-prometheus-port <int>

Задает порт, на котором доступна статистика Angie в формате Prometheus.

Формат: `[1024 - 65535]` (по умолчанию `8083`)

3.11 -default-server-tls-secret <строка>

Секрет с сертификатом TLS и ключом для TLS-терминирования на сервере по умолчанию.

- Если значение не задано, используются сертификат и ключ в файле `/etc/angie/secrets/default`.
- Если `/etc/angie/secrets/default` не существует, ANIC настроит в Angie отклонение TLS-подключений к серверу по умолчанию.
- Если секрет установлен, но ANIC не может получить его из API Kubernetes, или же не установлен, и ANIC не удается прочитать файл `/etc/angie/secrets/default`, то ANIC не запустится.

Формат: `<пространство имен>/<имя>`

3.12 -disable-ipv6

Явно отключает прослушиватели IPV6 для узлов, которые не поддерживают стек IPV6.

По умолчанию `false`.

3.13 -enable-cert-manager

Включает автоматическое управление сертификатами x509 для ресурсов VirtualServer с помощью cert-manager (cert-manager.io).

Требует *-enable-custom-resources*.

3.14 -enable-custom-resources

Включает пользовательские ресурсы.

По умолчанию `true`.

3.15 -enable-external-dns

Включает интеграцию с ExternalDNS для настройки общедоступных записей DNS у ресурсов VirtualServer с использованием ExternalDNS.

Требует наличия *-enable-custom-resources*.

3.16 -enable-jwt

Включает функцию аутентификации JWT в ресурсах Policy.

По умолчанию `false`.

3.17 -enable-leader-election

Позволяет выбирать лидера, чтобы избежать ситуации, когда несколько реплик контроллера сообщают о статусе ресурсов Ingress, VirtualServer и VirtualServerRoute; сообщать о статусе будет только одна реплика. По умолчанию `true`.

См. флаг *-report-ingress-status*.

3.18 -enable-oidc

Включает функцию аутентификации по OpenID Connect в ресурсах Policy.

По умолчанию `false`.

3.19 -enable-prometheus-metrics

Позволяет публиковать метрики Angie в формате Prometheus.

3.20 -enable-service-insight

Публикует конечную точку Service Insight для ANIC.

3.21 -enable-snippets

Включает пользовательские фрагменты конфигурации Angie в ресурсах Ingress, VirtualServer, VirtualServerRoute и TransportServer.

По умолчанию `false`.

3.22 -enable-tls-passthrough

Включает сквозную передачу данных по протоколу TLS на порту 443.

Требует наличия *-enable-custom-resources*.

3.23 -external-service <строка>

Указывает имя сервиса с типом LoadBalancer, через который поды ANIC делаются доступными извне. Внешний адрес сервиса используется для отчетов о состоянии ресурсов Ingress, VirtualServer и VirtualServerRoute.

Только для ресурсов Ingress: требует наличия *-report-ingress-status*.

3.24 -global-configuration <строка>

Ресурс GlobalConfiguration для глобальной настройки ANIC.

Формат:`<пространство имен>/<имя>`

Требует наличия *-enable-custom-resources*.

3.25 -health-status

Добавляет местоположение "/angie-health" к серверу по умолчанию. Местоположение отвечает кодом статуса 200 на любой запрос.

Это полезно для внешней проверки работоспособности ANIC.

3.26 -health-status-uri <строка>

Задает URI местоположения проверки работоспособности на сервере по умолчанию. Требует наличия *-health-status*.

По умолчанию `/angie-health`.

3.27 -ingress-class <строка>

Класс ANIC.

Должен быть развернут соответствующий ресурс IngressClass с именем, равным классу. В противном случае ANIC не запустится. ANIC обрабатывает только те ресурсы, которые принадлежат его классу, т. е. имеют ресурс поля `ingressClassName`, равный классу.

ANIC обрабатывает все ресурсы, у которых нет поля `ingressClassName`.

По умолчанию `angie`.

3.28 -ingresslink <строка>

Указывает имя ресурса IngressLink, через который предоставляется доступ к подам ANIC через систему BIG-IP. IP-адрес системы BIG-IP используется для отчетов о состоянии ресурсов Ingress, VirtualServer и VirtualServerRoute.

Только для ресурсов Ingress: требует наличия *-report-ingress-status*.

3.29 -ingress-template-path <строка>

Путь к шаблону конфигурации Ingress Angie для ресурса Ingress. По умолчанию для Angie используется `angie.ingress.tmpl`.

3.30 -leader-election-lock-name <строка>

Указывает в том же пространстве имен, где находится контроллер, имя ConfigMap, используемое для блокировки при выборе лидера.

Требует наличия *-enable-leader-election*.

