

Примеры использования Logging Operator

Раздел содержит примеры использования Logging Operator.

1. Отправка логов из Syslog-NG в Opensearch

В данном пункте рассказывается как настроить отправку логов из Syslog-NG в Opensearch, используя компонент Logging Operator.

Syslog-NG имеет модуль для отправки логов в ElasticSearch. Поскольку OpenSearch является ответвлением ElasticSearch, этот модуль также можно использовать для отправки логов в OpenSearch. OpenSearch поддерживает авторизацию по TLS, поэтому вы можете использовать сертификаты для настройки Output, но также доступна авторизация по имени пользователя и паролю.

1.1. Предварительные условия

- ✓ Вы ознакомились с архитектурой и концепциями Logging operator в Nova Container Platform .
- ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль cluster-admin в Kubernetes.
- ✓ Вы установили утилиту kubectl для работы с Kubernetes.
- ✓ Вы создали отдельное Namespace для компонентов Logging operator. Например, logging .

1.2. Добавление секрета

1. В Nova Console перейдите на вкладку **Workloads > Secrets** и выберите Namespace nova-logs .
2. Найдите секрет nova-opensearch и скопируйте его в Namespace, где установлен Syslog-NG (например, logging). Этот секрет содержит сертификаты, необходимые для авторизации в Opensearch.

1.3. Установка Output для Syslog-ng

Установите Output через kubectl или через Nova Console, используя следующий манифест:

```

apiVersion: logging.banzaicloud.io/v1beta1
kind: SyslogNGClusterOutput ①
metadata:
  name: syslogngoutput
  namespace: logging
spec:
  elasticsearch:
    logstash_prefix: nova-syslog ②
    url: https://nova-logs-cluster.nova-logs.svc.cluster.local:9200/_bulk
    tls:
      ca_file:
        mountFrom:
          secretKeyRef:
            key: ca.crt
            name: nova-opensearch
      cert_file:
        mountFrom:
          secretKeyRef:
            key: tls.crt
            name: nova-opensearch
      key_file:
        mountFrom:
          secretKeyRef:
            key: tls.key
            name: nova-opensearch
    peer_verify: false ③

```

① Происходит установка Output для всего кластера (Global Output).

② Имя индекса в Opensearch, куда будут отправляться логи. Индекс создастся автоматически.

③ Позволяет Logging Operator работать с недоверенными сертификатами.

1.4. Установка Flow для Syslog-ng

Установите Flow через `kubectl` или через Nova Console, используя следующий манифест:

```

apiVersion: logging.banzaicloud.io/v1beta1
kind: SyslogNGClusterFlow ①
metadata:
  name: syslogngflow
  namespace: logging
spec:
  globalOutputRefs:
    - syslogngoutput
  match:
    regexp: ②
    pattern: '*'

```

```
type: glob  
value: json.kubernetes.labels.app.kubernetes.io/instance
```

- ① Происходит установка Flow для всего кластера.
- ② Логи будут собираться со всех экземпляров и со всех Namespace.

1.5. Проверка в OpenSearch

1. Откройте консоль OpenSearch и перейдите на вкладку **Management > Index Management > Indices**.
2. Найдите индекс с именем `nova-syslog-data`. Если индекс не отображается, подождите несколько минут, так как создание индекса может занять некоторое время.

2. Настройка отправки логов из NeuVector на внешний сервер Syslog

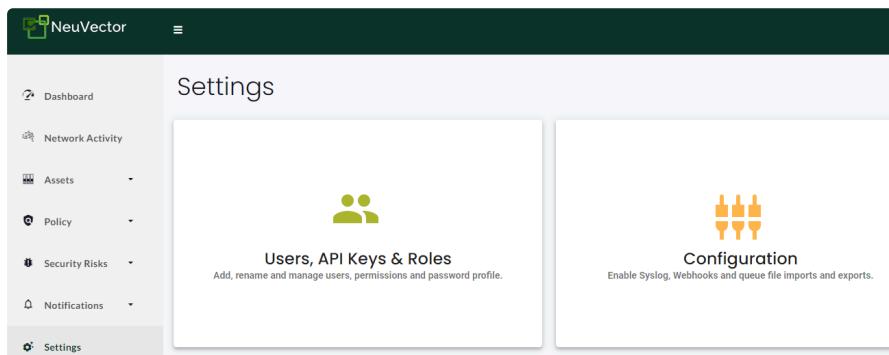
В данной статье будет рассматриваться отправка логов из NeuVector на внешний сервер Syslog средствами самого NeuVector.

