

ANIC

Angie Ingress Controller

версия 0.5.0

Руководство по установке и запуску

сент. 27, 2024

Оглавление

1	Аннотация	1
1.1	Общие сведения	1
2	Установка	3
2.1	Поддерживаемые дистрибутивы	3
2.2	Установка в K8S	3
3	Установка с помощью Helm	47
3.1	Вступление	47
3.2	Предварительные требования	47
3.3	Пользовательские определения ресурсов	47
3.4	Управление диаграммой с помощью реестра	48
3.5	Конфигурация	49
4	Аргументы командной строки	55
4.1	-enable-snippets	55
4.2	-default-server-tls-secret <строка>	55
4.3	-wildcard-tls-secret <строка>	56
4.4	-enable-custom-resources	56
4.5	-enable-leader-election	56
4.6	-enable-tls-passthrough	56
4.7	-tls-passthrough-port <int>	56
4.8	-enable-cert-manager	56
4.9	-enable-external-dns	57
4.10	-enable-oidc	57
4.11	-enable-jwt	57
4.12	-external-service <строка>	57
4.13	-ingresslink <строка>	57
4.14	-global-configuration <строка>	57
4.15	-health-status	58
4.16	-health-status-uri <строка>	58
4.17	-ingress-class <строка>	58
4.18	-ingress-template-path <строка>	58
4.19	-leader-election-lock-name <строка>	58
4.20	-main-template-path <строка>	58
4.21	-angie-configmaps <строка>	59
4.22	-angie-debug	59
4.23	-angie-reload-timeout <значение>	59
4.24	-angie-status	59
4.25	-angie-status-allow-cidrs <строка>	59
4.26	-angie-status-port <int>	59

4.27	-angie-status-prometheus <bool>	59
4.28	-angie-status-prometheus-path <строка>	60
4.29	-angie-status-prometheus-port <int>	60
4.30	-angie-status-prometheus-allow-cidrs	60
4.31	-proxy <строка>	60
4.32	-report-ingress-status	60
4.33	-transportserver-template-path <строка>	60
4.34	-v <значение>	60
4.35	-version	61
4.36	-virtualserver-template-path <строка>	61
4.37	-vmodule <значение>	61
4.38	-watch-namespace <строка>	61
4.39	-watch-namespace-label <строка>	61
4.40	-watch-secret-namespace <строка>	61
4.41	-enable-prometheus-metrics	61
4.42	-prometheus-metrics-listen-port <int>	61
4.43	-prometheus-tls-secret <строка>	62
4.44	-enable-service-insight	62
4.45	-service-insight-listen-port <int>	62
4.46	-service-insight-tls-secret <строка>	62
4.47	-ready-status	62
4.48	-ready-status-port	62
4.49	-disable-ipv6	63

ГЛАВА 1

Аннотация

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

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

Примечание

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

1.1 Общие сведения

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

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

ANIC использует широкий набор функций Ingress:

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

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

ГЛАВА 2

Установка

2.1 Поддерживаемые дистрибутивы

Название	Версии	Архитектуры
Alpine Linux	3.17	x86_64, ARM64
Debian	11 “Bullseye”	x86_64, ARM64
ALT Linux	10	x86_64, ARM64

2.2 Установка в K8S

2.2.1 Предварительные требования

Необходим доступ к Docker-образу в нашем репозитории:

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

Для текущей версии доступны следующие образы:

```
anic.docker.angie.software/anic:0.5.0-alpine
anic.docker.angie.software/anic:0.5.0-debian
anic.docker.angie.software/anic:0.5.0-altlinux
```

За доступом обращайтесь на 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:
  - ""
  resources:
  - secrets
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - configmaps
  verbs:
  - get
```

```

- list
- watch
- update
- create
- apiGroups:
  - ""
  resources:
  - pods
  verbs:
  - get
  - list
  - watch
  - update
- apiGroups:
  - ""
  resources:
  - namespaces
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - ""
  resources:
  - events
  verbs:
  - create
  - patch
  - list
- apiGroups:
  - coordination.k8s.io
  resources:
  - leases
  verbs:
  - get
  - list
  - watch
  - update
  - create
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - list
  - watch
  - get
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses/status
  verbs:
  - update
- apiGroups:
  - k8s.angie.software
  resources:
  - virtualservers

```



```

- virtualserverroutes
- globalconfigurations
- transportservers
- policies
verbs:
- list
- watch
- get
- apiGroups:
- k8s.angie.software
resources:
- virtualservers/status
- virtualserverroutes/status
- policies/status
- transportservers/status
- dnsendpoints/status
verbs:
- update
- apiGroups:
- networking.k8s.io
resources:
- ingressclasses
verbs:
- get
- apiGroups:
- cis.f5.com
resources:
- ingresslinks
verbs:
- list
- watch
- get
- apiGroups:
- cert-manager.io
resources:
- certificates
verbs:
- list
- watch
- get
- update
- create
- delete
- apiGroups:
- externaldns.angie.software
resources:
- dnsendpoints
verbs:
- list
- watch
- get
- update
- create
- delete
- apiGroups:
- externaldns.angie.software
resources:

```

```

- dnsendpoints/status
verbs:
- update
---
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: angie-ingress
subjects:
- kind: ServiceAccount
  name: angie-ingress
  namespace: angie-ingress
roleRef:
  kind: ClusterRole
  name: angie-ingress
  apiGroup: rbac.authorization.k8s.io
EOF

```

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
  namespace: angie-ingress
data:
EOF

```

5. Создайте IngressClass:

```

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

```

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

Пример Virtual Server

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: virtualservers.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: VirtualServer
    listKind: VirtualServerList
    plural: virtualservers
    shortNames:
      - vs
    singular: virtualserver
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
      - description: Current state of the VirtualServer. If the resource has a
→valid status, it means it has been validated and accepted by ANIC.
        jsonPath: .status.state
        name: State
        type: string
      - jsonPath: .spec.host
        name: Host
        type: string
      - jsonPath: .status.externalEndpoints[*].ip
        name: IP
        type: string
      - jsonPath: .status.externalEndpoints[*].hostname
        name: ExternalHostname
        priority: 1
        type: string
      - jsonPath: .status.externalEndpoints[*].ports
        name: Ports
        type: string
      - jsonPath: .metadata.creationTimestamp
        name: Age
        type: date
    name: v1
  schema:
    openAPIV3Schema:
      description: VirtualServer defines the VirtualServer resource.
      type: object
      properties:
        apiVersion:
          description: 'APIVersion defines the versioned schema of this
→representation of an object. Servers should convert recognized schemas to the
→latest internal value, and may reject unrecognized values. More info: https://
→git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
```

