

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 установятся 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:

YAML |

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



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:

YAML |

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

```
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
```

YAML | 



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 создается автоматически.

4.3. Установка Flow для FluentD

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

YAML | 

```

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

```

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

4.4. Проверка

Убедитесь, что логи появились в OpenSearch.

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

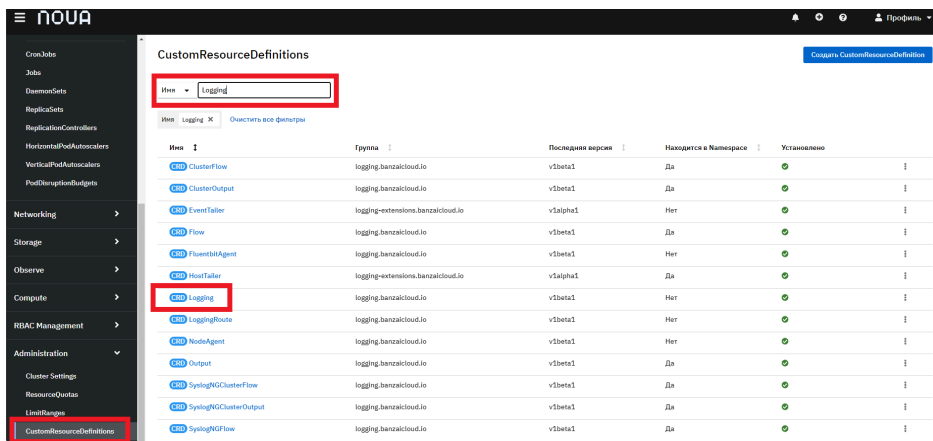
В Nova Container Platform *CRDs* представлены в веб-консоли на вкладке **Administration** > **CustomResourceDefinitions**.

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*.

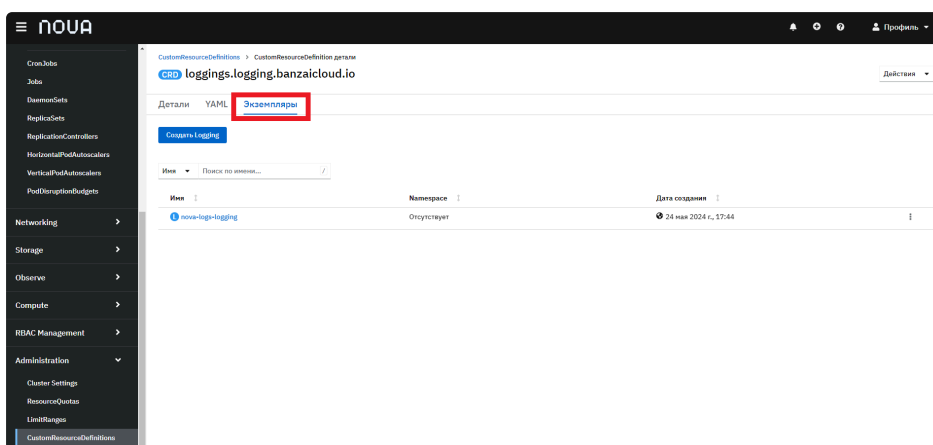
```
apiVersion: logging.banzaicloud.io/v1beta1
kind: Logging
```

YAML | 

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



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.



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.

