

---

# ANIC

ANIC

*Версия 1.0.0*

Руководство по установке и запуску

сент. 11, 2023

<b>1</b>	<b>Аннотация</b>	<b>1</b>
1.1	Общие сведения . . . . .	1
<b>2</b>	<b>Установка</b>	<b>2</b>
2.1	Поддерживаемые дистрибутивы . . . . .	2
2.2	Установка в K8S . . . . .	3
<b>3</b>	<b>Установка с помощью Helm</b>	<b>48</b>
3.1	Вступление . . . . .	48
3.2	Предварительные требования . . . . .	48
3.3	Пользовательские определения ресурсов . . . . .	48
3.4	Управление диаграммой с помощью реестра . . . . .	49
3.5	Конфигурация . . . . .	50
<b>4</b>	<b>Аргументы командной строки</b>	<b>56</b>
4.1	-enable-snippets . . . . .	56
4.2	-default-server-tls-secret <строка> . . . . .	56
4.3	-wildcard-tls-secret <строка> . . . . .	57
4.4	-enable-custom-resources . . . . .	57
4.5	-enable-leader-election . . . . .	57
4.6	-enable-tls-passthrough . . . . .	57
4.7	-tls-passthrough-port <int> . . . . .	57
4.8	-enable-cert-manager . . . . .	57
4.9	-enable-external-dns . . . . .	58
4.10	-external-service <строка> . . . . .	58
4.11	-ingresslink <строка> . . . . .	58
4.12	-global-configuration <строка> . . . . .	58
4.13	-health-status . . . . .	58
4.14	-health-status-uri <строка> . . . . .	58
4.15	-ingress-class <строка> . . . . .	59
4.16	-ingress-template-path <строка> . . . . .	59
4.17	-leader-election-lock-name <строка> . . . . .	59
4.18	-main-template-path <строка> . . . . .	59
4.19	-angie-configmaps <строка> . . . . .	59
4.20	-angie-debug . . . . .	59
4.21	-angie-reload-timeout <значение> . . . . .	60
4.22	-angie-status . . . . .	60
4.23	-angie-status-allow-cidrs <строка> . . . . .	60
4.24	-angie-status-port <int> . . . . .	60
4.25	-proxy <строка> . . . . .	60
4.26	-report-ingress-status . . . . .	60

4.27	-transportserver-template-path <строка>	60
4.28	-v<значение>	61
4.29	-version	61
4.30	-virtualserver-template-path <строка>	61
4.31	-vmodule <значение>	61
4.32	-watch-namespace <строка>	61
4.33	-watch-namespace-label <строка>	61
4.34	-watch-secret-namespace <строка>	61
4.35	-enable-prometheus-metrics	61
4.36	-prometheus-metrics-listen-port <int>	62
4.37	-prometheus-tls-secret <строка>	62
4.38	-enable-service-insight	62
4.39	-service-insight-listen-port <int>	62
4.40	-service-insight-tls-secret <строка>	62
4.41	-ready-status	62
4.42	-ready-status-port	63
4.43	-disable-ipv6	63

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.

## 1.1 Общие сведения

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

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

ANIC использует широкий набор функций Ingress:

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

## 2.1 Поддерживаемые дистрибутивы

### 2.1.1 Alpine

Версия	Платформа
3.17	x86_64, aarch64

### 2.1.2 ALT

Версия	Платформа
10	x86_64, aarch64

### 2.1.3 Debian

Версия	Платформа
11 "Bullseye"	x86_64, aarch64

## 2.2 Установка в K8S

### 2.2.1 Предварительные требования

- Необходим доступ к Docker-образу:

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

За доступом обращайтесь на [info@wbsrv.ru](mailto:info@wbsrv.ru).

### 2.2.2 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:
  - ""
```

(continues on next page)

(продолжение с предыдущей страницы)

```
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:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
- 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
```

(continues on next page)



(продолжение с предыдущей страницы)

```

- 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

```

### 2.2.3 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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
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 the Ingress
↪Controller.
        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
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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
                    type: object
                    properties:
                        code:
                            type: integer
                        url:
                            type: string
            return:
                description: ErrorPageReturn defines a return for
                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:

```

→for an ErrorPage.

→an ErrorPage.