```

→#resources'
  type: string
  kind:
    description: 'Kind is a string value representing the REST
→resource this object represents. Servers may infer this from the endpoint the
→client submits requests to. Cannot be updated. In CamelCase. More info: https://
→/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
→#types-kinds'
  type: string
  metadata:
    type: object
  spec:
    description: VirtualServerSpec is the spec of the VirtualServer
→resource.
  type: object
  properties:
    dos:
      type: string
    externalDNS:
      description: ExternalDNS defines externaldns sub-resource of a
→virtual server.
  type: object
  properties:
    enable:
      type: boolean
    labels:
      description: Labels stores labels defined for the Endpoint
      type: object
      additionalProperties:
        type: string
    providerSpecific:
      description: ProviderSpecific stores provider specific
→config
      type: array
      items:
        description: ProviderSpecificProperty defines specific
→property for using with ExternalDNS sub-resource.
      type: object
      properties:
        name:
          description: Name of the property
          type: string
        value:
          description: Value of the property
          type: string
      recordTTL:
        description: TTL for the record
        type: integer
        format: int64
      recordType:
        type: string
    host:
      type: string
    http-snippets:
      type: string
    ingressClassName:
      type: string

```

```

    policies:
      type: array
      items:
        description: PolicyReference references a policy by name and
→an optional namespace.
        type: object
        properties:
          name:
            type: string
          namespace:
            type: string
    routes:
      type: array
      items:
        description: Route defines a route.
        type: object
        properties:
          action:
            description: Action defines an action.
            type: object
            properties:
              pass:
                type: string
              proxy:
                description: ActionProxy defines a proxy in an
→Action.
                type: object
                properties:
                  requestHeaders:
                    description: ProxyRequestHeaders defines the
→request headers manipulation in an ActionProxy.
                    type: object
                    properties:
                      pass:
                        type: boolean
                      set:
                        type: array
                        items:
                          description: Header defines an HTTP Header.
                          type: object
                          properties:
                            name:
                              type: string
                            value:
                              type: string
                      responseHeaders:
                        description: ProxyResponseHeaders defines the
→response headers manipulation in an ActionProxy.
                        type: object
                        properties:
                          add:
                            type: array
                            items:
                              description: AddHeader defines an HTTP
→Header with an optional Always field to use with the add_header directive.
                              type: object
                              properties:

```

```

        always:
            type: boolean
        name:
            type: string
        value:
            type: string
    hide:
        type: array
        items:
            type: string
    ignore:
        type: array
        items:
            type: string
    pass:
        type: array
        items:
            type: string
    rewritePath:
        type: string
    upstream:
        type: string
    redirect:
        description: ActionRedirect defines a redirect in an
→Action.
        type: object
        properties:
            code:
                type: integer
            url:
                type: string
    return:
        description: ActionReturn defines a return in an
→Action.
        type: object
        properties:
            body:
                type: string
            code:
                type: integer
            type:
                type: string
    dos:
        type: string
    errorPages:
        type: array
        items:
            description: ErrorPage defines an ErrorPage in a Route.
            type: object
            properties:
                codes:
                    type: array
                    items:
                        type: integer
                redirect:
                    description: ErrorPageRedirect defines a redirect
→for an ErrorPage.

```

```

        type: object
        properties:
          code:
            type: integer
          url:
            type: string
        return:
          description: ErrorPageReturn defines a return for
↳ an ErrorPage.

        type: object
        properties:
          body:
            type: string
          code:
            type: integer
          headers:
            type: array
            items:
              description: Header defines an HTTP Header.
              type: object
              properties:
                name:
                  type: string
                value:
                  type: string
            type:
              type: string
        location-snippets:
          type: string
        matches:
          type: array
          items:
            description: Match defines a match.
            type: object
            properties:
              action:
                description: Action defines an action.
                type: object
                properties:
                  pass:
                    type: string
                  proxy:
                    description: ActionProxy defines a proxy in an
↳ Action.

                type: object
                properties:
                  requestHeaders:
                    description: ProxyRequestHeaders defines
↳ the request headers manipulation in an ActionProxy.
                    type: object
                    properties:
                      pass:
                        type: boolean
                      set:
                        type: array
                        items:
                          description: Header defines an HTTP

```

```

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

```



```

        code:
          type: integer
        type:
          type: string
      conditions:
        type: array
        items:
          description: Condition defines a condition in a
↳MatchRule.
          type: object
          properties:
            argument:
              type: string
            cookie:
              type: string
            header:
              type: string
            value:
              type: string
            variable:
              type: string
        splits:
          type: array
          items:
            description: Split defines a split.
            type: object
            properties:
              action:
                description: Action defines an action.
                type: object
                properties:
                  pass:
                    type: string
                  proxy:
                    description: ActionProxy defines a proxy
↳in an Action.
                    type: object
                    properties:
                      requestHeaders:
                        description: ProxyRequestHeaders
↳defines the request headers manipulation in an ActionProxy.
                        type: object
                        properties:
                          pass:
                            type: boolean
                          set:
                            type: array
                            items:
                              description: Header defines an
↳HTTP Header.
                              type: object
                              properties:
                                name:
                                  type: string
                                value:
                                  type: string
                              responseHeaders:

```

```

description: ProxyResponseHeaders
↳defines the response headers manipulation in an ActionProxy.
type: object
properties:
  add:
    type: array
    items:
      description: AddHeader defines
↳an HTTP Header with an optional Always field to use with the add_header
↳directive.
      type: object
      properties:
        always:
          type: boolean
        name:
          type: string
        value:
          type: string
      hide:
        type: array
        items:
          type: string
      ignore:
        type: array
        items:
          type: string
      pass:
        type: array
        items:
          type: string
      rewritePath:
        type: string
      upstream:
        type: string
    redirect:
      description: ActionRedirect defines a
↳redirect in an Action.
      type: object
      properties:
        code:
          type: integer
        url:
          type: string
      return:
      description: ActionReturn defines a
↳return in an Action.
      type: object
      properties:
        body:
          type: string
        code:
          type: integer
        type:
          type: string
      weight:
        type: integer
    path:

```

```

    type: string
  policies:
    type: array
    items:
      description: PolicyReference references a policy by
↳name and an optional namespace.
      type: object
      properties:
        name:
          type: string
        namespace:
          type: string
    route:
      type: string
    splits:
      type: array
      items:
        description: Split defines a split.
        type: object
        properties:
          action:
            description: Action defines an action.
            type: object
            properties:
              pass:
                type: string
              proxy:
                description: ActionProxy defines a proxy in an
↳Action.
                type: object
                properties:
                  requestHeaders:
                    description: ProxyRequestHeaders defines
↳the request headers manipulation in an ActionProxy.
                    type: object
                    properties:
                      pass:
                        type: boolean
                      set:
                        type: array
                        items:
                          description: Header defines an HTTP
↳Header.
                          type: object
                          properties:
                            name:
                              type: string
                            value:
                              type: string
                      responseHeaders:
                    description: ProxyResponseHeaders defines
↳the response headers manipulation in an ActionProxy.
                    type: object
                    properties:
                      add:
                        type: array
                        items:

```

