|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА − Российский технологический университет»**  **РТУ МИРЭА** |

**Институт информационных технологий (ИИТ)**

**Кафедра прикладной математики (ПМ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4**

по дисциплине «Методы анализа данных»

|  |  |  |
| --- | --- | --- |
| Студент группы: | ИМБО-02-21 Новичков С.Д. |  |
| Ассистент кафедры ПМ: | Горячев А.А. |  |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_2023\_г. |  | |

Москва 2023 г.

Работа с Apache NiFi

# Создание вашего первого потока данных

NiFi поддерживает аутентификацию пользователей через клиентские сертификаты, через имя пользователя и пароль, через Apache Knox или через OpenID Connect. По умолчанию считается, что единственный пользователь авторизуется через имя пользователя и пароль. Для аутентификации и синхронизации с пользователем Linux необходимо установить один из нескольких сложных механизмов аутентификации. Их установка выходит за рамки лабораторных работ и выходит за рамки ресурсов, доступных на виртуальной машине.

Откроем терминал и выполним следующую команду для запуска NiFi сервиса:



Рисунок 1 – Запуск сервиса NiFi

Откроем браузер Firefox и откроем веб-интерфейс NiFi по пути https://localhost:8443/nifi. Авторизуемся в сервисе NiFi. В качестве имени пользователя и пароля вводим student123456789.

Посмотрим на интерфейс NiFi и на всплывающее окно при добавлении обработчика. Нажмем на hadoop в списке категорий на левой вкладке. Обратим внимание, как меняется выбор доступных обработчиков в середине. С обработчиком Hadoop связано 16 обработчиков.

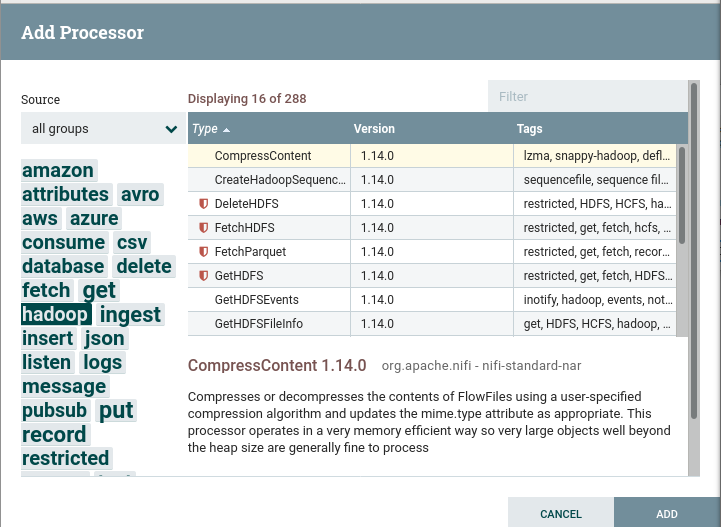


Рисунок 2 – Интерфейс выбора обработчика, связанного с hadoop

Попробуем скомбинировать Hadoop с другими категориями (например, put или get). Выберем сначала put и get одновременно. Заметим, что подходящих обработчиков, относящихся к Hadoop и которые выполяняют операции put и get, нет.

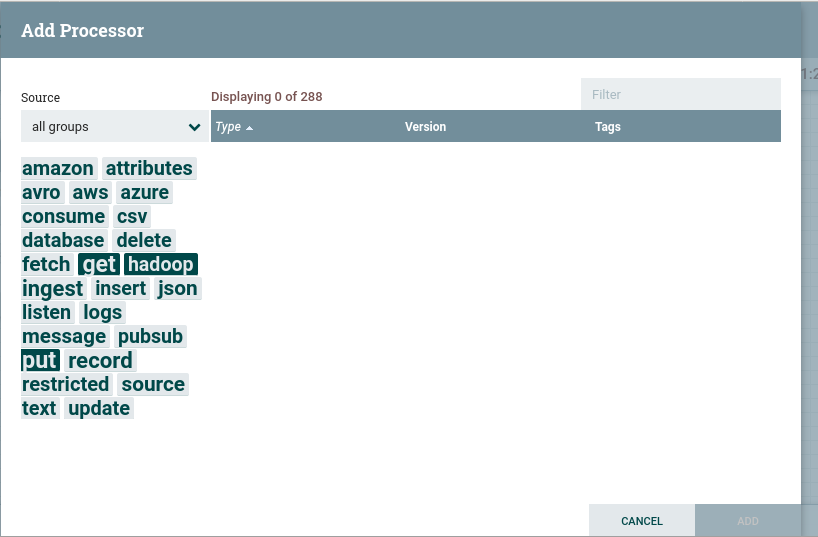


Рисунок 3 – Выбор категорий hadoop, put и get

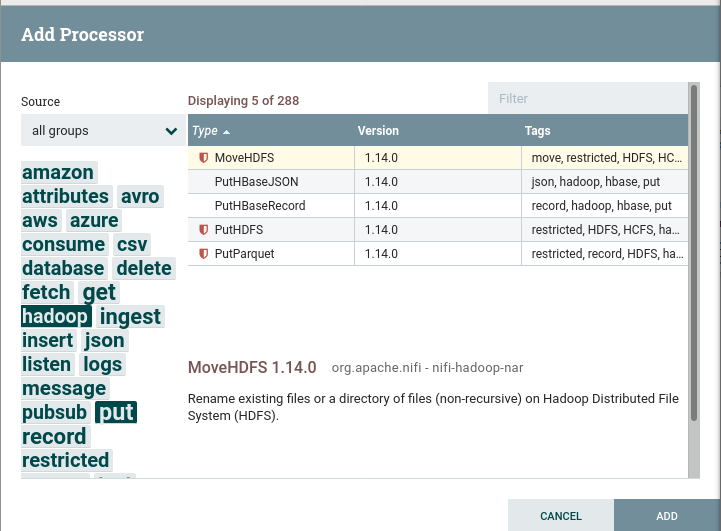


Рисунок 4 – Выбор категорий hadoop и put

Выбрав hadoop и put на левой панели введем "Gener" в фильтре в левом верхнем углу всплывающего окна. Доступных вариантов выбора снова нет, т.к. фильтр и категории работают вместе. Нет обработчика, в названии которого есть "Gener", который является частью как категории Hadoop, так и категории put.

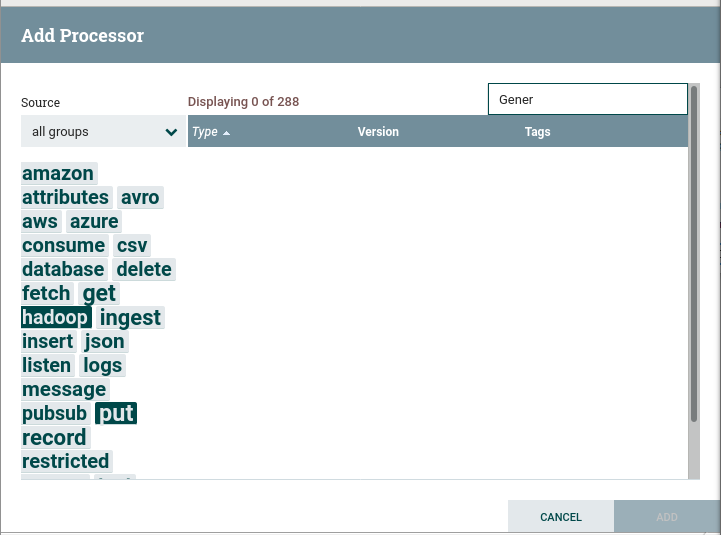


Рисунок 5 – Выбор категорий hadoop и put с применением фильтра Gener

Теперь удалим Hadoop и put из фильтров категорий и введем "Gener" в фильтре поиска. Выберем обработчик GenerateFlowFile, нажав на него, и затем нажмем Add.

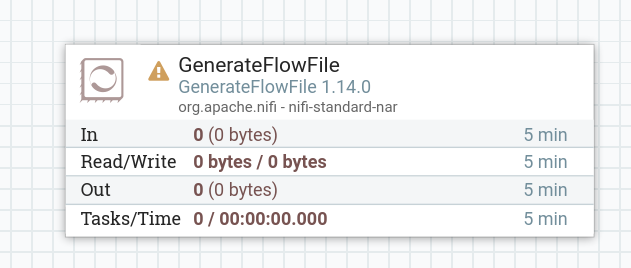


Рисунок 6 – Добавление обработчика GenerateFlowFile

Откроем всплывающее окно настройки обработчика, дважды щелкнув по обработчику GenerateFlowFile, либо щелкнув правой кнопкой мыши и выбрав configure. Настроим обработчик.