(continues on next page)

(продолжение с предыдущей страницы)

```

    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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

                                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

```

(continues on next page)



(продолжение с предыдущей страницы)

```

proxy:
  description: ActionProxy defines a proxy
  type: object
  properties:
    requestHeaders:
      description: ProxyRequestHeaders
      type: object
      properties:
        pass:
          type: boolean
        set:
          type: array
          items:
            description: Header defines an
            type: object
            properties:
              name:
                type: string
              value:
                type: string
        responseHeaders:
          description: ProxyResponseHeaders
          type: object
          properties:
            add:
              type: array
              items:
                description: AddHeader defines
                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

```

→in an Action.

→defines the request headers manipulation in an ActionProxy.

→HTTP Header.

→defines the response headers manipulation in an ActionProxy.

→an HTTP Header with an optional Always field to use with the add\_header directive.

(continues on next page)

(продолжение с предыдущей страницы)

```

                                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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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 the Ingress
↳Controller.
        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

```

(continues on next page)



(продолжение с предыдущей страницы)

```

    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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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.

```

(continues on next page)

(продолжение с предыдущей страницы)

```

type: object
properties:
  pass:
    type: string
  proxy:
    description: ActionProxy defines a proxy
    type: object
    properties:
      requestHeaders:
        description: ProxyRequestHeaders
        type: object
        properties:
          pass:
            type: boolean
          set:
            type: array
            items:
              description: Header defines an
              type: object
              properties:
                name:
                  type: string
                value:
                  type: string
            responseHeaders:
              description: ProxyResponseHeaders
              type: object
              properties:
                add:
                  type: array
                  items:
                    description: AddHeader defines
                    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

```

→in an Action.

→defines the request headers manipulation in an ActionProxy.

→HTTP Header.

→defines the response headers manipulation in an ActionProxy.

→an HTTP Header with an optional Always field to use with the add\_header directive.

(continues on next page)

(продолжение с предыдущей страницы)

```

        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:

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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

```

(continues on next page)



(продолжение с предыдущей страницы)

```

                                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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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.
  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:

```

→for an Upstream.

(continues on next page)

(продолжение с предыдущей страницы)

```

    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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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 the Ingress
→Controller.
      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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

    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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

        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 the Ingress Controller.
        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.

```

(continues on next page)



(продолжение с предыдущей страницы)

```

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
verifyClient:
  type: string
verifyDepth:
  type: integer
jwt:
  description: JWTAuth holds JWT authentication configuration.
  type: object
  properties:
    jwksURI:
      type: string
    keyCache:
      type: string
    realm:
      type: string
    secret:
      type: string
    token:
      type: string
oidc:
  description: OIDC defines an Open ID Connect policy.
  type: object
  properties:
    accessTokenEnable:
      type: boolean
    authEndpoint:
      type: string
    authExtraArgs:
      type: array
      items:
        type: string
    clientID:
      type: string
    clientSecret:
      type: string
    jwksURI:
      type: string
    redirectURI:
      type: string
    scope:
      type: string
    tokenEndpoint:
      type: string
    zoneSyncLeeway:
      type: integer
rateLimit:
  description: RateLimit defines a rate limit policy.
  type: object
  properties:
    burst:
      type: integer
    delay:
      type: integer
    dryRun:
      type: boolean
    key:
      type: string
```

(continues on next page)

(продолжение с предыдущей страницы)

```

    logLevel:
      type: string
    noDelay:
      type: boolean
    rate:
      type: string
    rejectCode:
      type: integer
    zoneSize:
      type: string
  waf:
    description: WAF defines an WAF policy.
    type: object
    properties:
      apBundle:
        type: string
      apPolicy:
        type: string
      enable:
        type: boolean
      securityLog:
        description: SecurityLog defines the security log of a WAF
        type: object
        properties:
          apLogConf:
            type: string
          enable:
            type: boolean
          logDest:
            type: string
      securityLogs:
        type: array
        items:
          description: SecurityLog defines the security log of a
        type: object
        properties:
          apLogConf:
            type: string
          enable:
            type: boolean
          logDest:
            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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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: JWTAuth holds JWT authentication configuration.
    type: object
    properties:
      realm:
        type: string
      secret:
        type: string
      token:
        type: 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
```

(continues on next page)

(продолжение с предыдущей страницы)

```

        properties:
            name:
                type: string
            port:
                type: integer
            protocol:
                type: string
    served: true
    storage: true
EOF