```

description: AddHeader defines an
→HTTP Header with an optional Always field to use with the add_header directive.
type: object
properties:
  always:
    type: boolean
  name:
    type: string
  value:
    type: string
hide:
  type: array
  items:
    type: string
ignore:
  type: array
  items:
    type: string
pass:
  type: array
  items:
    type: string
rewritePath:
  type: string
upstream:
  type: string
redirect:
description: ActionRedirect defines a redirect
→in an Action.
type: object
properties:
  code:
    type: integer
  url:
    type: string
return:
description: ActionReturn defines a return in
→an Action.
type: object
properties:
  body:
    type: string
  code:
    type: integer
  type:
    type: string
weight:
  type: integer
server-snippets:
  type: string
tls:
description: TLS defines TLS configuration for a VirtualServer.
type: object
properties:
  cert-manager:
description: CertManager defines a cert manager config for
→a TLS.

```

```

type: object
properties:
  cluster-issuer:
    type: string
  common-name:
    type: string
  duration:
    type: string
  issuer:
    type: string
  issuer-group:
    type: string
  issuer-kind:
    type: string
  renew-before:
    type: string
  usages:
    type: string
redirect:
  description: TLSRedirect defines a redirect for a TLS.
  type: object
  properties:
    basedOn:
      type: string
    code:
      type: integer
    enable:
      type: boolean
secret:
  type: string
upstreams:
  type: array
  items:
    description: Upstream defines an upstream.
    type: object
    properties:
      buffer-size:
        type: string
      buffering:
        type: boolean
      buffers:
        description: UpstreamBuffers defines Buffer
        type: object
        properties:
          number:
            type: integer
          size:
            type: string
      client-max-body-size:
        type: string
      connect-timeout:
        type: string
      fail-timeout:
        type: string
      healthCheck:
        description: HealthCheck defines the parameters for

```

```

→active Upstream HealthChecks.
    type: object
    properties:
      connect-timeout:
        type: string
      enable:
        type: boolean
      fails:
        type: integer
      grpcService:
        type: string
      grpcStatus:
        type: integer
      headers:
        type: array
        items:
          description: Header defines an HTTP Header.
          type: object
          properties:
            name:
              type: string
            value:
              type: string
      interval:
        type: string
      jitter:
        type: string
      keepalive-time:
        type: string
      mandatory:
        type: boolean
      passes:
        type: integer
      path:
        type: string
      persistent:
        type: boolean
      port:
        type: integer
      read-timeout:
        type: string
      send-timeout:
        type: string
      statusMatch:
        type: string
      tls:
        description: UpstreamTLS defines a TLS configuration.
→for an Upstream.
    type: object
    properties:
      enable:
        type: boolean
    keepalive:
      type: integer
    lb-method:
      type: string
    max-conns:

```

```

        type: integer
max-fails:
    type: integer
name:
    type: string
next-upstream:
    type: string
next-upstream-timeout:
    type: string
next-upstream-tries:
    type: integer
ntlm:
    type: boolean
port:
    type: integer
queue:
    description: UpstreamQueue defines Queue Configuration.
→for an Upstream.
    type: object
    properties:
        size:
            type: integer
        timeout:
            type: string
read-timeout:
    type: string
send-timeout:
    type: string
service:
    type: string
sessionCookie:
    description: SessionCookie defines the parameters for.
→session persistence.
    type: object
    properties:
        domain:
            type: string
        enable:
            type: boolean
        expires:
            type: string
        httpOnly:
            type: boolean
        name:
            type: string
        path:
            type: string
        secure:
            type: boolean
slow-start:
    type: string
subselector:
    type: object
    additionalProperties:
        type: string
tls:
    description: UpstreamTLS defines a TLS configuration for.

```

```

↳an Upstream.
        type: object
        properties:
            enable:
                type: boolean
        type:
            type: string
        use-cluster-ip:
            type: boolean
    status:
        description: VirtualServerStatus defines the status for the
↳VirtualServer resource.
        type: object
        properties:
            externalEndpoints:
                type: array
                items:
                    description: ExternalEndpoint defines the IP/ Hostname and
↳ports used to connect to this resource.
                    type: object
                    properties:
                        hostname:
                            type: string
                        ip:
                            type: string
                        ports:
                            type: string
                    message:
                        type: string
                    reason:
                        type: string
                    state:
                        type: string
        served: true
        storage: true
        subresources:
            status: {}
EOF

```

Пример VirtualServerRoute

```

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

```



```

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

```

```

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

```

```

        ignore:
            type: array
            items:
                type: string
        pass:
            type: array
            items:
                type: string
        rewritePath:
            type: string
        upstream:
            type: string
    redirect:
        description: ActionRedirect defines a redirect in an
→Action.
        type: object
        properties:
            code:
                type: integer
            url:
                type: string
    return:
        description: ActionReturn defines a return in an
→Action.
        type: object
        properties:
            body:
                type: string
            code:
                type: integer
            type:
                type: string
    dos:
        type: string
    errorPages:
        type: array
        items:
            description: ErrorPage defines an ErrorPage in a Route.
            type: object
            properties:
                codes:
                    type: array
                    items:
                        type: integer
                redirect:
                    description: ErrorPageRedirect defines a redirect
→for an ErrorPage.
                    type: object
                    properties:
                        code:
                            type: integer
                        url:
                            type: string
                return:
                    description: ErrorPageReturn defines a return for
→an ErrorPage.
                    type: object

```

```

        properties:
          body:
            type: string
          code:
            type: integer
          headers:
            type: array
            items:
              description: Header defines an HTTP Header.
              type: object
              properties:
                name:
                  type: string
                value:
                  type: string
              type:
                type: string
        location-snippets:
          type: string
        matches:
          type: array
          items:
            description: Match defines a match.
            type: object
            properties:
              action:
                description: Action defines an action.
                type: object
                properties:
                  pass:
                    type: string
                  proxy:
                    description: ActionProxy defines a proxy in an
→Action.
                    type: object
                    properties:
                      requestHeaders:
                        description: ProxyRequestHeaders defines
→the request headers manipulation in an ActionProxy.
                        type: object
                        properties:
                          pass:
                            type: boolean
                          set:
                            type: array
                            items:
                                description: Header defines an HTTP
→Header.
                                type: object
                                properties:
                                  name:
                                    type: string
                                  value:
                                    type: string
                                responseHeaders:
                                  description: ProxyResponseHeaders defines
→the response headers manipulation in an ActionProxy.

```