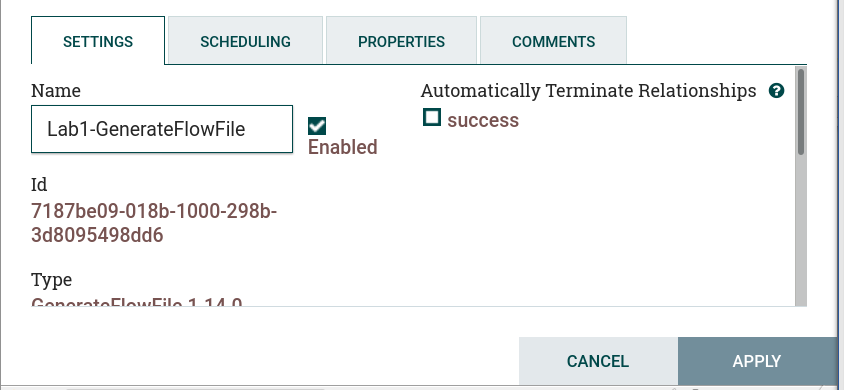


Рисунок 7 – Именование обработчика

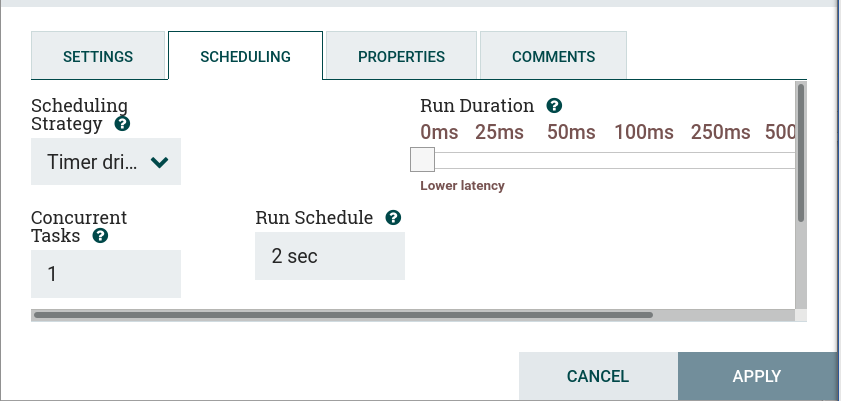


Рисунок 8 – Настраивание времени срабатывания Run Sсhedule

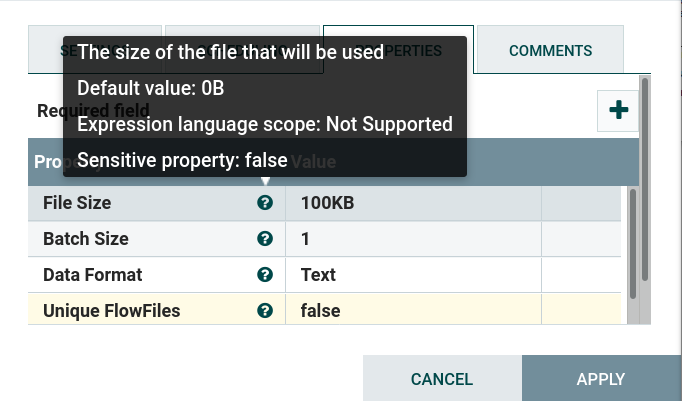


Рисунок 9 – Настройка размера файла

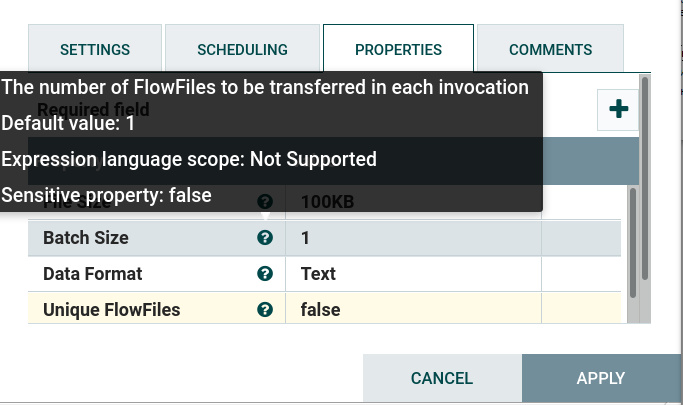


Рисунок 10 – Настройка размера пакета

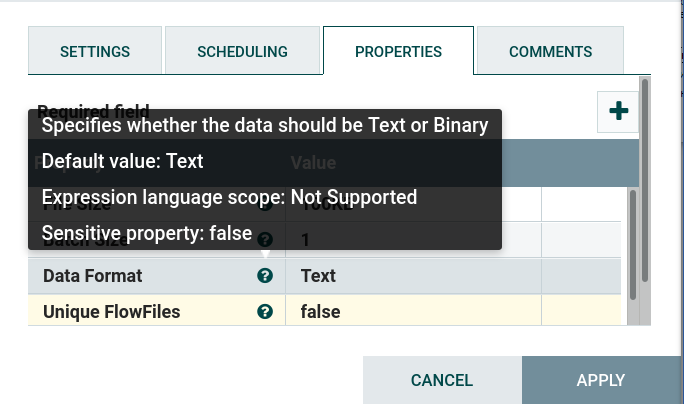


Рисунок 11 – Настройка формата данных

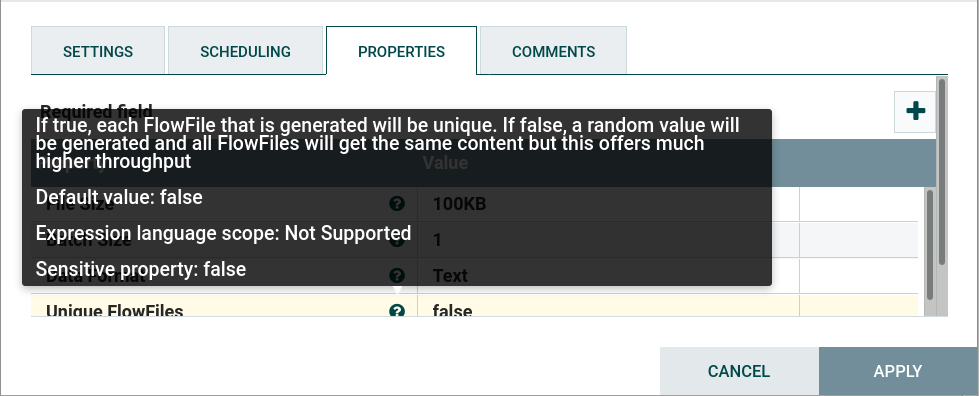


Рисунок 12 – Настройка уникальности файлов

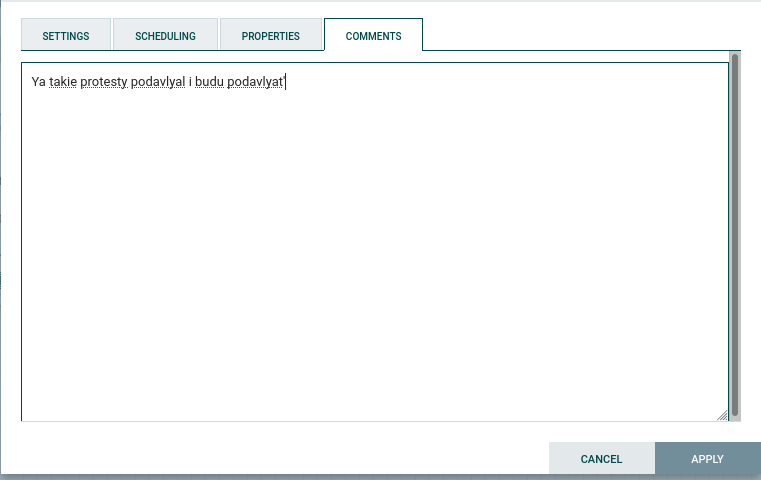


Рисунок 13 – Настройка комментария

По завершению нажимаем на APPLY для сохранения изменений.

Рассмотрим работу с обработчиком. Выберем обработчик Lab1-GenerateFlowFile нажатием на него. Сделаем копию обработчика с помощью горячих клавиш.

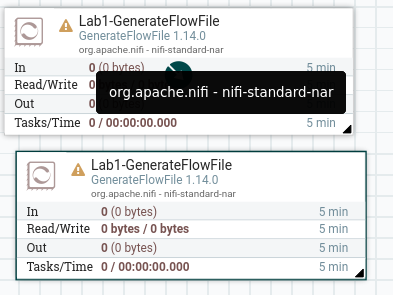


Рисунок 14 – Создание копии обработчика

Выберите оба обработчика щелчком по ним с зажатой клавишей Shift.

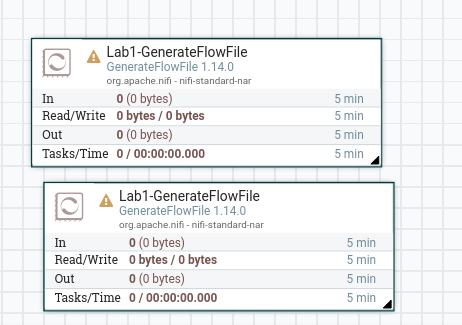


Рисунок 15 – Выделение двух обработчиков

Отключим второй обработчик Lab1-GenerateFlowFile. Это можно выполнить нажатием кнопки Disable из панели операций или из контекстного меню.

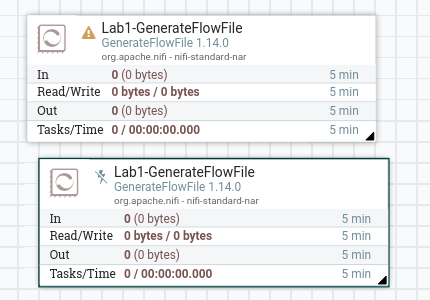


Рисунок 16 – Отключение одного из обработчиков

Теперь удалим отключенный обработчик, который мы только что создали с помощью кнопки Delete из контекстного меню.

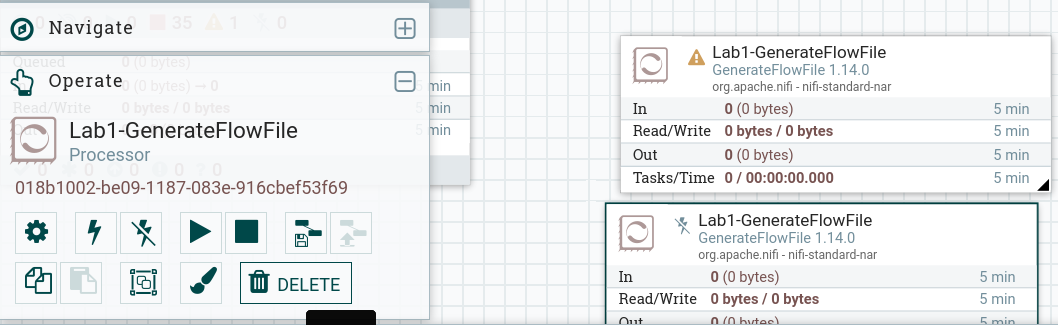


Рисунок 17 – Удаление копии обработчика

Добавим и настроим на холст обработчик PutFile. Откроем главное меню (Global menu) в левом верхнем углу и выберем пункт справки (Help).