3.31 -main-template-path <строка>

Путь к основному шаблону конфигурации Angie.

- По умолчанию для Angie используется `angie.ingress.tmpl`.

3.32 -prometheus-metrics-listen-port <int>

Задает порт, на котором публикуются метрики Prometheus.

Формат: `[1024 - 65535]` (по умолчанию `9113`)

3.33 -prometheus-tls-secret <строка>

Секрет с сертификатом TLS и ключом для TLS-терминирования конечной точки метрик Prometheus.

- Если аргумент не задан, конечная точка Prometheus не будет использовать TLS-соединение.
- Если аргумент задан, но ANIC не может получить секрет из API Kubernetes, то ANIC не запустится.

3.34 -proxy <строка>

Задает использование прокси-сервера для подключения к API Kubernetes, запускаемого командой "kubectl proxy". **Только в целях тестирования**.

ANIC не запускает Angie и не записывает на диск никакие сгенерированные файлы конфигурации Angie.

3.35 -ready-status

Включает конечную точку готовности `/angie-ready`. Конечная точка возвращает код успеха, когда Angie загрузил всю конфигурацию после запуска.

По умолчанию `true`.

3.36 -ready-status-port

HTTP-порт для конечной точки готовности.

Формат: `[1024 - 65535]` (по умолчанию `8081`)

3.37 -report-ingress-status

Обновляет поле адреса в статусе ресурсов Ingress.

Требуется флаг *-external-service* или *-ingresslink*, либо ключ `external-status-address` в ConfigMap.

3.38 -service-insight-listen-port <int>

Задает порт, на котором публикуется Service Insight.

Формат: `[1024 - 65535]` (по умолчанию `9114`)

3.39 -service-insight-tls-secret <строка>

Секрет с сертификатом TLS и ключом для TLS-терминирования конечной точки Service Insight.

- Если аргумент не задан, конечная точка Service Insight не будет использовать TLS-соединение.
- Если аргумент задан, но ANIC не может получить секрет из API Kubernetes, то ANIC не запустится.

Формат: `<пространство имен>/<имя>`

3.40 -tls-passthrough-port <int>

Задает порт для сквозной передачи данных по протоколу TLS. Формат: `[1024 - 65535]` (по умолчанию `443`)

Требует включить *-enable-custom-resources*.

3.41 -transportserver-template-path <строка>

Путь к шаблону конфигурации TransportServer Angie для ресурса TransportServer.

- По умолчанию для Angie используется`angie.transportserver.tmpl`.

3.42 -v <значение>

Уровень детализации записи логов. Значение по умолчанию — `1`, при этом значении записывается минимальное количество логов. Значение `3` полезно для устранения неполадок.

3.43 -version

Выводит версию, хэш git-коммита и дату сборки, затем завершает работу.

3.44 -virtualserver-template-path <строка>

Путь к шаблону конфигурации VirtualServer Angie для ресурса VirtualServer.

- По умолчанию для Angie используется `angie.ingress.tmpl`.

3.45 -vmodule <значение>

Разделенный запятыми список параметров pattern=N для ведения журнала с фильтрацией файлов.

3.46 -watch-namespace <строка>

Разделенный запятыми список пространств имен, за ресурсами которых должен следить ANIC. По умолчанию ANIC отслеживает все пространства имен. Нельзя использовать вместе с "watch-namespace-label".

3.47 -watch-namespace-label <строка>

Настраивает в ANIC просмотр только пространств имен с меткой foo=bar. По умолчанию ANIC отслеживает все пространства имен. Нельзя использовать вместе с "watch-namespace".

3.48 -watch-secret-namespace <строка>

Разделенный запятыми список пространств имен, за которыми ANIC должен следить на предмет наличия секретов. Если этот параметр не настроен, ANIC отслеживает одни и те же пространства имен для всех ресурсов. См. также "watch-namespace" и "watch-namespace-label".

3.49 -wildcard-tls-secret <строка>

Секрет с сертификатом TLS и ключом для TLS-терминирования каждого узла Ingress или VirtualServer, для которого включено TLS-терминирование, но секрет не указан.

- Если аргумент не задан, для таких узлов Ingress и VirtualServer Angie прервет любую попытку установить TLS-соединение.
- Если аргумент задан, но ANIC не может получить секрет из API Kubernetes, то ANIC не запустится.

Формат: <пространство имен>/<имя>

ГЛАВА 4

Версии Angie Ingress Controller (ANIC)

4.1 2024

4.1.1 ANIC 0.6.0

Дата выпуска: 26.12.2024.