```

        type: object
        properties:
          add:
            type: array
            items:
              description: AddHeader defines an
→HTTP Header with an optional Always field to use with the add_header directive.
              type: object
              properties:
                always:
                  type: boolean
                name:
                  type: string
                value:
                  type: string
            hide:
              type: array
              items:
                type: string
            ignore:
              type: array
              items:
                type: string
            pass:
              type: array
              items:
                type: string
            rewritePath:
              type: string
            upstream:
              type: string
        redirect:
          description: ActionRedirect defines a redirect
→in an Action.
          type: object
          properties:
            code:
              type: integer
            url:
              type: string
        return:
          description: ActionReturn defines a return in
→an Action.
          type: object
          properties:
            body:
              type: string
            code:
              type: integer
            type:
              type: string
        conditions:
          type: array
          items:
            description: Condition defines a condition in a
→MatchRule.
            type: object

```



```

                                type: object
                                properties:
                                  always:
                                    type: boolean
                                  name:
                                    type: string
                                  value:
                                    type: string
                                hide:
                                  type: array
                                  items:
                                    type: string
                                ignore:
                                  type: array
                                  items:
                                    type: string
                                pass:
                                  type: array
                                  items:
                                    type: string
                                rewritePath:
                                  type: string
                                upstream:
                                  type: string
                                redirect:
                                  description: ActionRedirect defines a
→redirect in an Action.
                                type: object
                                properties:
                                  code:
                                    type: integer
                                  url:
                                    type: string
                                return:
                                  description: ActionReturn defines a
→return in an Action.
                                type: object
                                properties:
                                  body:
                                    type: string
                                  code:
                                    type: integer
                                  type:
                                    type: string
                                weight:
                                  type: integer
                                path:
                                  type: string
                                policies:
                                  type: array
                                  items:
                                    description: PolicyReference references a policy by
→name and an optional namespace.
                                    type: object
                                    properties:
                                      name:
                                        type: string

```

```

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

```



```

        hide:
          type: array
          items:
            type: string
        ignore:
          type: array
          items:
            type: string
        pass:
          type: array
          items:
            type: string
        rewritePath:
          type: string
        upstream:
          type: string
    redirect:
      description: ActionRedirect defines a redirect
      type: object
      properties:
        code:
          type: integer
        url:
          type: string
    return:
      description: ActionReturn defines a return in
      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
          type: object
          properties:
            number:
              type: integer
            size:
              type: string

```

```

client-max-body-size:
  type: string
connect-timeout:
  type: string
fail-timeout:
  type: string
healthCheck:
  description: HealthCheck defines the parameters for
↳active Upstream HealthChecks.
  type: object
  properties:
    connect-timeout:
      type: string
    enable:
      type: boolean
    fails:
      type: integer
    grpcService:
      type: string
    grpcStatus:
      type: integer
    headers:
      type: array
      items:
        description: Header defines an HTTP Header.
        type: object
        properties:
          name:
            type: string
          value:
            type: string
    interval:
      type: string
    jitter:
      type: string
    keepalive-time:
      type: string
    mandatory:
      type: boolean
    passes:
      type: integer
    path:
      type: string
    persistent:
      type: boolean
    port:
      type: integer
    read-timeout:
      type: string
    send-timeout:
      type: string
    statusMatch:
      type: string
    tls:
      description: UpstreamTLS defines a TLS configuration
↳for an Upstream.
      type: object

```

```

        properties:
            enable:
                type: boolean
    keepalive:
        type: integer
    lb-method:
        type: string
    max-conns:
        type: integer
    max-fails:
        type: integer
    name:
        type: string
    next-upstream:
        type: string
    next-upstream-timeout:
        type: string
    next-upstream-tries:
        type: integer
    ntlm:
        type: boolean
    port:
        type: integer
    queue:
        description: UpstreamQueue defines Queue Configuration
→for an Upstream.
        type: object
        properties:
            size:
                type: integer
            timeout:
                type: string
    read-timeout:
        type: string
    send-timeout:
        type: string
    service:
        type: string
    sessionCookie:
        description: SessionCookie defines the parameters for
→session persistence.
        type: object
        properties:
            domain:
                type: string
            enable:
                type: boolean
            expires:
                type: string
            httpOnly:
                type: boolean
            name:
                type: string
            path:
                type: string
            secure:
                type: boolean

```

```

        slow-start:
          type: string
        subselector:
          type: object
          additionalProperties:
            type: string
        tls:
          description: UpstreamTLS defines a TLS configuration for
↳an Upstream.
          type: object
          properties:
            enable:
              type: boolean
          type:
            type: string
        use-cluster-ip:
          type: boolean
      status:
        description: VirtualServerRouteStatus defines the status for the
↳VirtualServerRoute resource.
        type: object
        properties:
          externalEndpoints:
            type: array
            items:
              description: ExternalEndpoint defines the IP/ Hostname and
↳ports used to connect to this resource.
              type: object
              properties:
                hostname:
                  type: string
                ip:
                  type: string
                ports:
                  type: string
              message:
                type: string
              reason:
                type: string
              referencedBy:
                type: string
              state:
                type: string
      served: true
      storage: true
      subresources:
        status: {}
EOF

```

Пример TransportServer

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: transportservers.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: TransportServer
    listKind: TransportServerList
    plural: transportservers
    shortNames:
      - ts
    singular: transportserver
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
      - description: Current state of the TransportServer. If the resource has
↳a valid status, it means it has been validated and accepted by ANIC.
        jsonPath: .status.state
        name: State
        type: string
      - jsonPath: .status.reason
        name: Reason
        type: string
      - jsonPath: .metadata.creationTimestamp
        name: Age
        type: date
    name: v1alpha1
    schema:
      openAPIV3Schema:
        description: TransportServer defines the TransportServer resource.
        type: object
        properties:
          apiVersion:
            description: 'APIVersion defines the versioned schema of this
↳representation of an object. Servers should convert recognized schemas to the
↳latest internal value, and may reject unrecognized values. More info: https://
↳git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↳#resources'
            type: string
          kind:
            description: 'Kind is a string value representing the REST
↳resource this object represents. Servers may infer this from the endpoint the
↳client submits requests to. Cannot be updated. In CamelCase. More info: https://
↳/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↳#types-kinds'
            type: string
          metadata:
            type: object
          spec:
            description: TransportServerSpec is the spec of the
↳TransportServer resource.
```

```

type: object
properties:
  action:
    description: Action defines an action.
    type: object
    properties:
      pass:
        type: string
  host:
    type: string
  ingressClassName:
    type: string
  listener:
    description: TransportServerListener defines a listener for a
↳TransportServer.
    type: object
    properties:
      name:
        type: string
      protocol:
        type: string
      serverSnippets:
        type: string
      sessionParameters:
        description: SessionParameters defines session parameters.
        type: object
        properties:
          timeout:
            type: string
      streamSnippets:
        type: string
      tls:
        description: TLS defines TLS configuration for a
↳TransportServer.
        type: object
        properties:
          secret:
            type: string
      upstreamParameters:
        description: UpstreamParameters defines parameters for an
↳upstream.
        type: object
        properties:
          connectTimeout:
            type: string
          nextUpstream:
            type: boolean
          nextUpstreamTimeout:
            type: string
          nextUpstreamTries:
            type: integer
          udpRequests:
            type: integer
          udpResponses:
            type: integer
      upstreams:
        type: array