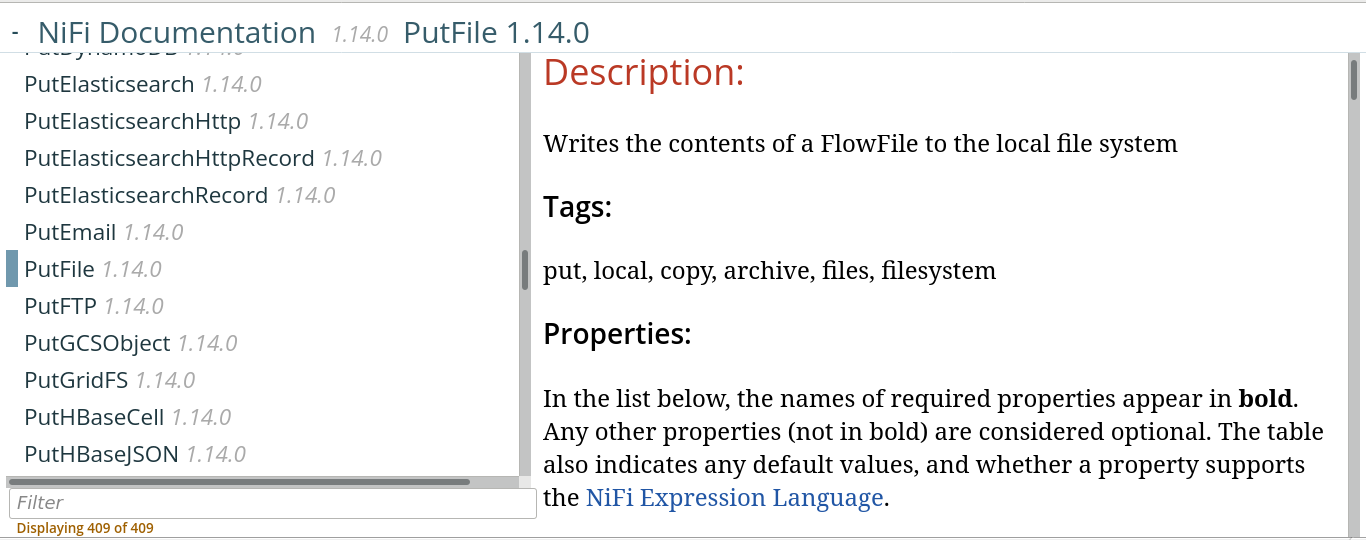


Рисунок 18 – Справка по обработчику PutFile

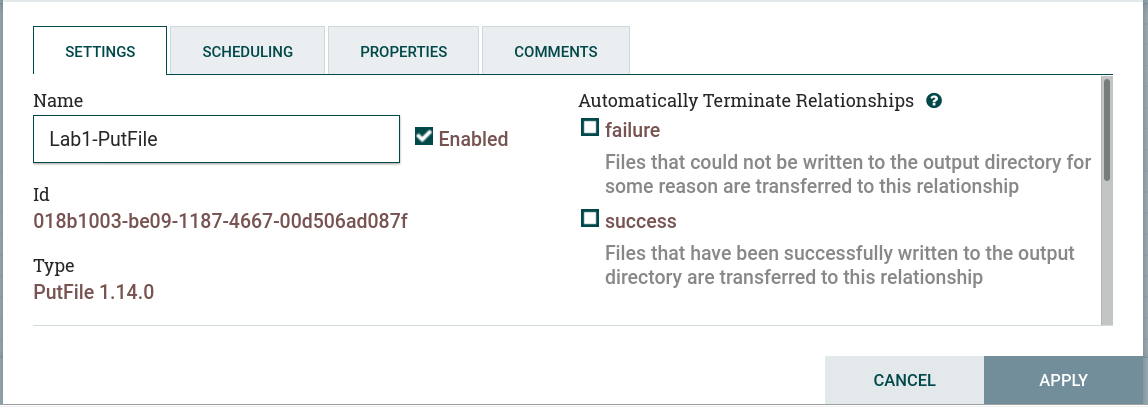


Рисунок 19 – Именование обработчика

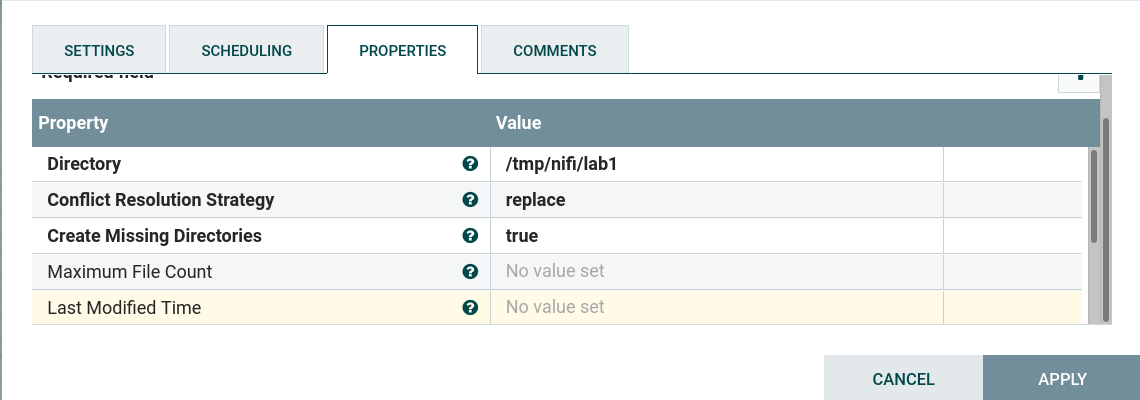


Рисунок 20 – Настройка свойств обработчика

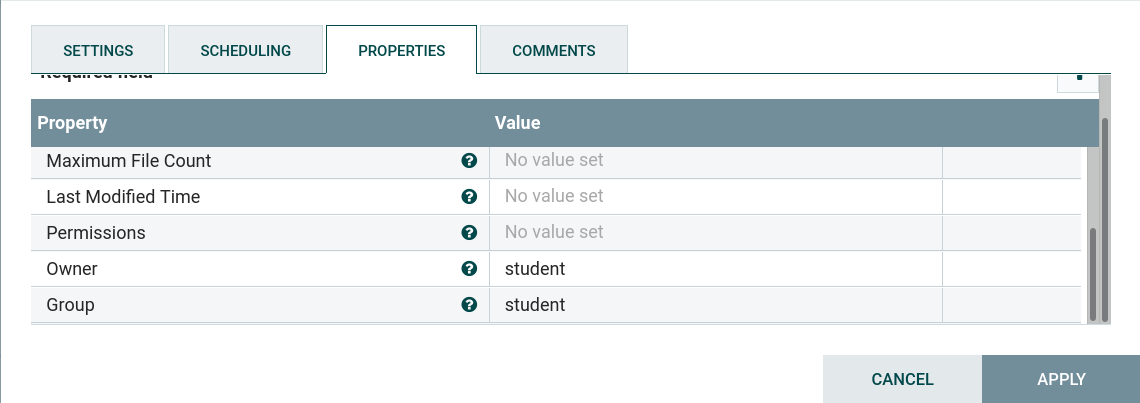


Рисунок 21 – Настройка свойств обработчика

Создадим связь между обработчиками Lab1-GenerateFlowFile и Lab1-PutFile нажатием и перетаскиванием иконки из центра обработчика Lab1-GenerateFlowFile в центр обработчика Lab1-PutFile. Выберем пункт success в пункте For Relationship и добавим (Add)

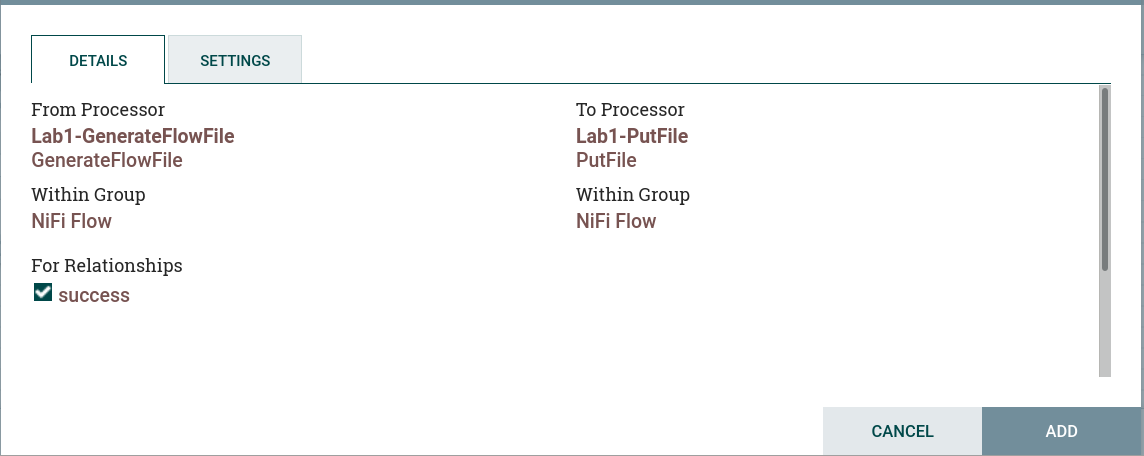


Рисунок 22 – Связывание обработчиков

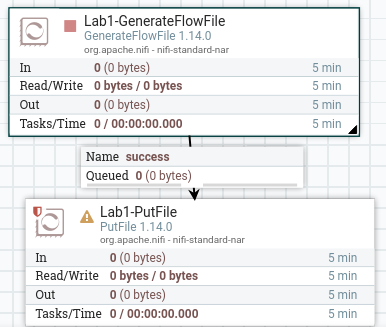


Рисунок 23 – Связанные обработчики

Желтый треугольник с восклицательным знаком в обработчике Lab1-PutFile показывает, что в настройках обработчика есть проблемы. Это происходит, потому что все отношения должны быть либо перенаправлены другому обработчику, либо завершены.