```

## 2.2.4 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

```

(continues on next page)

(продолжение с предыдущей страницы)

```

#     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:

```

(continues on next page)



(продолжение с предыдущей страницы)

```

- --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 разворачивать 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

```

(continues on next page)

(продолжение с предыдущей страницы)

```
    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

## 3.1 Вступление

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

## 3.2 Предварительные требования

---

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

---

- Версия Kubernetes  $\geq 1.22$
- Helm 3.0+
- Скачайте образ Angie Ingress Controller (ANIC) и перенесите его в свой личный реестр
- Обновите поле `controller.image.repository` файла `values.yaml` соответственно.

## 3.3 Пользовательские определения ресурсов

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

### 3.3.1 Скачивание диаграммы

Установить диаграммы для ANIC можно из *нашего репозитория* <<https://git.angie.software/web-server/anic-helm-charts/>>. За доступом обращайтесь на [info@wbsrv.ru](mailto: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`.

### 3.3.2 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`).

### 3.3.3 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 Ingress Controller.

## 3.4 Управление диаграммой с помощью реестра

### 3.4.1 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 Ingress Controller из реестра контейнеров.

### 3.4.2 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 0.1.12
```

### 3.4.3 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*.

## 3.5 Конфигурация

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

Параметр	Описание	По умолчанию
<code>controller</code>	Имя набора демонов или развертывания Ingress Controller.	Создается автоматически
<code>controller</code>	Тип установки Ingress Controller - <code>deployment</code> или <code>daemonset</code> (развертывание или набор демонов).	<code>deployment</code>
<code>controller</code>	Позволяет устанавливать <i>аннотации</i> для развертывания или набора демонов.	<code>{}</code>
<code>controller</code>	Развертывает Ingress Controller для Angie PRO.	<code>false</code>
<code>controller</code>	Время ожидания в миллисекундах, в течение которого Ingress Controller будет ожидать успешной перезагрузки Angie после изменения или при начальном запуске.	<code>60000</code>
<code>controller</code>	Позволяет подам Ingress Controller использовать сетевое пространство имен хоста.	<code>false</code>
<code>controller</code>	Политика DNS для подов Ingress Controller.	<code>ClusterFirst</code>
<code>controller</code>	Включает отладку для Angie. Требуется задать значение <code>error-log-level: debug</code> в ConfigMap через <code>controller.config.entries</code> .	<code>false</code>
<code>controller</code>	Уровень ведения журнала Ingress Controller.	<code>1</code>
<code>controller</code>	Дайджест образа Ingress Controller.	<code>Her</code>
<code>controller</code>	Репозиторий образов Ingress Controller.	<code>myregistry.host.ru/angie-ingress</code>
<code>controller</code>	Тег образа Ingress Controller.	<code>0.1.2</code>
<code>controller</code>	Политика скачивания образа Ingress Controller.	<code>IfNotPresent</code>
<code>controller</code>	Жизненный цикл подов Ingress Controller.	<code>{}</code>
<code>controller</code>	Имя пользовательской ConfigMap, используемой Ingress Controller. Если имя задано, то конфигурация по умолчанию игнорируется.	<code>&lt;&lt;&gt;</code>

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Пара-метр	Описание	По умол-чанию
<i>controller</i>	Имя ConfigMap, используемой Ingress Controller.	Созда-ется авто-мати-чески
<i>controller</i>	Аннотации к ConfigMap в Ingress Controller.	{}
<i>controller</i>	Записи в ConfigMap для настройки конфигурации Angie.	{}
<i>controller</i>	Список пользовательских портов, которые должны быть доступны в поде Ingress Controller. Следует обычному синтаксису yaml Kubernetes для кон-тейнерных портов.	[]
<i>controller</i>	Сертификат TLS в кодировке base64 для сервера HTTPS по умолчанию.	Нет
<p><b>Примечание:</b> Рекомендуется указать свой собственный сертификат. Аль-тернативное решение: полный пропуск секрета сервера по умолчанию при-ведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p>		
<i>controller</i>	Ключ TLS в кодировке base64 для сервера HTTPS по умолчанию.	Нет