```

```

        items:
          description: Upstream defines an upstream.
          type: object
          properties:
            failTimeout:
              type: string
            healthCheck:
              description: HealthCheck defines the parameters for a
↳active Upstream HealthChecks.
              type: object
              properties:
                enable:
                  type: boolean
                fails:
                  type: integer
                interval:
                  type: string
                jitter:
                  type: string
                match:
              description: Match defines the parameters of a
↳custom health check.
              type: object
              properties:
                expect:
                  type: string
                send:
                  type: string
                passes:
                  type: integer
                port:
                  type: integer
                timeout:
                  type: string
                loadBalancingMethod:
                  type: string
                maxConns:
                  type: integer
                maxFails:
                  type: integer
                name:
                  type: string
                port:
                  type: integer
                service:
                  type: string
          status:
            description: TransportServerStatus defines the status for the
↳TransportServer resource.
            type: object
            properties:
              message:
                type: string
              reason:
                type: string
              state:
                type: string

```

```
served: true
storage: true
subresources:
  status: {}
EOF
```

Пример Policy

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: policies.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: Policy
    listKind: PolicyList
    plural: policies
    shortNames:
      - pol
    singular: policy
  scope: Namespaced
  versions:
    - additionalPrinterColumns:
      - description: Current state of the Policy. If the resource has a valid
→status, it means it has been validated and accepted by ANIC.
        jsonPath: .status.state
        name: State
        type: string
      - jsonPath: .metadata.creationTimestamp
        name: Age
        type: date
    name: v1
  schema:
    openAPIV3Schema:
      description: Policy defines a Policy for VirtualServer and
→VirtualServerRoute resources.
      type: object
      properties:
        apiVersion:
          description: 'APIVersion defines the versioned schema of this
→representation of an object. Servers should convert recognized schemas to the
→latest internal value, and may reject unrecognized values. More info: https://
→git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
→#resources'
          type: string
        kind:
          description: 'Kind is a string value representing the REST
→resource this object represents. Servers may infer this from the endpoint the
→client submits requests to. Cannot be updated. In CamelCase. More info: https://
→git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
→#types-kinds'
```



```

    type: string
  metadata:
    type: object
  spec:
    description: PolicySpec is the spec of the Policy resource. The
    ↳spec includes multiple fields, where each field represents a different policy.
    ↳Only one policy (field) is allowed.
    type: object
    properties:
      accessControl:
        description: AccessControl defines an access policy based on
        ↳the source IP of a request.
        type: object
        properties:
          allow:
            type: array
            items:
              type: string
          deny:
            type: array
            items:
              type: string
      basicAuth:
        description: 'BasicAuth holds HTTP Basic authentication
        ↳configuration policy status: preview'
        type: object
        properties:
          realm:
            type: string
          secret:
            type: string
      egressMTLS:
        description: EgressMTLS defines an Egress MTLs policy.
        type: object
        properties:
          ciphers:
            type: string
          protocols:
            type: string
          serverName:
            type: boolean
          sessionReuse:
            type: boolean
          sslName:
            type: string
          tlsSecret:
            type: string
          trustedCertSecret:
            type: string
          verifyDepth:
            type: integer
          verifyServer:
            type: boolean
      ingressClassName:
        type: string
      ingressMTLS:
        description: IngressMTLS defines an Ingress MTLs policy.

```

```

type: object
properties:
  clientCertSecret:
    type: string
  crlFileName:
    type: string
  verifyClient:
    type: string
  verifyDepth:
    type: integer
jwt:
  description: JWT holds JWT authentication configuration.
  realm: string
  secret: string
  token: string
oidc:
  description: OIDC defines an Open ID Connect policy.
  type: object
  properties:
    clientID:
      type: string
    clientSecret:
      type: string
    authEndpoint:
      type: string
    jwksURI:
      type: string
    tokenEndpoint:
      type: string
    scope:
      type: string
    accessTokenEnable:
      type: boolean
rateLimit:
  description: RateLimit defines a rate limit policy.
  type: object
  properties:
    burst:
      type: integer
    delay:
      type: integer
    dryRun:
      type: boolean
    key:
      type: string
    logLevel:
      type: string
    noDelay:
      type: boolean
    rate:
      type: string
    rejectCode:
      type: integer
    zoneSize:
      type: string
status:
  description: PolicyStatus is the status of the policy resource

```

```

    type: object
    properties:
      message:
        type: string
      reason:
        type: string
      state:
        type: string
    served: true
    storage: true
    subresources:
      status: {}
- name: v1alpha1
  schema:
    openAPIV3Schema:
      description: Policy defines a Policy for VirtualServer and
↳VirtualServerRoute resources.
      type: object
      properties:
        apiVersion:
          description: 'APIVersion defines the versioned schema of this
↳representation of an object. Servers should convert recognized schemas to the
↳latest internal value, and may reject unrecognized values. More info: https://
↳git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↳#resources'
          type: string
        kind:
          description: 'Kind is a string value representing the REST
↳resource this object represents. Servers may infer this from the endpoint the
↳client submits requests to. Cannot be updated. In CamelCase. More info: https://
↳/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
↳#types-kinds'
          type: string
        metadata:
          type: object
        spec:
          description: PolicySpec is the spec of the Policy resource. The
↳spec includes multiple fields, where each field represents a different policy.
↳Only one policy (field) is allowed.
          type: object
          properties:
            accessControl:
              description: AccessControl defines an access policy based on
↳the source IP of a request.
              type: object
              properties:
                allow:
                  type: array
                  items:
                    type: string
                deny:
                  type: array
                  items:
                    type: string
            egressMTLS:
              description: EgressMTLS defines an Egress MTLS policy.
              type: object

```