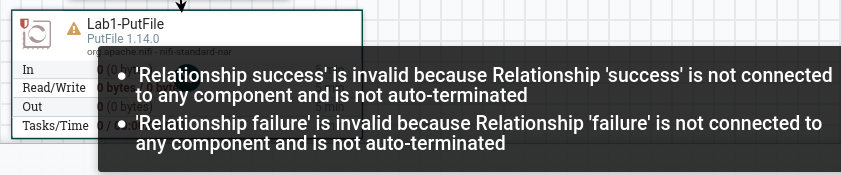


Рисунок 24 – Незавершенные отношения

Есть пара отношений, которые необходимо завершить. Поскольку после сохранения FlowFile’ов на диск нам не нужно будет пересылать их дальше по потоку, настроим их автозавершение с этого обработчика. Перейдем на вкладку SETTING в обработчике и проверим, что отношения failure и success автоматически завершаются.

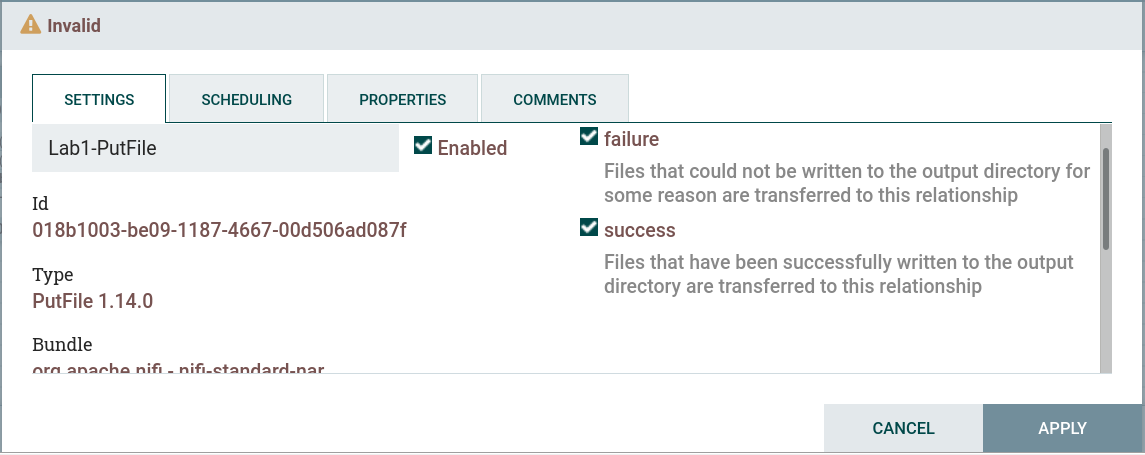


Рисунок 25 – Автозавершение отношений

Теперь оба обработчика должны находится в состоянии «остановлен» (stopped, красный квадрат), а мы сможем их запустить. Выберем оба обработчика и щелкнем по кнопке Run.

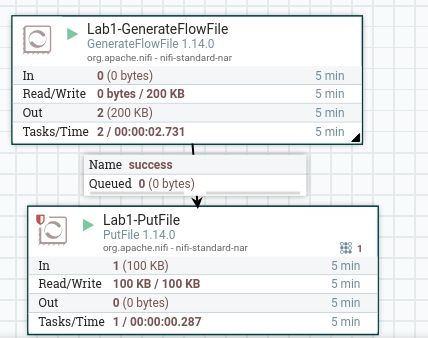


Рисунок 26 – Запуск обработчиков

Остановим оба обработчика. Процедура аналогична запуску, но на этот раз нужно выбрать иконку остановки (stop).

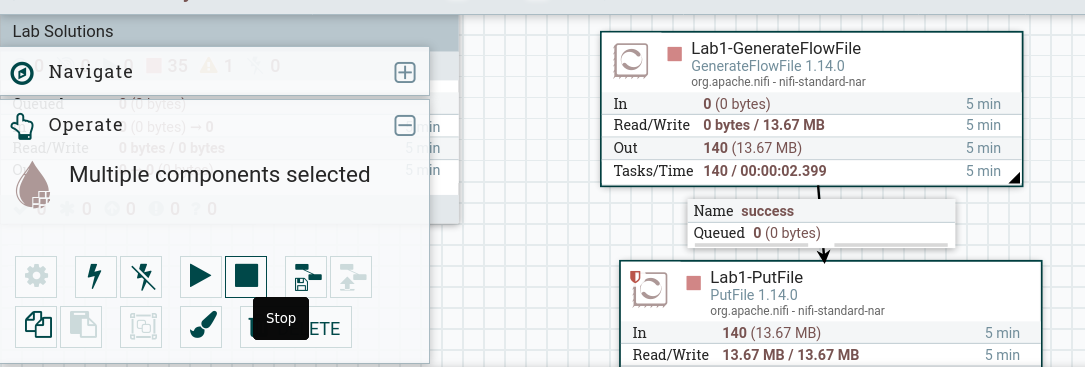


Рисунок 27 – Остановка обработчиков

Проверим конвейер данных. Откроем терминал и перейдем в локальную директорию /tmp/nifi (например, с помощью команды cd). Выполните команду ls для проверки того, что обработчик PutFile сохранил FlowFile’ы в эту директорию.

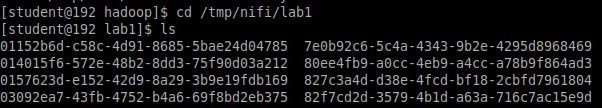


Рисунок 28 – Проверка директории

# Создание связей

В этой лабораторной работе мы рассмотрим отношения и их связи между обработчиками. Отношение в обработчике представляет собой возможный результат работы этого обработчика. Доступные связи будут определяться типом обработчика, а также атрибутами во FlowFile’е. Например, success или failure — это очень распространенные отношения, доступные в большинстве обработчиков. Напомним, что все связи должны быть либо направлены в выходной обработчик, либо быть автозавершаемыми.

Для организации потоков данных можно использовать группу обработки. Вместо размещения всех потоков данных на корневом холсте часто проще организовать проекты и управлять ими, создавая группы обработки и создавать потоки данных внутри этой группы. Группы обработки могут быть даже вложенными, создавая иерархию групп обработки. Использование групп обработки таким образом аналогично созданию каталогов и подкаталогов для организации проектов.

Добавляем новую группу обработки в корневой холст перетаскиванием иконки группы обработки на холст. Назовем группу Lab 2.

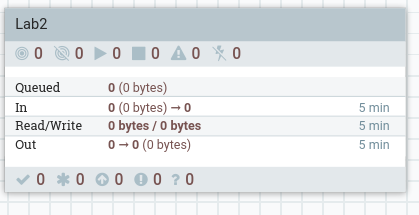


Рисунок 29 – Создание группы Lab2

Дважды щелкнем по вновь созданной группе обработки. Нам будет предоставлен чистый холст, на котором мы сможем создавать потоки данных. Nifi также сообщит нам, где мы находимся в иерархии групп обработки. Мы можем перемещаться между группами обработки, нажав на нужную.

Добавьте на холст обработчик GenerateFlowFile и настроим его.