2.1. Предварительные условия

- ✓ Вы ознакомились с архитектурой и концепциями NeuVector в Nova Container Platform.
- ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль `cluster-admin` в Kubernetes.
- ✓ Вы установили утилиту `kubectl` для работы с Kubernetes.

2.2. Настройка NeuVector

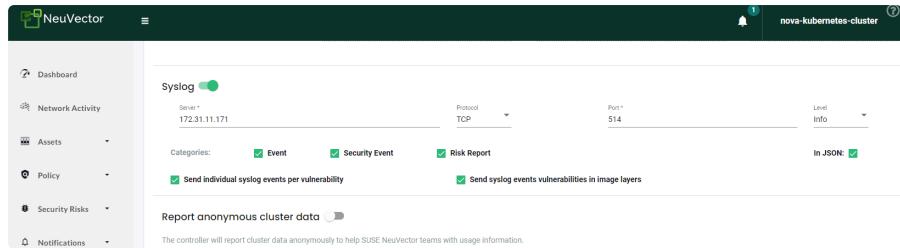
1. Откройте консоль NeuVector и перейдите на вкладку **Settings > Configuration**.



2. Найдите блок **Syslog** в нижней части страницы. Настройте параметры в соответствии с настройками внешнего Syslog сервера. В данном случае оставлен стандартный порт,

выбран протокол TCP.

Выберите нужные параметры. В данном случае настроено получение всех логов с сервера, а также активирована опция `In JSON`, которая позволяет получать данные в JSON формате.



3. После настройки сгенерируйте событие в логе или просмотрите логи на внешнем сервере Syslog. В приведённом примере было настроено правило, запрещающее установку pod'a, если образ не подписан через cosign, и была сделана попытка создания pod'a.

3. Настройка отправки логов из Neuvector в OpenSearch

В данной разделе описывается процесс отправки логов из NeuVector в OpenSearch с использованием Syslog-*ng* и Fluentd через компонент Logging Operator.

3.1. Предварительные условия

- ✓ Вы ознакомились с архитектурой и концепциями Logging operator в Nova Container Platform .
 - ✓ Вы ознакомились с архитектурой и концепциями NeuVector в Nova Container Platform .
 - ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль cluster-admin в Kubernetes.
 - ✓ Вы установили утилиту kubectl для работы с Kubernetes.
 - ✓ Создано новое Namespace. Например, logging .

3.2. Отправка логов через Fluentd

3.2.1. Включаем запись событий из Neuvector в логи pod'a контроллера

1. Откройте консоль Neuvector.
2. Перейдите на вкладку **Settings > Configuration** и раскройте блок **Notification Configuration**.
3. Включите опцию *Write notifications to controller pod logs* и укажите необходимые опции.

3.2.2. Настройка Fluentd

Создайте новый Fluentd или настройте уже существующий согласно [инструкции](#). В данном случае был настроен уже существующий Fluentd.



Рекомендуется создать новый Fluentd, чтобы не нарушить работу встроенного компонента.

3.2.3. Установка Output

Установите Output согласно [инструкции](#) через `kubectl` или через Nova Console, используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: ClusterOutput ①
metadata:
  name: os-output
  namespace: nova-logs
spec:
  opensearch:
    buffer:
      flush_interval: 30s
      path: /buffers/neuvector
      retry_timeout: 96h
      retry_wait: 2s
      chunk_limit_records: 1000000
      timekey: 1h
      flush_thread_count: 2
      delayed_commit_timeout: 150s
      retry_max_interval: 180s
      chunk_limit_size: 300K
      flush_mode: interval
    logstash_prefix: nova-neuvector ②
    port: 9200
    logstash_format: true
    scheme: https
```

[YAML](#) | [JSON](#)

```
host: nova-logs-cluster.nova-logs.svc.cluster.local
user: nova-logging
ssl_verify: false
reload_on_failure: true
reconnect_on_error: true
password:
  valueFrom:
    secretKeyRef:
      key: password
      name: nova-logging-credentials
include_timestamp: true
```

- ① Происходит установка Output для всего кластера.
- ② Указывается имя индекса в Opensearch, куда будут направляться логи. Индекс создаётся автоматически.

3.2.4. Установка Flow

Установите Flow через `kubectl` или через Nova Console используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: ClusterFlow ①
metadata:
  name: os-flow
  namespace: nova-logs
spec:
  globalOutputRefs:
  - os-output
  match:
  - select:
    labels:
      app: neuvendor-controller-pod ②
```

YAML | ▾

- ① Происходит установка Flow для всего кластера.
- ② В Opensearch будут отправляться только логи с контейнера `neuvendor-controller-pod`.