Добавления

Новые функции и настройки:

- Добавлены дополнительные поля в метриках Prometheus для ANIC:
  - `controller_pod`
  - `controller_namespace`
  - `controller_class`
- Добавлена возможность настраивать и осуществлять авторизацию клиента на основании результатов подзапроса.
- Версия Angie PRO обновлена до `1.8.1`.

4.1.2 ANIC 0.5.0

Дата выпуска: 30.09.2024.

Добавления

Новые функции и настройки:

- Добавлена возможность настраивать OIDC-авторизацию.
- Появилась возможность настройки JWT-авторизацию.
- Добавлена аннотация `angie.software/configmap` с параметром `namespace/config-name`, которая позволяет определить расширенный ConfigMap для заданного Ingress-ресурса.
- Директива staticLocations позволяет задавать расположение для раздачи статических файлов.
- Параметр *angie-status-prometheus-path* позволяет менять путь для статистики Angie в формате Prometheus.
- Добавлены параметры настройки SSL для ресурса VirtualServer:
  - `ssl_session_timeout`
  - `ssl_session_cache`
  - `ssl_session_tickets`
  - `ssl_stapling`
  - `ssl_stapling_verify`
- Для директивы ssl_prefer_server_ciphers теперь можно задавать значение `off`.
- Появилась возможность добавлять директиву s_map в конфигурацию Angie PRO, см. примеры настройки.
- Появилась поддержка директивы activeHealthProbes в Angie PRO.
- Версия Angie PRO обновлена до `1.7.0`.

Для облегчения процесса миграции сделаны следующие улучшения:

- Параметры типа `boolean` теперь могут принимать значения `true` или `false`, `t` или `f`, `on` или `off` и `1` или `0`.
- Параметр `proxy-buffers` директивы Upstream.buffers теперь может принимать только значение количества буферов `number`, без указания размера `size`. Если значение `size` не указано, то по умолчанию будет задано `8K`.

Исправления

- Исправлена ошибка при указании значения `HTTPS` для `backend-protocol`.
- Исправлено отображение IP в статусе `k8s`.
- Параметр `include-year` больше нельзя изменять, его значение теперь всегда `true`.

4.1.3 ANIC 0.4.0

Дата выпуска: 04.06.2024.

Добавления

Новые функции и настройки:

- Добавлены алиасы для следующих аннотаций:
  - `angie.software/force-ssl-redirect` — перенаправляет HTTP-запросы на HTTPS.
  - `angie.software/proxy-body-size` — устанавливает максимальный размер для тела запроса, которое может обработать проксируемый сервер.
  - `angie.software/proxy-buffer-size` — определяет размер буфера для чтения заголовков ответов от проксируемого сервера.
  - `angie.software/proxy-buffering` — включает или отключает буферизацию ответов от проксируемого сервера.
  - `angie.software/proxy-buffers-number` — определяет количество буферов, используемых для хранения ответа от проксируемого сервера.
  - `angie.software/proxy-max-temp-file-size` — задает максимальный размер временного файла, используемого для буферизации больших ответов.
- Директива server_tokens теперь может принимать строковые значения.
- Версия Angie PRO обновлена до `1.5.2`.

Исправления

- Исправлены требования к `resolver-addresses`.

4.1.4 ANIC 0.3.0

Дата выпуска: 02.03.2024.

Добавления

Новые функции и настройки:

- Добавлена аннотация `angie.software/force-ssl-redirect`, с помощью которой можно переводить небезопасные HTTP-запросы на защищенные HTTPS, что позволит исправить возможные ошибки при использовании SSL.

  > **Примечание**
  >
  > В связи с особенностями работы аннотации `force-ssl-redirect` рекомендуем использовать альтернативную настройку — `backend-protocol` со значением `HTTPS`.

- В Helm-чарты добавлены CRD (Common Resource Definitions), такие как Virtual Server, Virtual Server Route,TransportServer, Policies. Теперь пользователи Helm могут использовать эти определения в своих проектах, что значительно расширяет возможности настройки веб-сервера по сравнению со стандартным ресурсом Ingress.

Исправления

- Теперь ANIC запускается от имени пользователя `angie`.

4.2 2023

4.2.1 ANIC 0.2.0

Дата выпуска: 23.11.2023.

Добавления

- Добавлена поддержка консоли Console Light для мониторинга активности в реальном времени.
- Теперь можно отключать протокол `ipv6` с помощью *-disable-ipv6*.
- Добавлена точка подключения Prometheus `/ps8` для мониторинга статуса.
- Добавлена поддержка `sticky cookie` и `sticky route`.
- В параметры конфига добавлена переменная `$proxy_upstream_name` для использования в формате логов.

Исправления

- Исправлена ошибка с отсутствием прав доступа к `angie-syslog.sock`.

ГЛАВА 5

Права на интеллектуальную собственность