<p><b>Примечание:</b> Рекомендуется указать свой собственный ключ. Альтерна-тивное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p>		
<i>controller</i>	Секрет с сертификатом TLS и ключом для сервера HTTPS по умол-чанию. Значение должно соответствовать следующему формату: <i>&lt;про-странство имен&gt;/&lt;имя&gt;</i> . Используется в качестве альтернативы указа-нию сертификата и ключа с помощью параметров <i>controller.defaultTLS.cert</i> и <i>controller.defaultTLS.key</i> .	Нет
<p><b>Примечание:</b> Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p>		
<i>controller</i>	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 та-ких узлов Ingress или VirtualServer.	Нет
<i>controller</i>	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
<i>controller</i>	Секрет с сертификатом TLS и ключом для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Значение долж-но соответствовать следующему формату: <i>&lt;пространство имен&gt;/&lt;имя&gt;</i> . Используется в качестве альтернативы указанию сертификата и ключа с по-мощью параметров <i>controller.wildcardTLS.cert</i> и <i>controller.wildcardTLS.key</i> .	Нет
<i>controller</i>	Селектор узлов для назначения подов Ingress Controller.	{}
<i>controller</i>	Период плавного завершения работы пода Ingress Controller.	30
<i>controller</i>	Допуски подов Ingress Controller.	[]
<i>controller</i>	Привязка подов Ingress Controller.	{}

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controller</i>	Ограничения на распространение топологии подов Ingress Controller.	{}
<i>controller</i>	Дополнительные переменные окружения, которые должны быть установлены на подах Ingress Controller.	[]
<i>controller</i>	Тома подов Ingress Controller.	[]
<i>controller</i>	Точки подключения томов подов Ingress Controller.	[]
<i>controller</i>	Значение <code>initContainers</code> для подов Ingress Controller.	[]
<i>controller</i>	Дополнительные контейнеры (например, сайдкап) для подов Ingress Controller.	[]
<i>controller</i>	Ресурсы подов Ingress Controller.	requests: cpu=100m,memory=128M
<i>controller</i>	Количество реплик развертывания Ingress Controller.	1
<i>controller</i>	Класс Ingress Controller. Должен быть развернут ресурс <code>IngressClass</code> с именем, тождественным этому классу. В противном случае Ingress Controller не запустится. Ingress Controller обрабатывает только те ресурсы, которые принадлежат его классу, т. е. их ресурс поля <code>ingressClassName</code> тождествен классу. Ingress Controller обрабатывает все ресурсы <code>VirtualServer</code> , <code>VirtualServerRoute</code> и <code>TransportServer</code> , которые не имеют поля <code>ingressClassName</code> , во всех версиях kubernetes.	angie
<i>controller</i>	Новым Ingress без указанного поля <code>ingressClassName</code> будет присвоен класс, указанный в <code>controller.ingressClass</code> .	false
<i>controller</i>	Разделенный запятыми список пространств имен, за ресурсами которых должен следить Ingress Controller. По умолчанию Ingress Controller отслеживает все пространства имен. Взаимоисключающие с <code>controller.watchNamespaceLabel</code> . Обратите внимание, что при настройке нескольких пространств имен с использованием опции <code>Helm cli -set</code> строка должна быть заключена в двойные кавычки, а запятые экранированы с помощью обратной косой черты - например, <code>-set controller.watchNamespace=&gt;default,anic</code> .	«»
<i>controller</i>	Настраивает в Ingress Controller просмотр только пространств имен с меткой <code>foo=bar</code> . По умолчанию Ingress Controller отслеживает все пространства имен. Взаимоисключающая с <code>controller.watchNamespace</code> настройка.	«»
<i>controller</i>	Разделенный запятыми список пространств имен, за которыми Ingress Controller должен следить в поисках ресурсов типа <code>Secret</code> . Если этот параметр не настроен, Ingress Controller отслеживает одни и те же пространства имен в поисках всех ресурсов. См. также <code>controller.watchNamespace</code> и <code>controller.watchNamespaceLabel</code> . Обратите внимание, что при настройке нескольких пространств имен с использованием опции <code>Helm cli -set</code> строка должна быть заключена в двойные кавычки, а запятые экранированы с помощью обратной косой черты - например, <code>-set controller.watchSecretNamespace=&gt;default,angie-ingress</code> .	«»
<i>controller</i>	Включает пользовательские ресурсы.	true
<i>controller</i>	Включает политики предварительного просмотра. Этот параметр устарел. Чтобы включить политики OIDS, используйте вместо него <code>controller.enableOIDC</code> .	false