3.2.5. Проверка в OpenSearch

1. Откройте консоль OpenSearch и перейдите на вкладку **Management > Index Management > Indices**.
2. Найдите индекс с именем `nova-neuvendor-data`. Если такого индекса ещё нет, то подождите несколько минут.

3.3. Отправка логов через Syslog-ng

3.3.1. Включение записи событий из Neuvendor в логи pod'a контроллера

1. Откройте консоль Neuvector.
2. Перейдите на вкладку **Settings > Configuration** и раскройте блок **Notification Configuration**.
3. Включите опцию *Write notifications to controller pod logs* и укажите необходимые опции.

3.3.2. Настройка Syslog-ng

Создайте Syslog-ng согласно [инструкции](#).

3.3.3. Добавление секрета

1. В Nova Console перейдите на вкладку **Workloads > Secrets** и Namespace `nova-logs`.
2. Найдите секрет `nova-opensearch` и скопируйте его в Namespace, где установлен Syslog-ng. В данном случае - `logging`. Этот секрет содержит сертификаты и ключи, которые нужны для авторизации в Opensearch.

3.3.4. Установка Output

Установите Output через `kubectl` или через Nova Console, используя следующий манифест:

```
YAML | □
apiVersion: logging.banzaicloud.io/v1beta1
kind: SyslogNGClusterOutput
metadata:
  name: random-log-output
  namespace: logging
spec:
  elasticsearch:
    logstash_prefix: nova-neuvector
    url: https://nova-logs-cluster.nova-logs.svc.cluster.local:9200/_bulk
    tls:
      ca_file:
        mountFrom:
          secretKeyRef:
            key: ca.crt
            name: nova-opensearch
      cert_file:
        mountFrom:
          secretKeyRef:
            key: tls.crt
            name: nova-opensearch
```

```
key_file:  
  mountFrom:  
    secretKeyRef:  
      key: tls.key  
      name: nova-opensearch  
  peer_verify: false
```

3.3.5. Установка Flow

Установите Flow через `kubectl` или через Nova Console, используя **любой** из следующих манифестов:

```
apiVersion: logging.banzaicloud.io/v1beta1  
kind: SyslogNGClusterFlow  
metadata:  
  name: random-log-flow  
  namespace: logging  
spec:  
  filters:  
    - match:  
        and:  
          - regexp:  
              pattern: notification  
          - regexp:  
              pattern: neuvector-controller-pod-0  
  globalOutputRefs:  
    - random-log-output
```

YAML | □



В этом манифесте происходит установка Flow для всего кластера. В Opensearch будут отправляться только логи, которые содержат слова `notification` и `neuvector-controller-pod-0`.

```
apiVersion: logging.banzaicloud.io/v1beta1  
kind: SyslogNGClusterFlow  
metadata:  
  name: random-log-flow  
  namespace: logging  
spec:  
  globalOutputRefs:  
    - random-log-output  
  match:  
    - regexp:  
        pattern: neuvector-controller-pod  
        type: glob  
        value: json.kubernetes.labels.app
```

YAML | □

 В этом манифесте происходит установка Flow для всего кластера. В OpenSearch будут отправляться только логи с контейнера `neuvector-controller-pod`, у которого есть соответствующая метка.

3.3.6. Проверка в OpenSearch

1. Откройте консоль OpenSearch и перейдите на вкладку **Management > Index Management > Indices**.
2. Найдите индекс с именем `nova-neuvector-data`. Если такого индекса ещё нет, то подождите несколько минут.

4. Настройка отправки логов kubelet в OpenSearch

4.1. Предварительные условия

- ✓ Вы ознакомились с архитектурой и концепциями Logging operator в Nova Container Platform.
- ✓ У вас есть доступ к кластеру с учетной записью, имеющей роль `cluster-admin` в Kubernetes.
- ✓ Вы установили утилиту `kubectl` для работы с Kubernetes.