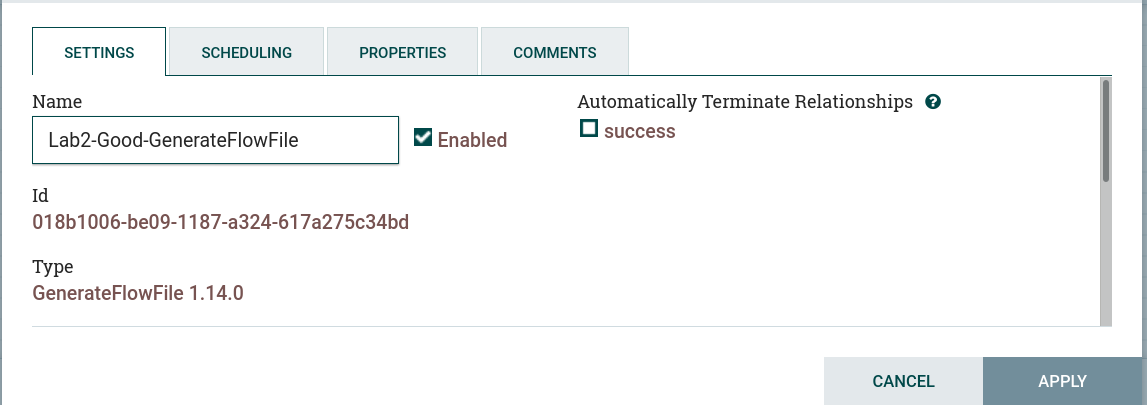


Рисунок 30 – Именование обработчика

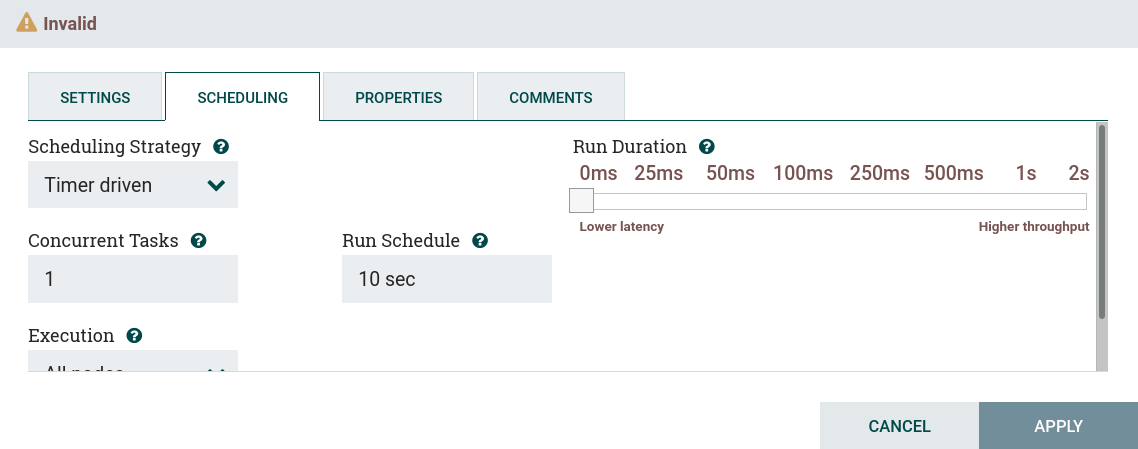


Рисунок 31 – Настройка Run Schedule

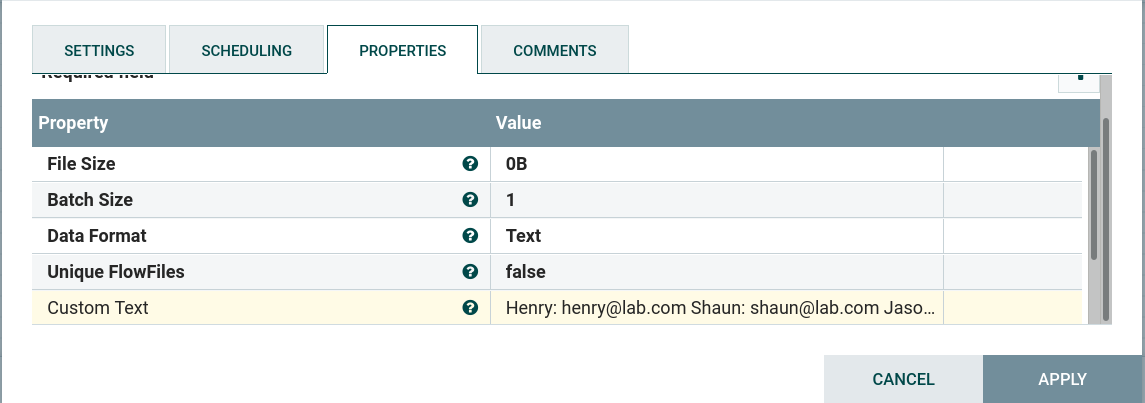


Рисунок 32 – Настройка свойств

Добавим на холст второй обработчик GenerateFlowFile и настроим его.

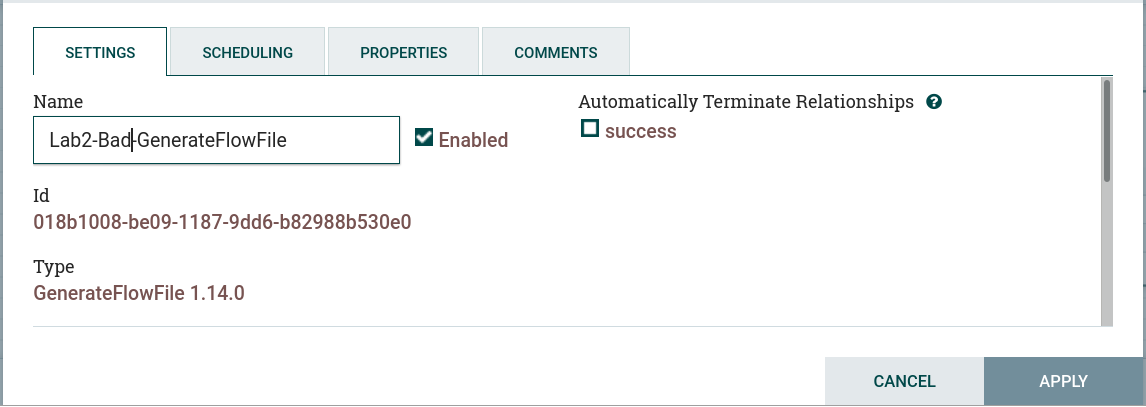


Рисунок 33 – Именование объекта

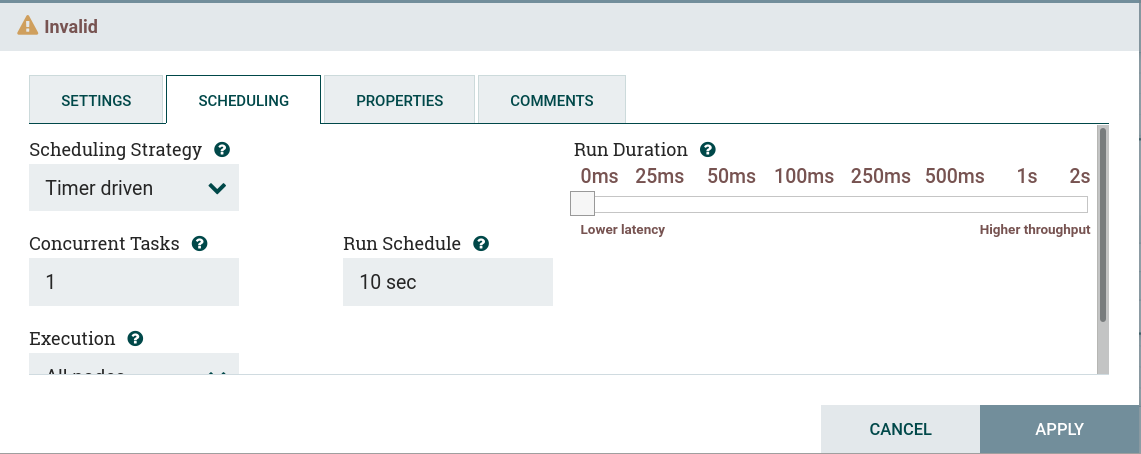


Рисунок 34 – Настройка Run Schedule

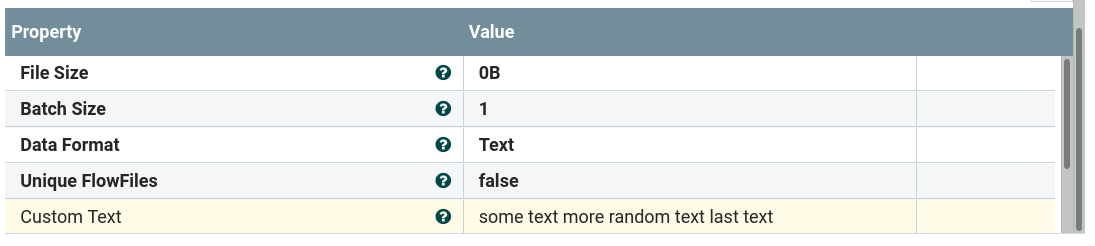


Рисунок 35 – Настройка свойств обработчика

Добавьте на холст обработчик SplitText и назовем его Lab2-SplitText. Настроим отношения failure и original, как автозавершаемые. После того, как обработчик SplitText попытается разделить входящий текст, существует три возможных отношения. Если попытка разделить текст завершится неудачей, FlowFile’ы для этого результата будут перенаправлены на отношение failure. Аналогично, если разделение завершится успешно, оно будет перенаправлено на отношение splits. Наконец, этот обработчик позволяет нам также отправлять исходный текст через отношение original. Автоматически завершая отношения failure и original, мы эффективно предотвращаем дальнейшую передачу любых FlowFile’ов вниз по потоку для этих двух ситуаций.

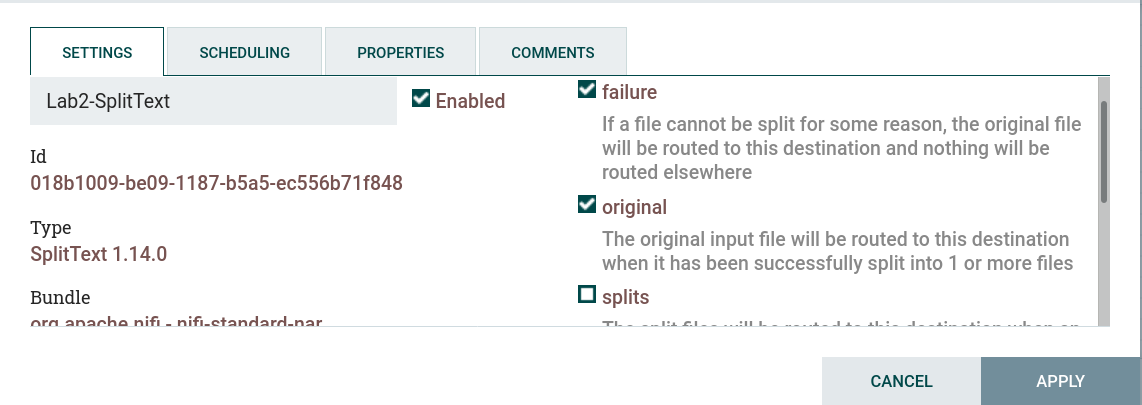


Рисунок 36 – Именование и настройка отношений обработчика



Рисунок 37 – Настройка Run Schedule обработчика

Установим Run Schedule на 0 секунд. Это позволит обработчику обрабатывать любые FlowFile’ы по мере того, как они становятся доступны для обработки.