<i>controller</i>	Включает политики OIDS.	false
<p><b>Предупреждение:</b> Пока не работает, но вскоре будет.</p>		
<i>controller</i>	Включает передачу данных по протоколу TLS на порту 443. Требуется <code>controller.enableCustomResources</code> .	false

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controller</i>	Включает автоматическое управление сертификатами x509 для ресурсов виртуального сервера с помощью cert-manager (cert-manager.io). Требуется <i>controller.enableCustomResources</i> .	false
<i>controller</i>	Включает интеграцию с ExternalDNS для настройки общедоступных записей DNS у ресурсов VirtualServer с использованием ExternalDNS. Требуется <i>controller.enableCustomResources</i> .	false
<i>controller</i>	Создает пользовательский ресурс GlobalConfiguration. Требуется <i>controller.enableCustomResources</i> .	false
<i>controller</i>	Спецификация GlobalConfiguration для определения параметров глобальной конфигурации Ingress Controller.	{}
<i>controller</i>	Включает пользовательские фрагменты конфигурации Angie в ресурсах Ingress, VirtualServer, VirtualServerRoute и TransportServer.	false
<i>controller</i>	Добавляет местоположение «/angie-health» на сервер по умолчанию. Местоположение отвечает кодом статуса 200 на любой запрос. Это полезно для внешней проверки работоспособности Ingress Controller.	false
<i>controller</i>	Задаёт URI местоположения состояния работоспособности на сервере по умолчанию. Требуется <i>controller.HealthStatus</i> .	«/angie-health»
<i>controller</i>	Включает в Angie API.	true
<i>controller</i>	Задаёт порт, на котором доступен Angie API.	8080
<i>controller</i>	Добавляет блоки IP или CIDR в список разрешенных для Angie API. Несколько IP или CIDR разделяются запятыми.	127.0.0.1,::1
<i>controller</i>	Класс приоритета подов Ingress Controller.	Нет
<i>controller</i>	Создаёт сервис для предоставления доступа к подам Ingress Controller.	true
<i>controller</i>	Тип сервиса, который необходимо создать для Ingress Controller.	LoadBalancer
<i>controller</i>	Внешняя политика трафика сервиса. Значение Local сохраняет исходный IP-адрес клиента.	Local
<i>controller</i>	Аннотации сервиса Ingress Controller.	{}
<i>controller</i>	Экстра-метки сервиса.	{}
<i>controller</i>	Статический IP-адрес для балансировщика нагрузки. Для <i>controller.service.type</i> должно быть установлено значение <i>LoadBalancer</i> . Поставщик облачных услуг должен поддерживать эту функцию.	«»
<i>controller</i>	Список внешних IP-адресов для сервиса Ingress Controller.	[]
<i>controller</i>	Диапазоны IP-адресов (CIDR), которым разрешен доступ к балансировщику нагрузки. Для <i>controller.service.type</i> должно быть установлено значение <i>LoadBalancer</i> . Поставщик облачных услуг должен поддерживать эту функцию.	[]
<i>controller</i>	Имя сервиса.	Создается автоматически
<i>controller</i>	Список пользовательских портов, которые будут доступны через сервис Ingress Controller. Следует обычному синтаксису yaml Kubernetes для портов сервиса.	[]
<i>controller</i>	Включает HTTP-порт для сервиса Ingress Controller.	true
<i>controller</i>	HTTP-порт сервиса Ingress Controller.	80
<i>controller</i>	Пользовательский NodePort для HTTP-порта. Для <i>controller.service.type</i> должно быть установлено значение <i>NodePort</i> .	«»
<i>controller</i>	Целевое значение HTTP-порта сервиса Ingress Controller.	80
<i>controller</i>	Включает порт HTTPS для сервиса Ingress Controller.	true
<i>controller</i>	HTTPS-порт сервиса Ingress Controller.	443
<i>controller</i>	Пользовательский NodePort для HTTPS-порта. Для <i>controller.service.type</i> должно быть установлено значение <i>NodePort</i> .	«»

continues on next page



Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controller</i>	Целевой порт HTTPS-порта сервиса Ingress Controller.	443
<i>controller</i>	Аннотации учетной записи сервиса Ingress Controller.	{}
<i>controller</i>	Имя учетной записи сервиса подов Ingress Controller. Используется для RBAC.	Создается автоматически
<i>controller</i>	Имя секретного файла, содержащего учетные данные реестра Docker. Секрет должен находиться в том же пространстве имен, что и выпуск Helm.	«»
<i>controller</i>	Имя serviceMonitor.	Создается автоматически
<i>controller</i>	Создает пользовательский ресурс ServiceMonitor.	false
<i>controller</i>	Метки объектов Kubernetes для применения к объекту serviceMonitor.	«»
<i>controller</i>	Набор меток, позволяющих выбирать конечные точки для serviceMonitor.	«»