4.2. Настройка

1. Создайте агенты Fluent Bit, которые будут собирать логи сервиса kubelet.

```
apiVersion: logging-extensions.banzaicloud.io/v1alpha1
kind: HostTailer
metadata:
  name: nova-node-kubelet-host-tailer
  namespace: nova-logs
spec:
  systemdTailers:
    - containerOverrides:
        image: 'hub.nova-platform.io/registry/fluent/fluent-bit:2.1.8'
        name: nova-node-kubelet
        systemdFilter: kubelet.service ①
  workloadMetaOverrides:
    labels:
      nova-log-type: nova-node-kubelet ②
  workloadOverrides:
    tolerations:
      - operator: Exists
```

- ① Фильтр по имени сервиса.
- ② Метка, которая будет присутствовать на созданных ресурсом pod'ах.

2. Создайте точку выхода для логов.

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: Output
metadata:
  name: nova-node-kubelet-output
  namespace: nova-logs
spec:
  opensearch:
    buffer:
      flush_interval: 30s
      path: /buffers/kubelet ①
      retry_timeout: 96h
      retry_wait: 2s
      chunk_limit_records: 1000000
      timekey: 1h
      flush_thread_count: 2
      delayed_commit_timeout: 150s
      retry_max_interval: 180s
      chunk_limit_size: 300K
      flush_mode: interval
    logstash_prefix: nova-node-kubelet ②
    port: 9200
    logstash_format: true
    scheme: https
    host: nova-logs-cluster.nova-logs.svc.cluster.local ③
    user: nova-logging ④
    ssl_verify: false
    reload_on_failure: true
    reconnect_on_error: true
    password: ⑤
    valueFrom:
      secretKeyRef:
        key: password
        name: nova-logging-credentials
  include_timestamp: true
```

YAML | □

- ① Путь до буфера.
- ② Имя нового индекса в Opensearch.
- ③ Адрес Opensearch.
- ④ Имя пользователя в Opensearch.
- ⑤ Пароль пользователя в Opensearch.

3. Создайте правило фильтрации логов для отправки их в Opensearch.

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: Flow
metadata:
  name: nova-node-kubelet-flow
  namespace: nova-logs
spec:
  localOutputRefs:
    - nova-node-kubelet-output ①
  match:
    - select:
        labels:
          nova-log-type: nova-node-kubelet ②
```

① Точка выхода для отфильтрованных логов.

② Метка для pod'ов, с которых будут собираться логи.

Установка Logging Operator

Logging Operator уже установлен в Nova Container Platform. Для начала работы с логированием необходимо установить следующие компоненты:

- Fluent Bit Agent
- Syslog-ng или FluentD

Примечание

- Выбор между Syslog-ng и FluentD зависит от требований по передаче логов. Можно развернуть любое количество экземпляров этих сервисов, но каждый из них должен находиться в отдельном Namespace.
- При установке модуля OpenSearch - Fluent Bit Agent и FluentD устанавливаются автоматически.

1. Предварительные условия

- ✓ Вы ознакомились с архитектурой и концепциями Logging operator в Nova Container Platform.
- ✓ У вас есть доступ к кластеру с учетной записью, обладающей ролью `cluster-admin` в Kubernetes.
- ✓ Вы установили утилиту `kubectl` для работы с Kubernetes.
- ✓ Вы создали отдельное Namespace для компонентов Logging operator. Например, `logging`.

2. (Опционально) Установка Fluent Bit Agent

Установите Fluent Bit Agent через `kubectl` или через Nova Console используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: FluentbitAgent
metadata:
  name: nova-logs-fluentbit
  namespace: nova-logs
spec:
  tolerations:
```

YAML |

```
- operator: Exists
filterKubernetes:
  Kube_URL: "https://kubernetes.default.svc.cluster.local:443"
  tls.verify: "Off"
inputTail:
  Refresh_Interval: "60"
  Rotate_Wait: "5"
```



Представленный манифест выполняет только предварительную настройку агентов, а не их установку. Агенты устанавливаются автоматически лишь после установки Syslog-*ng* или FluentD.

3. Настройка сбора логов через *Syslog-*ng**

В этой части статьи описан процесс настройки отправки логов через *Syslog-*ng** на внешний *Syslog* сервер.

3.1. Установка *Syslog-*ng**

Установите *Syslog-*ng** через `kubectl` или через Nova Console, используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: Logging
metadata:
  name: syslogng
  namespace: logging
spec:
  controlNamespace: logging
  allowClusterResourcesFromAllNamespaces: true
  syslogNG: {}
```

YAML | □



Опция `allowClusterResourcesFromAllNamespaces` добавляется, если требуется создание единой системы логирования в кластере(Fluentd или *Syslog-*ng**). Это позволяет использовать свои Output и Flow в каждом Namespace кластера.