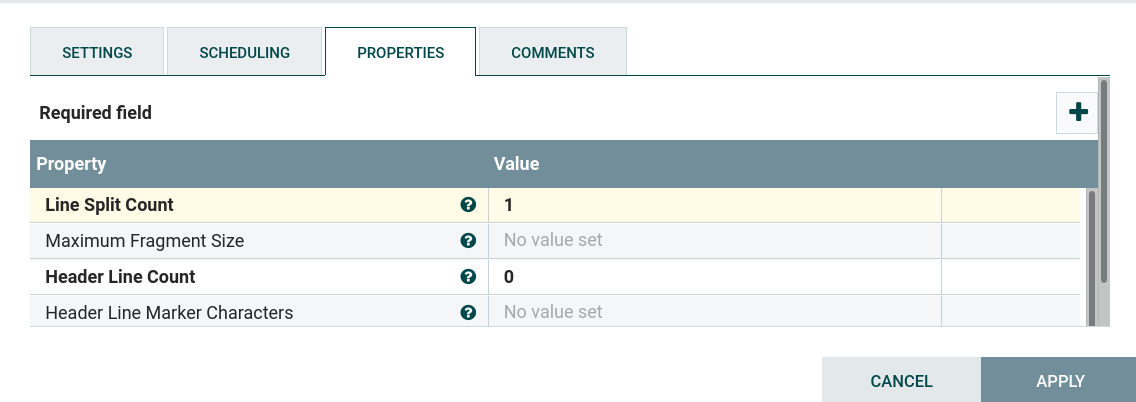


Рисунок 38 – Настройка свойств обработчика

Установим Line Split Count (кол-во разбиений строк) значение 1. Это позволит разбивать каждую строку во входящем FlowFile’е на отдельный FlowFile. Поскольку на вход приходят FlowFile’ы от обработчика GenerateFlowFile, который отправляет текст, состоящий из нескольких строк, этот обработчик будет генерировать несколько FlowFile’ов из одного.

Соединим обработчик GenerateFlowFile с обработчиком SplitText в отношении success

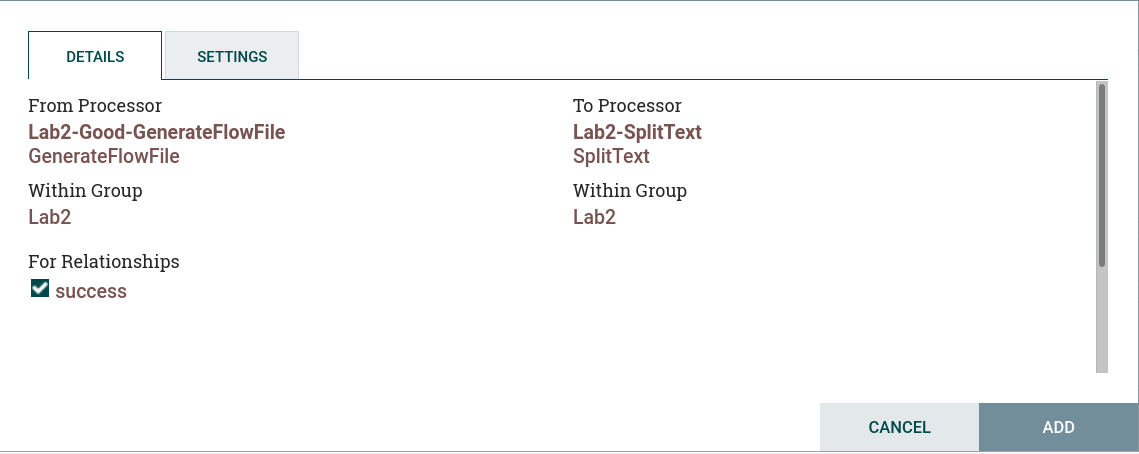


Рисунок 39 – Соединение обработчиков

Таким же образом соединим Lab2-Bad-GenerateFlowFile и Lab2-SplitText.

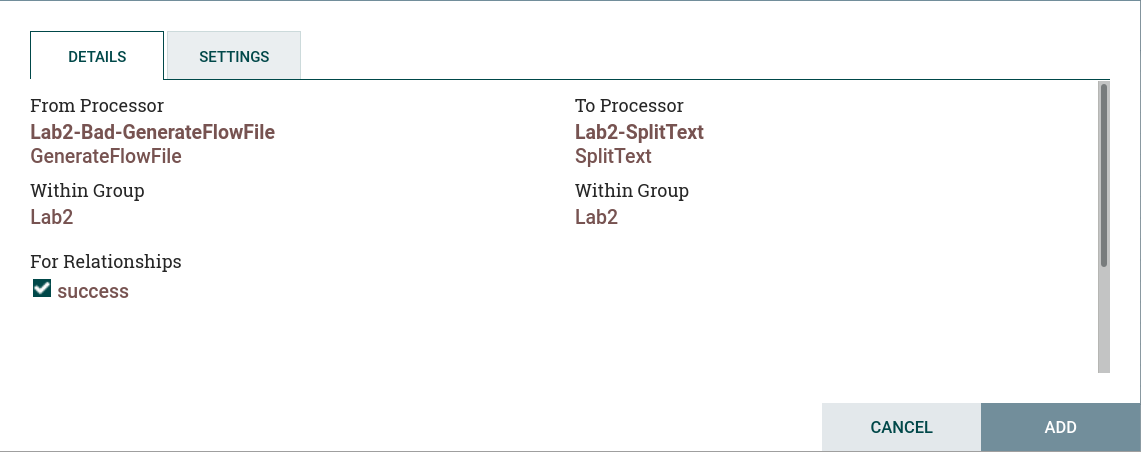


Рисунок 40 – Соединение обработчиков

Теперь наш холст выглядит так:

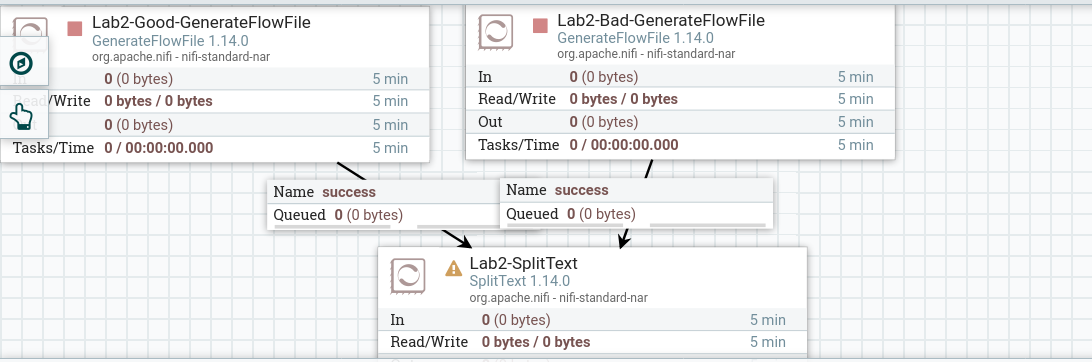


Рисунок 41 – Холст после настройки обработчиков

Добавьте на холст обработчик ExtractText и настроим его.

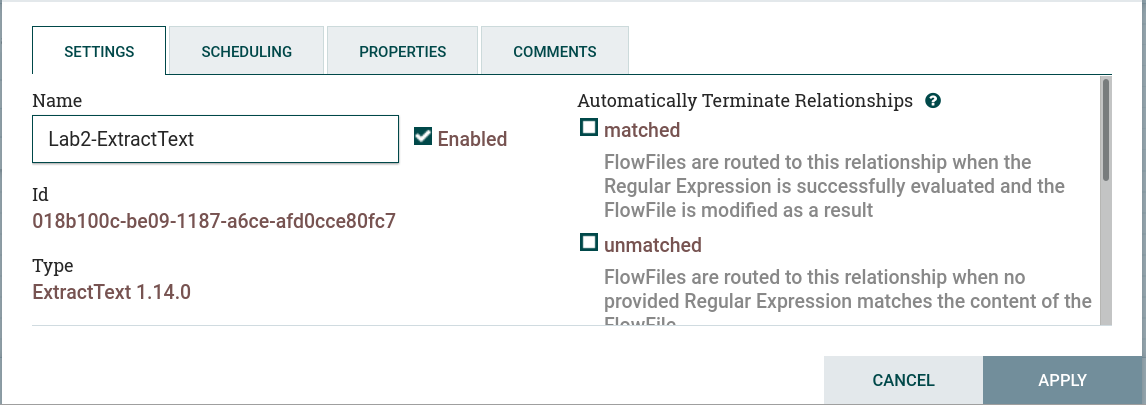


Рисунок 42 – Именование обработчика

На вкладке Properties мы хотим добавить пользовательский атрибут. Для этого нажмем плюс () сверху справа. Во всплывающем указываем имя атрибута - names.

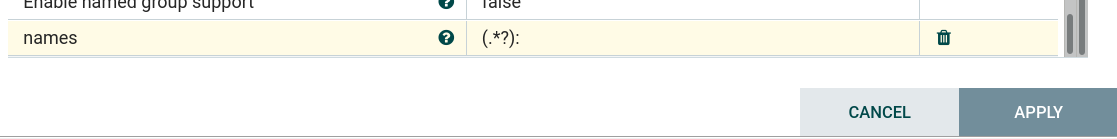


Рисунок 43 – Добавление пользовательского атрибута

Это регулярное выражение, которому удовлетворяют 0 или более любых символов до знака двоеточие (:). Круглые скобки захватывают блок. Значение, охваченное скобками, будет присвоено свойству names.

Соединим обработчик Lab2-SplitText с обработчиком Lab2-ExtractText в отношении splits.