<i>controller</i>	Список конечных точек, разрешенных в составе этого serviceMonitor.	«»
<i>controller</i>	Добавляет в поле адреса в статусе ресурсов Ingress внешний адрес Ingress Controller. Нужно также указать источник внешнего адреса через внешнюю службу через <i>controller.reportIngressStatus.ExternalService</i> , либо через <i>controller.reportIngressStatus.ingressLink</i> , либо через запись <i>external-status-address</i> в ConfigMap через <i>controller.config.entries</i> .	true
<p><b>Примечание:</b> Значение <i>controller.config.entries.external-status-address</i> имеет приоритет над остальными.</p>		
<i>controller</i>	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 <i>controller.reportIngressStatus.enable</i> должно быть задано как <i>true</i> . Значение по умолчанию создается автоматически и включается, когда <i>controller.service.create</i> имеет значение <i>true</i> , а <i>controller.service.type</i> - значение <i>LoadBalancer</i> .	Создается автоматически
<i>controller</i>	Указывает имя ресурса IngressLink, который предоставляет доступ к подам Ingress Controller через систему BIG-IP. IP-адрес системы BIG-IP используется для отчетов о состоянии ресурсов Ingress, VirtualServer и VirtualServerRoute. Значение <i>controller.reportIngressStatus.enable</i> должно быть задано как <i>true</i> .	«»
<i>controller</i>	Включает выбор лидера, чтобы избежать ситуации, когда несколько реплик контроллера сообщают о состоянии ресурсов Ingress. Значение <i>controller.reportIngressStatus.enable</i> должно быть задано как <i>true</i> .	true
<i>controller</i>	Указывает имя ConfigMap в том же пространстве имен, что и контроллер, которое используется для блокировки выбора лидера. Значение <i>controller.reportIngressStatus.enableLeaderElection</i> должно быть задано как <i>true</i> .	Создается автоматически
<i>controller</i>	Аннотации к конфигурационной карте выборов лидера.	{}
<i>controller</i>	Аннотации пода Ingress Controller.	{}
<i>controller</i>	Дополнительные экстра-метки для пода Ingress Controller.	{}

continues on next page

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controlle</i>	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 Ingress Controller, которая использует конечную точку готовности.	true
<i>controlle</i>	HTTP-порт для конечной точки готовности.	8081
<i>controlle</i>	Число секунд с запуска пода Ingress Controller до инициирования проверки готовности.	0
<i>controlle</i>	Включает сбор метрик задержки для апстримов. Требуется <i>prometheus.create</i> .	false
<i>controlle</i>	Задаёт 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. <a href="#">здесь</a> .	0
<i>controlle</i>	Включает HorizontalPodAutoscaling.	false
<i>controlle</i>	Аннотации HorizontalPodAutoscaler для Ingress Controller.	{}
<i>controlle</i>	Минимальное число реплик для НРА.	1
<i>controlle</i>	Максимальное число реплик для НРА.	3
<i>controlle</i>	Целевой процент загрузки ЦП.	50
<i>controlle</i>	Целевой процент использования памяти.	50
<i>controlle</i>	Включает PodDisruptionBudget.	false
<i>controlle</i>	Аннотации к бюджету сбоев пода Ingress Controller	{}
<i>controlle</i>	Количество подов Ingress Controller, которые должны быть доступны. Взаимоисключающая с «maxUnavailable» настройка.	0
<i>controlle</i>	Количество подов Ingress Controller, которые могут быть недоступны. Взаимоисключающая с «minAvailable» настройка.	0
<i>controlle</i>	Задаёт стратегию замены старых подов новыми. Документация по <a href="#">стратегии обновления развертывания</a> и <a href="#">стратегии обновления набора демонов</a>	{}
<i>controlle</i>	В явной форме отключает прослушиватели IPV6 для узлов, которые не поддерживают стек IPV6.	false
<i>controlle</i>	Настраивает корневую файловую систему как доступную только для чтения и добавляет тома для временных данных.	false
<i>rbac.crea</i>	Настраивает RBAC.	true
<i>promethe</i>	Публикует метрики Angie в формате Prometheus.	true
<i>promethe</i>	Настраивает порт для получения метрик.	9113
<i>promethe</i>	Настраивает схему HTTP, используемую для подключений к конечной точке Prometheus.	http
<i>promethe</i>	Пространство имен или имя TLS-секрета Kubernetes. Если секрет указан, он используется для защиты конечной точки Prometheus с помощью TLS-соединений.	«»

---

## Аргументы командной строки

---

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

- 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 перечислены доступные аргументы командной строки:

### 4.1 `-enable-snippets`

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