3.2. Установка Output для *Syslog-*ng**

Установите Output через `kubectl` или через Nova Console, используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: SyslogNGClusterOutput ①
```

YAML | □

```
metadata:
  name: random-log-output
  namespace: logging
spec:
  syslog:
    host: 10.251.19.20 ②
    port: 514 ③
    transport: tcp ④
```

- ① Происходит установка Output для всего кластера (Global Output).
② Полное имя или IP адрес Syslog сервера.
③ Порт для подключения к Syslog серверу.
④ Протокол, указанный в настройках Syslog сервера.

3.3. Установка Flow для Syslog-ng

Установите Flow через `kubectl` или через Nova Console используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: SyslogNGClusterFlow
metadata:
  name: random-log-flow
  namespace: logging
spec:
  globalOutputRefs:
  - random-log-output
match:
  regexp:
    value: json.kubernetes.labels.app.kubernetes.io/instance
    pattern: "*"
    type: glob
```



В этом манифесте происходит установка Flow для всего кластера. Логи будут собираться со всех экземпляров и из всех Namespace.

3.4. Проверка

Убедитесь, что логи успешно отображаются на Syslog сервере.

4. Настройка сбора логов через FluentD

В этой части статьи описан процесс настройки отправки логов через FluentD на внутренний OpenSearch.

4.1. Установка FluentD

При установке модуля OpenSearch, автоматически устанавливается FluentD. В дальнейшем можно использовать его. Однако, для более гибкой настройки и управления логами, **рекомендуется** развернуть отдельный экземпляр FluentD.

► Установка нового FluentD

► Использование установленного FluentD

4.2. Установка Output для FluentD

Можно настроить отправку логов как на внутренний, так и на внешний OpenSearch, а также в другой сборщик логов.

Все варианты настройки можно найти в документации по [CustomResourceDefinitions](#) `Output` или `ClusterOutput`.

В данном случае настроим отправку логов на внутренний OpenSearch.

Авторизация в OpenSearch возможна как по имени и паролю, так и по сертификату и ключу. Для авторизации по имени и паролю потребуется секрет `nova-logging-credentials` из Namespace `nova-logs`.

Для авторизации по сертификату и ключу потребуется секрет `nova-opensearch` из Namespace `nova-logs`.

В данном примере будет использоваться авторизация по имени и паролю.

1. Скопируйте секрет `nova-logging-credentials` из Namespace `nova-logs` в `logging`.
2. Установите Output через `kubectl` или через Nova Console, используя следующий манифест:

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: ClusterOutput ①
metadata:
  name: os-output
  namespace: logging
spec:
  opensearch:
    buffer:
      flush_interval: 30s
      path: /buffers/all-logs
      retry_timeout: 96h
      retry_wait: 2s
      chunk_limit_records: 1000000
      timekey: 1h
```

YAML | □

```

flush_thread_count: 2
delayed_commit_timeout: 150s
retry_max_interval: 180s
chunk_limit_size: 300K
flush_mode: interval
logstash_prefix: nova-node-all ②
port: 9200
logstash_format: true
scheme: https
host: nova-logs-cluster.nova-logs.svc.cluster.local
user: nova-logging
ssl_verify: false
reload_on_failure: true
reconnect_on_error: true
password:
  valueFrom:
    secretKeyRef:
      key: password
      name: nova-logging-credentials
include_timestamp: true

```

- ① Происходит установка Output для всего кластера.
- ② Указывается имя индекса в OpenSearch, куда будут переданы логи. Индекс создастся автоматически.

4.3. Установка Flow для FluentD

Установите Flow через `kubectl` или через Nova Console, используя следующий манифест:

```

apiVersion: logging.banzaicloud.io/v1beta1
kind: ClusterFlow ①
metadata:
  name: os-flow
  namespace: logging
spec:
  globalOutputRefs:
    - os-output
  match:
    - select:
        namespaces: ["test","test-syslog"] ②

```

[YAML](#) | [JSON](#)

- ① Происходит установка Flow для всего кластера.
- ② Логи будут собираться со всех экземпляров в указанных Namespace'x.

4.4. Проверка

Убедитесь, что логи появились в OpenSearch.

2025 orionsoft. Все права защищены.

Custom Resource Definitions

В API Kubernetes *resource* — это конечная точка, которая хранит коллекцию объектов API определенного типа. Например, встроенный ресурс *Pods* содержит коллекцию объектов *Pod*.

Custom resource definition (CRD) определяет новый уникальный тип объекта в кластере и позволяет серверу API Kubernetes управлять всем его жизненным циклом.

Объекты *custom resource (CR)* создаются из *CRDs*, добавленных в кластер администратором, что позволяет всем пользователям кластера добавлять новый тип ресурса в проекты.



Хотя создавать *CRDs* могут только администраторы кластера, разработчики могут создавать *Custom Resources* из существующего *CRD*, если у них есть разрешение на чтение и запись.

1. Custom Resource Definitions в Nova Container Platform

В Nova Container Platform *CRDs* представлены в веб-консоли на вкладке **Administration > CustomResourceDefinitions**.

На этой странице можно найти *Custom Resources* и созданные для него экземпляры. Например, в [статье](#) по установке Logging operator описывается добавление ресурса `_Syslog-ng`, в манифесте которого можно увидеть `apiVersion` и `kind`.

```
apiVersion: logging.banzaicloud.io/v1beta1
kind: Logging
```

[YAML](#) | [Edit](#)

В поиске на странице **CustomResourceDefinitions** найдите нужный ресурс, используя значение `kind` из манифеста.

CustomResourceDefinitions

Имя	Группа	Последняя версия	Находится в Namespace	Установлено
<code>ClusterFlow</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>ClusterOutput</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>EventTailer</code>	<code>logging-extensions.banzaicloud.io</code>	v1alpha1	Нет	●
<code>Flow</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>FluentBitAgent</code>	<code>logging.banzaicloud.io</code>	v1alpha1	Нет	●
<code>HostTailer</code>	<code>logging-extensions.banzaicloud.io</code>	v1alpha1	Да	●
<code>Logging</code>	<code>logging.banzaicloud.io</code>	v1beta1	Нет	●
<code>LoggingRoute</code>	<code>logging.banzaicloud.io</code>	v1beta1	Нет	●
<code>NodeAgent</code>	<code>logging.banzaicloud.io</code>	v1beta1	Нет	●
<code>Output</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>SyslogNGClusterFlow</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>SyslogNGClusterOutput</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●
<code>SyslogNGFlow</code>	<code>logging.banzaicloud.io</code>	v1beta1	Да	●

Также убедитесь, что в колонке *Группа* значение соответствует значению `apiVersion` из манифеста. Перейдите в этот ресурс и откройте вкладку **Экземпляры**. На этой странице можно увидеть все экземпляры выбранного ресурса, а также создать новый экземпляр.

CustomResourceDefinition > CustomResourceDefinition детали

`Logging` logging.logging.banzaicloud.io

Детали YAML Экземпляры

Создать Logging

Имя	Название	Дата создания
<code>nova-logs-logging</code>	Отсутствует	24 мая 2024 г., 17:44

Нажмите на кнопку **[Создать <тип ресурса>]**. В открывшемся окне виден общий код для этого ресурса. Также справа находится поле **Схема**, которое содержит общую информацию о свойствах объекта, эти данные помогут вам составить собственный манифест.

Создать Logging

Создайте, введяруческое определение YAML или JSON, или перетащите файл в редактор.

```
1 apiVersion: logging.banzaicloud.io/v1beta1
2 kind: Logging
3 metadata:
4   name: example
5   spec: {}
```

Logging

Схема

- `apiVersion`
 - String
 - 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/api-architecture/api-conventions.md#resources>
- `kind`
 - String
 - 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/api-architecture/api-conventions.md#types-kinds>
- `metadata`
 - Object
 - Standard object's metadata. More info: <https://git.k8s.io/community/contributors/devel/api-architecture/api-conventions.md#metadata>

Просмотреть детали

Создать Отмена Документ