```

properties:
  ciphers:
    type: string
  protocols:
    type: string
  serverName:
    type: boolean
  sessionReuse:
    type: boolean
  sslName:
    type: string
  tlsSecret:
    type: string
  trustedCertSecret:
    type: string
  verifyDepth:
    type: integer
  verifyServer:
    type: boolean
ingressMTLS:
  description: IngressMTLS defines an Ingress MTLs policy.
  type: object
  properties:
    clientCertSecret:
      type: string
    verifyClient:
      type: string
    verifyDepth:
      type: integer
jwt:
  description: JWT holds JWT authentication configuration.
  realm: string
  secret: string
  token: string
rateLimit:
  description: RateLimit defines a rate limit policy.
  type: object
  properties:
    burst:
      type: integer
    delay:
      type: integer
    dryRun:
      type: boolean
    key:
      type: string
    logLevel:
      type: string
    noDelay:
      type: boolean
    rate:
      type: string
    rejectCode:
      type: integer
    zoneSize:
      type: string
served: true

```

```
storage: false
EOF
```

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

Пример

```
$ kubectl apply -f - <<EOF
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  annotations:
    controller-gen.kubebuilder.io/version: v0.11.3
  creationTimestamp: null
  name: globalconfigurations.k8s.angie.software
spec:
  group: k8s.angie.software
  names:
    kind: GlobalConfiguration
    listKind: GlobalConfigurationList
    plural: globalconfigurations
    shortNames:
      - gc
    singular: globalconfiguration
  scope: Namespaced
  versions:
    - name: v1alpha1
      schema:
        openAPIV3Schema:
          description: GlobalConfiguration defines the GlobalConfiguration
→resource.
          type: object
          properties:
            apiVersion:
              description: 'APIVersion defines the versioned schema of this
→representation of an object. Servers should convert recognized schemas to the
→latest internal value, and may reject unrecognized values. More info: https://
→git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
→#resources'
              type: string
            kind:
              description: 'Kind is a string value representing the REST
→resource this object represents. Servers may infer this from the endpoint the
→client submits requests to. Cannot be updated. In CamelCase. More info: https://
→/git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md
→#types-kinds'
              type: string
            metadata:
              type: object
            spec:
              description: GlobalConfigurationSpec is the spec of the
→GlobalConfiguration resource.
              type: object
              properties:
                listeners:
```

```

        type: array
        items:
          description: Listener defines a listener.
          type: object
          properties:
            name:
              type: string
            port:
              type: integer
            protocol:
              type: string
      served: true
      storage: true
EOF

```

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
#     emptyDir: {}
#     - name: angie-lib
#     emptyDir: {}
#     - name: angie-log
#     emptyDir: {}
containers:
- image: docker.angie.software/angie-ingress:latest
  imagePullPolicy: IfNotPresent
  name: angie-ingress
  ports:
  - name: http
    containerPort: 80
  - name: https
    containerPort: 443
  - name: readiness-port
    containerPort: 8081
  - name: prometheus
    containerPort: 9113
  readinessProbe:
    httpGet:
      path: /angie-ready
      port: readiness-port
      periodSeconds: 1
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
  #limits
  # cpu: "1"
  # memory: "1Gi"
  securityContext:
    allowPrivilegeEscalation: false
    runAsUser: 101 #angie
    runAsNonRoot: true
    capabilities:
      drop:
        - ALL
#     volumeMounts:
#     - mountPath: /etc/angie
#       name: angie-etc
#     - mountPath: /var/cache/angie
#       name: angie-cache
#     - mountPath: /var/lib/angie
#       name: angie-lib
#     - mountPath: /var/log/angie
#       name: angie-log
env:
- name: POD_NAMESPACE
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
- name: POD_NAME
  valueFrom:
    fieldRef:
      fieldPath: metadata.name

```

```

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

```

- **DaemonSet**: используйте этот тип, если планируете 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

```



```

    containerPort: 9113
  readinessProbe:
    httpGet:
      path: /angie-ready
      port: readiness-port
    periodSeconds: 1
  resources:
    requests:
      cpu: "100m"
      memory: "128Mi"
  env:
  - name: POD_NAMESPACE
    valueFrom:
      fieldRef:
        fieldPath: metadata.namespace
  - name: POD_NAME
    valueFrom:
      fieldRef:
        fieldPath: metadata.name
  args:
  - -angie-configmaps=$(POD_NAMESPACE)/angie-config
  #- -default-server-tls-secret=$(POD_NAMESPACE)/default-server-secret
  #- -include-year
  #- -v=3 # Enables extensive logging. Useful for troubleshooting.
  #- -report-ingress-status
  #- -external-service=angie-ingress
  #- -enable-prometheus-metrics
  #- -global-configuration=$(POD_NAMESPACE)/angie-configuration

```

EOF

ГЛАВА 3

Установка с помощью Helm

3.1 Вступление

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

3.2 Предварительные требования

i Примечание

Вся документация должна использоваться только с последней стабильной версией ANIC.

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

3.3 Пользовательские определения ресурсов

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

3.3.1 Скачивание диаграммы

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

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

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 ANIC.

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 ANIC 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 0.5.0
```

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 ANIC.	Создается автоматически
<code>controller</code>	Тип установки ANIC - <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 ANIC для Angie PRO.	<code>false</code>
<code>controller</code>	Время ожидания в миллисекундах, в течение которого ANIC будет ожидать успешной перезагрузки Angie после изменения или при начальном запуске.	<code>60000</code>
<code>controller</code>	Позволяет подам ANIC использовать сетевое пространство имен хоста.	<code>false</code>
<code>controller</code>	Политика DNS для подов ANIC.	<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 ANIC.	<code>1</code>
<code>controller</code>	Дайджест образа ANIC.	<code>Her</code>
<code>controller</code>	Репозиторий образов ANIC.	<code>myregistry.host.ru/angie-ingress</code>
<code>controller</code>	Тег образа ANIC.	<code>0.1.2</code>
<code>controller</code>	Политика скачивания образа ANIC.	<code>IfNotPresent</code>
<code>controller</code>	Жизненный цикл подов ANIC.	<code>{}</code>
<code>controller</code>	Имя пользовательской ConfigMap, используемой ANIC. Если имя задано, то конфигурация по умолчанию игнорируется.	<code>«»</code>