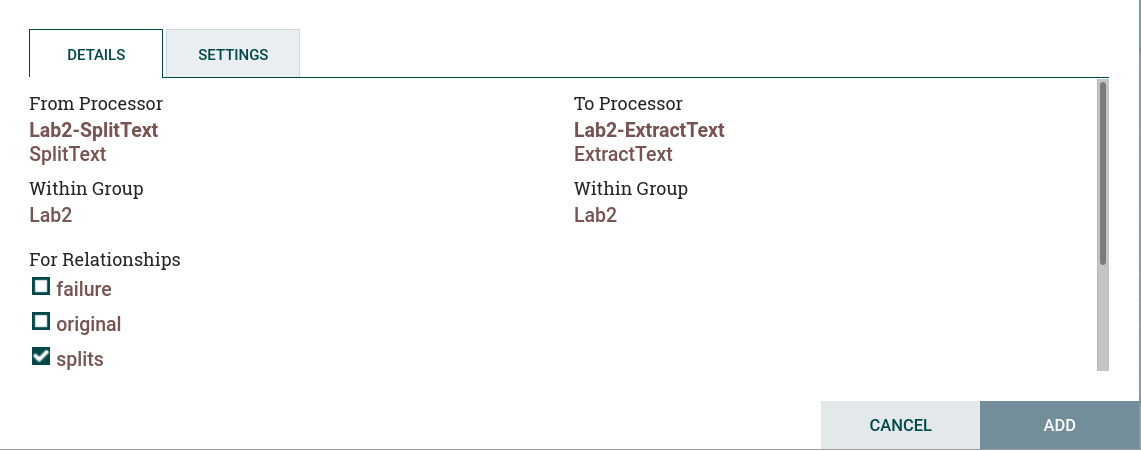


Рисунок 44 – Соединение обработчиков

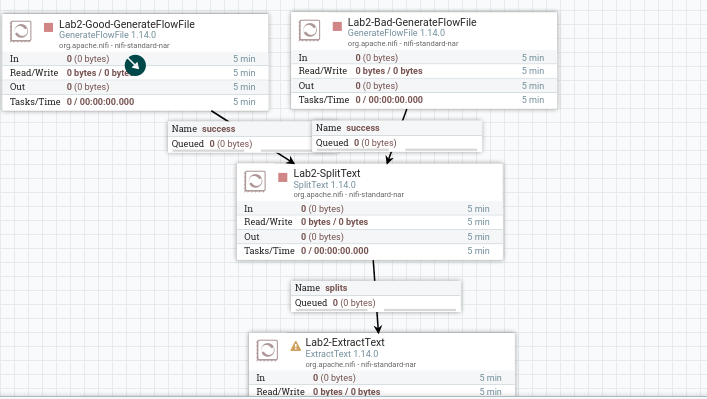


Рисунок 45 – Холст после настройки связей между обработчиками

Обратим внимание на желтый восклицательный знак на обработчике Lab2-ExtractText. Проблема заключается в том, что есть два отношения, сгенерированные из Lab2-ExtractText, которые не завершается. Мы должны либо автоматически завершать их, либо передать FlowFile’ы для каждого отношения дальше по потоку.

Создадим временный пункт назначения исходящего потока с помощью воронки (Funnel). Подключим отношение matched к одной из воронок, а отношение unmatched - к другой.



Рисунок 46 – Соединение обработчика с воронкой



Рисунок 47 – Соединение обработчика с воронкой

Воронка позволяет принимать связи из нескольких источников и пересылать их нижестоящему обработчику. Однако многие разработчики NiFi используют воронку в качестве временного места назначения для потоков данных, которые все еще находятся в разработке. В этом контексте мы и будем использовать воронку. Когда мы закончим, наш конвейер потока данных должен выглядеть так, как показано ниже.

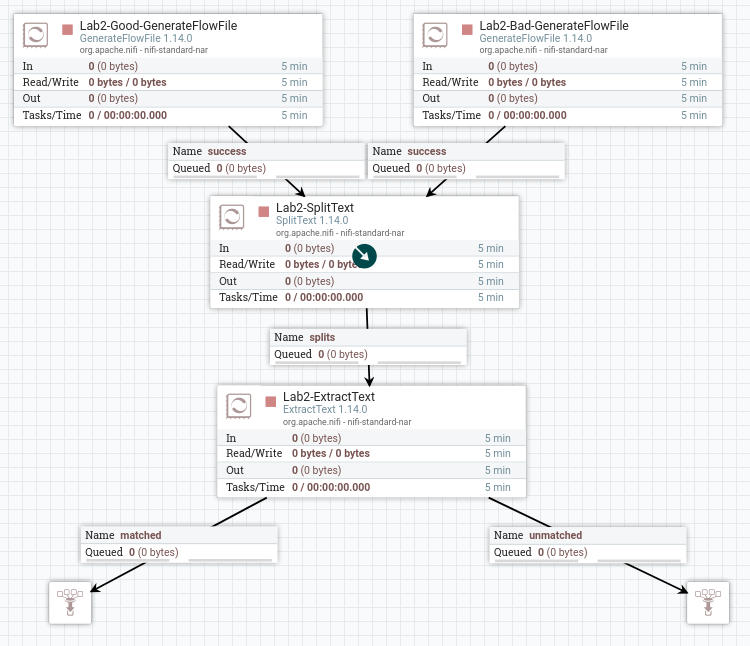


Рисунок 48 – Итоговый холст

# Навигация в потоках данных

В этой лабораторной работе мы будем запускать обработчики, созданные в лабораторной работе 2, и наблюдать за FlowFile’ами по мере их создания. Наконец, мы возьмем часть потока данных и создадим шаблон для повторного использования в следующих работах.

Создадим новую группу обработки на корневом холсте и назовем ее Lab3. Перейдем к потоку, созданному во второй работе, и выберем все его компоненты, включая обработчики, связи и воронки и скопируем.

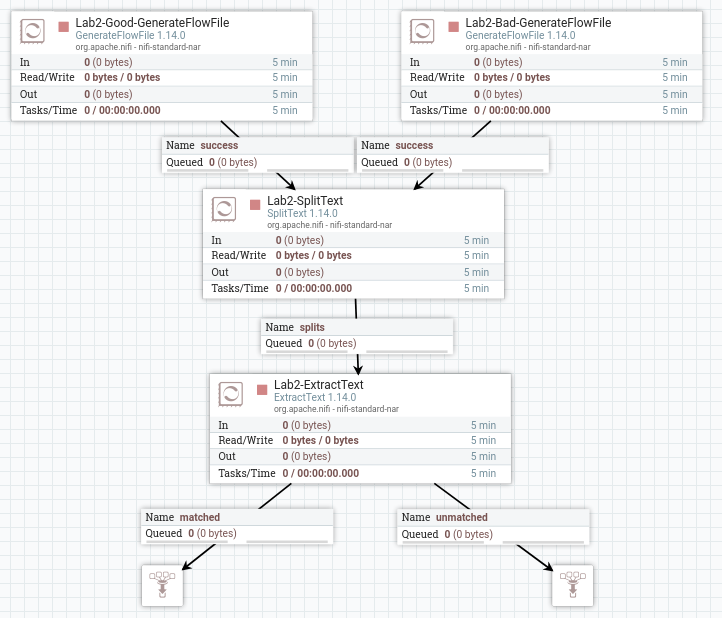


Рисунок 49 – Вставка потока из работы 2

Запустим обработчик Lab2-Good-GenerateFlowFile. Выберем Lab2-Good-GenerateFlowFile. Нажмем на иконку запуска () с панели операций

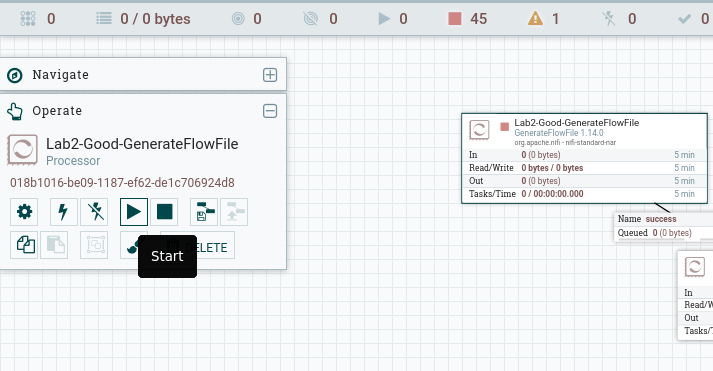


Рисунок 50 – Запуск обработчика

Дадим обработчику Lab2-Good-GenerateFlowFile поработать около 30 секунд. Вспомним, что мы настроили период срабатывания обработчика на уровне 10 секунд, поэтому за время работы сгенерируется 3 FlowFile’а. Остановим обработчик, нажав на значок остановки () на панели операций.

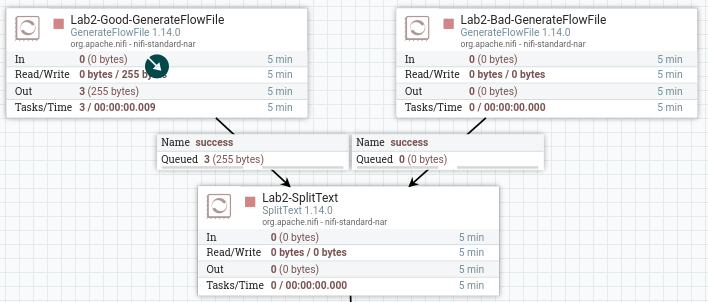


Рисунок 51 – Результат работы 30 секунд обработчика

Вызовем контекстное меню щелчком правой кнопкой мыши по связи, где набрались в очередь FlowFile’ы. Выберем List queue.

Появится окно, где будут отображены FlowFile’ы.