По умолчанию `false`.

### 4.2 `-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 Ingress Controller не запустится.

Формат: <пространство имен>/<имя>

### 4.3 -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>

### 4.4 -enable-custom-resources

Включает пользовательские ресурсы.

По умолчанию `true`.

### 4.5 -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`.

### 4.6 -enable-tls-passthrough

Включает сквозную передачу данных по протоколу TLS на порту 443.

Требует наличия `-enable-custom-resources`.

### 4.7 -tls-passthrough-port <int>

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

Требует включить `-enable-custom-resources`.

### 4.8 -enable-cert-manager

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

Требует `-enable-custom-resources`.

## 4.9 -enable-external-dns

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

Требует наличия *-enable-custom-resources*.

## 4.10 -external-service <строка>

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

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

## 4.11 -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*.

## 4.12 -global-configuration <строка>

Ресурс GlobalConfiguration для глобальной настройки ANIC.

Формат: <пространство имен>/<имя>

Требует наличия *-enable-custom-resources*.

## 4.13 -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.

## 4.14 -health-status-uri <строка>

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

По умолчанию /angie-health.

## 4.15 `-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`.

## 4.16 `-ingress-template-path` <строка>

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

## 4.17 `-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`.

## 4.18 `-main-template-path` <строка>

Путь к основному шаблону конфигурации Angie.

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

## 4.19 `-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>

## 4.20 `-angie-debug`

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

## 4.21 `-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.

## 4.22 `-angie-status`

Включает Angie `stub_status`.

По умолчанию `true`.

## 4.23 `-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`)

## 4.24 `-angie-status-port` <int>

Задаёт порт, на котором доступен Angie `stub_status`.

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

## 4.25 `-proxy` <строка>

Задаёт 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.

## 4.26 `-report-ingress-status`

Обновляет поле адреса в статусе ресурсов Ingress.

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

## 4.27 `-transportserver-template-path` <строка>

Путь к шаблону конфигурации TransportServer Angie для ресурса TransportServer.

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

#### 4.28 `-v<значение>`

Уровень ведения журнала для журналов V.

#### 4.29 `-version`

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

#### 4.30 `-virtualserver-template-path <строка>`

Путь к шаблону конфигурации VirtualServer Angie для ресурса VirtualServer.

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

#### 4.31 `-vmodule <значение>`

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

#### 4.32 `-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`».

#### 4.33 `-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`».

#### 4.34 `-watch-secret-namespace <строка>`

Разделенный запятыми список пространств имен, за которыми Ingress Controller 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`».

#### 4.35 `-enable-prometheus-metrics`

Позволяет публиковать метрики Angie в формате Prometheus.



### 4.36 `-prometheus-metrics-listen-port` <int>

Задаёт порт, на котором публикуются метрики Prometheus.

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

### 4.37 `-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.

### 4.38 `-enable-service-insight`

Публикует конечную точку Service Insight для ANIC.

### 4.39 `-service-insight-listen-port` <int>

Задаёт порт, на котором публикуется Service Insight.

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

### 4.40 `-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>

### 4.41 `-ready-status`

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

По умолчанию `true`.

## 4.42 -ready-status-port

HTTP-порт для конечной точки готовности.

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

## 4.43 -disable-ipv6

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

По умолчанию `false`.