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<code>controller</code>	Имя ConfigMap, используемой ANIC.	Создается автоматически
<code>controller</code>	Аннотации к ConfigMap в ANIC.	{}
<code>controller</code>	Записи в ConfigMap для настройки конфигурации Angie.	{}
<code>controller</code>	Список пользовательских портов, которые должны быть доступны в поде ANIC. Следует обычному синтаксису yaml Kubernetes для контейнерных портов.	[]
<code>controller</code>	Сертификат TLS в кодировке base64 для сервера HTTPS по умолчанию.	Нет
<div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>Примечание</p> <p>Рекомендуется указать свой собственный сертификат. Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p> </div>		
<code>controller</code>	Ключ TLS в кодировке base64 для сервера HTTPS по умолчанию.	Нет
<div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>Примечание</p> <p>Рекомендуется указать свой собственный ключ. Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p> </div>		
<code>controller</code>	Секрет с сертификатом TLS и ключом для сервера HTTPS по умолчанию. Значение должно соответствовать следующему формату: <code><пространство имен>/<имя></code> . Используется в качестве альтернативы указанию сертификата и ключа с помощью параметров <code>controller.defaultTLS.cert</code> и <code>controller.defaultTLS.key</code> .	Нет
<div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff;"> <p>Примечание</p> <p>Альтернативное решение: полный пропуск секрета сервера по умолчанию приведет к тому, что Angie будет по умолчанию отклонять TLS-подключения к серверу.</p> </div>		
<code>controller</code>	Сертификат TLS в кодировке base64 для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Если параметр не задан, Angie прервет любую попытку установить TLS-соединение для таких узлов Ingress или VirtualServer.	Нет
<code>controller</code>	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
<code>controller</code>	Секрет с сертификатом TLS и ключом для каждого узла Ingress или VirtualServer, у которого включен TLS, но не указан секрет. Значение должно соответствовать следующему формату: <code><пространство имен>/<имя></code> . Используется в качестве альтернативы указанию сертификата и ключа с помощью параметров <code>controller.wildcardTLS.cert</code> и <code>controller.wildcardTLS.key</code> .	Нет
<code>controller</code>	Селектор узлов для назначения подов ANIC.	{}
<code>controller</code>	Период плавного завершения работы пода ANIC.	30
<code>controller</code>	Допуски подов ANIC.	[]

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<code>controller.</code>	Привязка подов ANIC.	{}
<code>controller.</code>	Ограничения на распространение топологии подов ANIC.	{}
<code>controller.</code>	Дополнительные переменные окружения, которые должны быть установлены на подах ANIC.	{}
<code>controller.</code>	Тома подов ANIC.	{}
<code>controller.</code>	Точки подключения томов подов ANIC.	{}
<code>controller.</code>	Значение <code>initContainers</code> для подов ANIC.	{}
<code>controller.</code>	Дополнительные контейнеры (например, сайдкап) для подов Ingress Controller.	{}
<code>controller.</code>	Ресурсы подов ANIC.	<code>requests: cpu=100m,memory=128M</code>
<code>controller.</code>	Количество реплик развертывания ANIC.	1
<code>controller.</code>	Класс ANIC. Должен быть развернут ресурс <code>IngressClass</code> с именем, тождественным этому классу. В противном случае ANIC не запустится. ANIC обрабатывает только те ресурсы, которые принадлежат его классу, т. е. их ресурс поля <code>ingressClassName</code> тождествен классу. ANIC обрабатывает все ресурсы <code>VirtualServer</code> , <code>VirtualServerRoute</code> и <code>TransportServer</code> , которые не имеют поля <code>ingressClassName</code> , во всех версиях Kubernetes.	<code>angie</code>
<code>controller.</code>	Новым Ingress без указанного поля <code>ingressClassName</code> будет присвоен класс, указанный в <code>controller.ingressClass</code> .	<code>false</code>
<code>controller.</code>	Разделенный запятыми список пространств имен, за ресурсами которых должен следить ANIC. По умолчанию ANIC 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=>default,anic</code> .	«»
<code>controller.</code>	Настраивает в ANIC просмотр только пространств имен с меткой <code>foo=bar</code> . По умолчанию ANIC отслеживает все пространства имен. Взаимоисключающая с <code>controller.watchNamespace</code> настройка.	«»
<code>controller.</code>	Разделенный запятыми список пространств имен, за которыми Ingress Controller должен следить в поисках ресурсов типа <code>Secret</code> . Если этот параметр не настроен, ANIC 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=>default,angie-ingress</code> .	«»
<code>controller.</code>	Включает пользовательские ресурсы.	<code>true</code>
<code>controller.</code>	Включает передачу данных по протоколу TLS на порту 443. Требуется <code>controller.enableCustomResources</code> .	<code>false</code>
<code>controller.</code>	Включает автоматическое управление сертификатами x509 для ресурсов виртуального сервера с помощью <code>cert-manager</code> (<code>cert-manager.io</code>). Требуется <code>controller.enableCustomResources</code> .	<code>false</code>
<code>controller.</code>	Включает интеграцию с <code>ExternalDNS</code> для настройки общедоступных записей DNS у ресурсов <code>VirtualServer</code> с использованием <code>ExternalDNS</code> . Требуется <code>controller.enableCustomResources</code> .	<code>false</code>
<code>controller.</code>	Создает пользовательский ресурс <code>GlobalConfiguration</code> . Требуется <code>controller.enableCustomResources</code> .	<code>false</code>
<code>controller.</code>	Спецификация <code>GlobalConfiguration</code> для определения параметров глобальной конфигурации ANIC.	{}

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<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 ANIC.	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 ANIC.	Нет
<i>controller</i>	Создаёт сервис для предоставления доступа к подам ANIC.	true
<i>controller</i>	Тип сервиса, который необходимо создать для ANIC.	LoadBalancer
<i>controller</i>	Внешняя политика трафика сервиса. Значение Local сохраняет исходный IP-адрес клиента.	Local
<i>controller</i>	Аннотации сервиса ANIC.	{}
<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 ANIC.	[]
<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 ANIC. Следует обычному синтаксису yaml Kubernetes для портов сервиса.	[]
<i>controller</i>	Включает HTTP-порт для сервиса ANIC.	true
<i>controller</i>	HTTP-порт сервиса ANIC.	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 ANIC.	80
<i>controller</i>	Включает порт HTTPS для сервиса ANIC.	true
<i>controller</i>	HTTPS-порт сервиса ANIC.	443
<i>controller</i>	Пользовательский NodePort для HTTPS-порта. Для <i>controller.service.type</i> должно быть установлено значение <i>NodePort</i> .	«»
<i>controller</i>	Целевой порт HTTPS-порта сервиса ANIC.	443
<i>controller</i>	Аннотации учетной записи сервиса ANIC.	{}
<i>controller</i>	Имя учетной записи сервиса подов ANIC. 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.	«»

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controller</i>	Имя <code>serviceMonitor</code> .	Создается автоматически
<i>controller</i>	Создает пользовательский ресурс <code>ServiceMonitor</code> .	false
<i>controller</i>	Метки объектов Kubernetes для применения к объекту <code>serviceMonitor</code> .	«»
<i>controller</i>	Набор меток, позволяющих выбирать конечные точки для <code>serviceMonitor</code> .	«»
<i>controller</i>	Список конечных точек, разрешенных в составе этого <code>serviceMonitor</code> .	«»
<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 <code>controller.reportIngressStatus.ExternalService</code> , либо через <code>controller.reportIngressStatus.ingressLink</code> , либо через запись <code>external-status-address</code> в ConfigMap через <code>controller.config.entries</code> .	true
<p>Примечание</p> <p>Значение <code>controller.config.entries.external-status-address</code> имеет приоритет над остальными.</p>		
<i>controller</i>	Указывает имя сервиса с типом <code>LoadBalancer</code> , через который Ingress Controller будет доступен извне. Внешний адрес сервиса используется для отчетов о состоянии ресурсов Ingress, <code>VirtualServer</code> и <code>VirtualServerRoute</code> . Значение <code>controller.reportIngressStatus.enable</code> должно быть задано как <code>true</code> . Значение по умолчанию создается автоматически и включается, когда <code>controller.service.create</code> имеет значение <code>true</code> , а <code>controller.service.type</code> - значение <code>LoadBalancer</code> .	Создается автоматически
<i>controller</i>	Указывает имя ресурса <code>IngressLink</code> , который предоставляет доступ к подам ANIC через систему BIG-IP. IP-адрес системы BIG-IP используется для отчетов о состоянии ресурсов Ingress, <code>VirtualServer</code> и <code>VirtualServerRoute</code> . Значение <code>controller.reportIngressStatus.enable</code> должно быть задано как <code>true</code> .	«»
<i>controller</i>	Включает выбор лидера, чтобы избежать ситуации, когда несколько реплик контроллера сообщают о состоянии ресурсов Ingress. Значение <code>controller.reportIngressStatus.enable</code> должно быть задано как <code>true</code> .	true
<i>controller</i>	Указывает имя ConfigMap в том же пространстве имен, что и контроллер, которое используется для блокировки выбора лидера. Значение <code>controller.reportIngressStatus.enableLeaderElection</code> должно быть задано как <code>true</code> .	Создается автоматически
<i>controller</i>	Аннотации к конфигурационной карте выборов лидера.	{}
<i>controller</i>	Аннотации пода ANIC.	{}
<i>controller</i>	Дополнительные экстра-метки для пода ANIC.	{}
<i>controller</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 ANIC, которая использует конечную точку готовности.	true
<i>controller</i>	HTTP-порт для конечной точки готовности.	8081
<i>controller</i>	Число секунд с запуска пода ANIC до инициирования проверки готовности.	0
<i>controller</i>	Включает сбор метрик задержки для апстримов. Требуется <code>prometheus.create</code> .	false
<i>controller</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. здесь .	0

продолжается на следующей странице

Таблица 1 – продолжение с предыдущей страницы

Параметр	Описание	По умолчанию
<i>controller</i>	Включает HorizontalPodAutoscaling.	false
<i>controller</i>	Аннотации HorizontalPodAutoscaler для ANIC.	{}
<i>controller</i>	Минимальное число реплик для НРА.	1
<i>controller</i>	Максимальное число реплик для НРА.	3
<i>controller</i>	Целевой процент загрузки ЦП.	50
<i>controller</i>	Целевой процент использования памяти.	50
<i>controller</i>	Включает PodDisruptionBudget.	false
<i>controller</i>	Аннотации к бюджету сбоя пода ANIC.	{}
<i>controller</i>	Количество подов ANIC, которые должны быть доступны. Взаимоисключающая с «maxUnavailable» настройка.	0
<i>controller</i>	Количество подов ANIC, которые могут быть недоступны. Взаимоисключающая с «minAvailable» настройка.	0
<i>controller</i>	Задаёт стратегию замены старых подов новыми. Документация по стратегии обновления развертывания и стратегии обновления набора демонов	{}
<i>controller</i>	В явной форме отключает прослушиватели IPV6 для узлов, которые не поддерживают стек IPV6.	false
<i>controller</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 ANIC:

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

Ниже 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 удастся прочитать файл `/etc/angie/secrets/default`, то ANIC 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>

Задает 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 -enable-oidc

Включает функцию аутентификации по OpenID Connect в ресурсах Policy.

По умолчанию `false`.

4.11 -enable-jwt

Включает функцию аутентификации JWT в ресурсах Policy.

По умолчанию `false`.

4.12 -external-service <строка>

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

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

4.13 -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*.

4.14 -global-configuration <строка>

Ресурс GlobalConfiguration для глобальной настройки ANIC.

Формат: <пространство имен>/<имя>

Требует наличия *-enable-custom-resources*.

4.15 -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.

4.16 -health-status-uri <строка>

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

По умолчанию /angie-health.

4.17 -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`.

4.18 -ingress-template-path <строка>

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

4.19 -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*.

4.20 -main-template-path <строка>

Путь к основному шаблону конфигурации Angie.

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

4.21 `-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>

4.22 `-angie-debug`

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

4.23 `-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.

4.24 `-angie-status`

Включает Angie `stub_status`.

По умолчанию `true`.

4.25 `-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`)

4.26 `-angie-status-port` <int>

Задаёт порт, на котором доступен Angie `stub_status`.

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

4.27 `-angie-status-prometheus` <bool>

Включает или отключает выдачу статистики Angie в формате Prometheus.

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

4.28 `-angie-status-prometheus-path` <строка>

Позволяет менять путь для публикации статистики Angie в формате Prometheus.

По умолчанию используется `/p8s`.

4.29 `-angie-status-prometheus-port` <int>

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

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

4.30 `-angie-status-prometheus-allow-cidrs`

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

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

4.31 `-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.

4.32 `-report-ingress-status`

Обновляет поле адреса в статусе ресурсов Ingress.

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

4.33 `-transportserver-template-path` <строка>

Путь к шаблону конфигурации TransportServer Angie для ресурса TransportServer.

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

4.34 `-v` <значение>

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

4.35 `-version`

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

4.36 `-virtualserver-template-path` <строка>

Путь к шаблону конфигурации VirtualServer Angie для ресурса VirtualServer.

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

4.37 `-vmodule` <значение>

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

4.38 `-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`».

4.39 `-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`».

4.40 `-watch-secret-namespace` <строка>

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

4.41 `-enable-prometheus-metrics`

Позволяет публиковать метрики Angie в формате Prometheus.

4.42 `-prometheus-metrics-listen-port` <int>

Задает порт, на котором публикуются метрики Prometheus.

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

4.43 `-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.

4.44 `-enable-service-insight`

Публикует конечную точку Service Insight для ANIC.

4.45 `-service-insight-listen-port` <int>

Задаёт порт, на котором публикуется Service Insight.

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

4.46 `-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>

4.47 `-ready-status`

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

По умолчанию `true`.

4.48 `-ready-status-port`

HTTP-порт для конечной точки готовности.

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

4.49 -disable-ipv6

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

По умолчанию `false`.