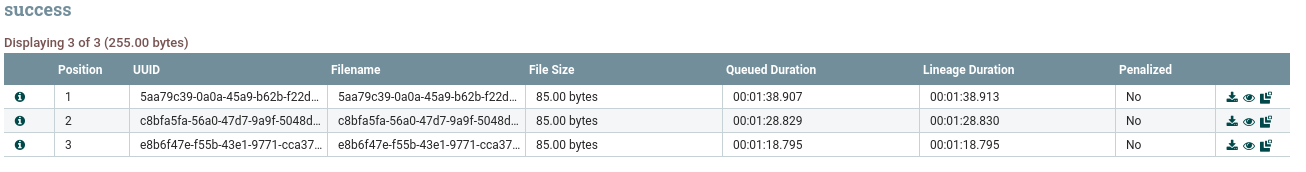


Рисунок 52 – Очередь FlowFile

Мы можем получить подробную информацию о FlowFile’ах, включая информацию об атрибутах. Нажмем на иконку информации () в столбце слева, появится всплывающее окно:

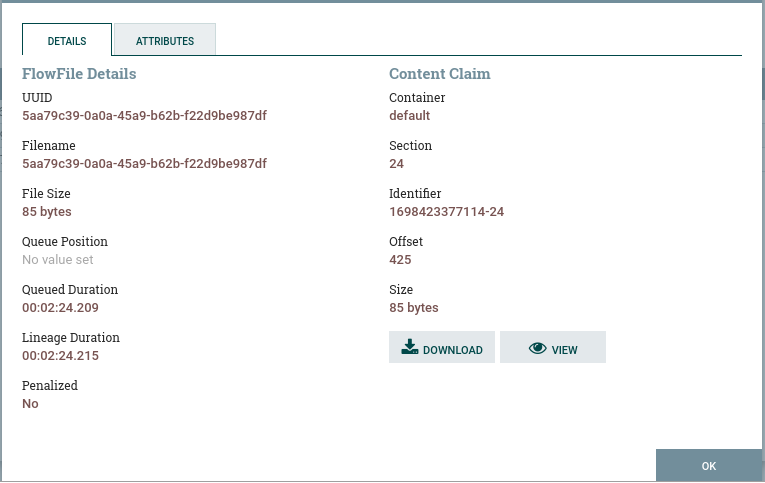


Рисунок 53 – Сведения о FlowFile

С каждым FlowFile’ом связан уникальный UUID. Он позволяет NiFi отслеживать каждый FlowFile и записывать его происхождение. Происхождение — это информация о том, как и когда FlowFile был создан, изменен, скопирован, клонирован, удален и т.д. В нем отслеживается каждое изменение во времени. Это одна из особенностей NiFi. Мы также видим, что FlowFile’ы имеют физическое воплощение на диске, то есть это физические файлы с именем, которые сохраняются в течение некоторого заданного периода времени. При этом, эта часть FlowFile’а не содержит фактического содержимого, поскольку это было бы очень неэффективно. Сам FlowFile содержит указатель на другой файл - файл содержимого, в котором хранится фактическое содержимое FlowFile’а. Мы можем просмотреть или загрузить содержимое, используя значок загрузки ( ) или просмотра ( ) на вкладке сведений (details).

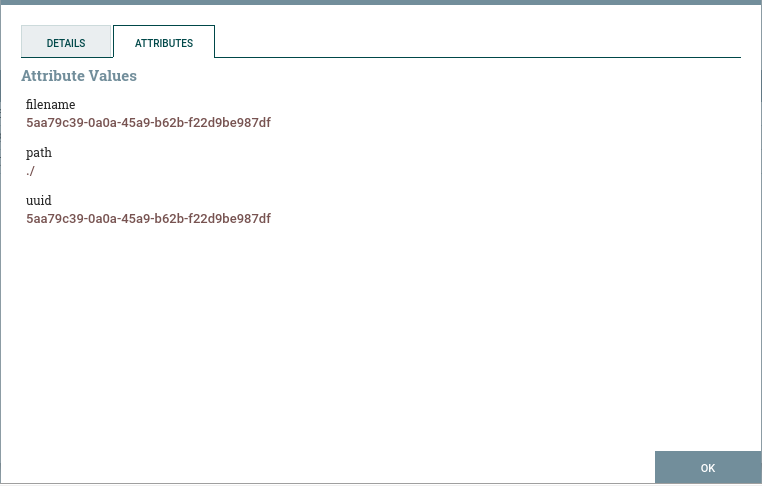


Рисунок 54 – Атрибуты FlowFile

На вкладке Attributes мы можем просмотреть все атрибуты, связанные с этим FlowFile’ом. Каждый FlowFile будет иметь несколько атрибутов по умолчанию, таких как имя файла, uuid, путь и т.д. Кроме того, если пользователь создал какие-либо пользовательские атрибуты, они также будут отображаться здесь.

Посмотрим содержимое FlowFile. Мы должны увидеть текст, которые мы вводили в поле для ввода произвольного текста в обработчике Lab2-Good-GenerateFlowFile. А именно:

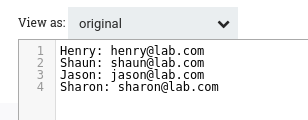


Рисунок 55 - Содержимое FlowFile

Запустим только обработчик Lab2-Bad-GenerateFlowFile примерно на 30 секунд, а затем остановим.

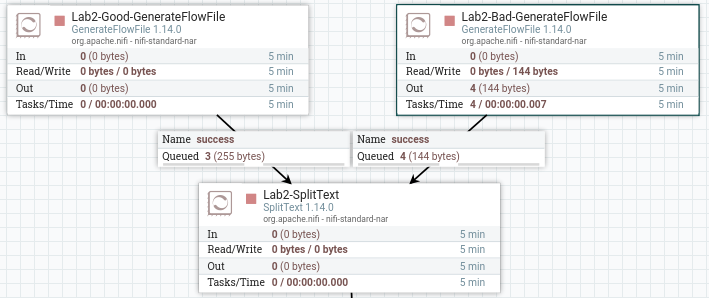


Рисунок 56 – Запуск обработчика Lab2-Bad-GenerateFlowFile

Просмотрим содержимое FlowFile’ов в очереди из этого отношения.

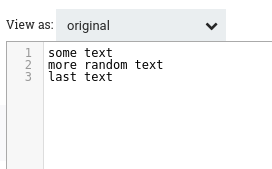


Рисунок 57 - Содержимое FlowFile’ов

Мы также увидим введенный текст. Мы вводили «some text, more random text, last text» на отдельных строках.

Запустим только текстовый обработчик Lab2-Split и дадим ему поработать, пока не увидим, что все FlowFile’ы в очереди из входящего потока удалены. Каждый FlowFile из Lab2-Good-GenerateFlowFile будет разделен на отдельные FlowFile’ы для каждой строки. Поскольку существует четыре строки, это приведет к появлению четырех новых FlowFile’ов. Аналогично, в каждом FlowFile’е из Lab2-Bad-GenerateFlowFile есть три строки. Это приведет к появлению трех новых FlowFile’ов.

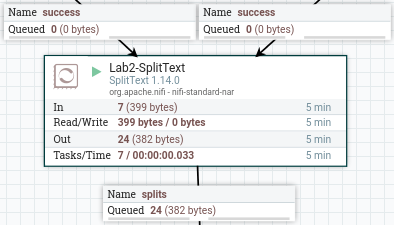


Рисунок 58 – Работа обработчика Lab2-Split

Посмотрим информацию о FlowFile’ах, в том числе атрибуты и содержимое.

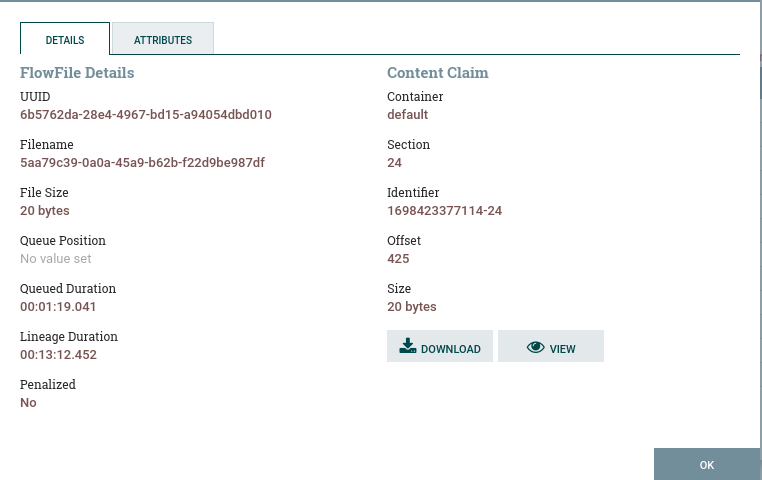


Рисунок 59 – Сведения о FlowFile’е в очереди

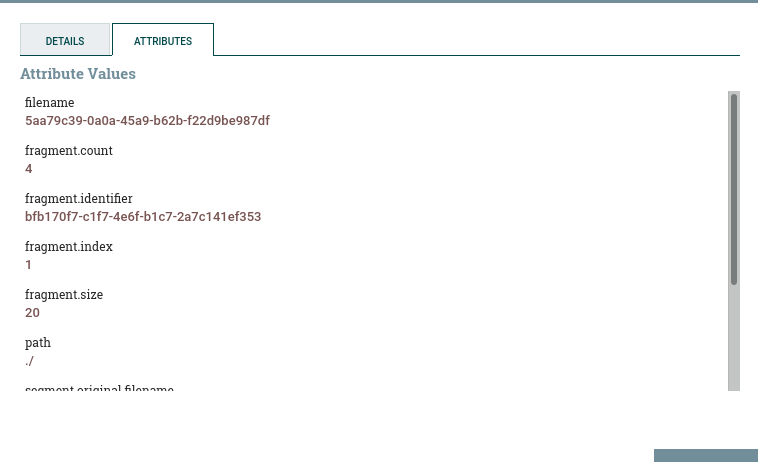


Рисунок 60 – Атрибуты FlowFile в очереди



Рисунок 61 – Атрибуты FlowFile в очереди



Рисунок 62 – Содержимое FlowFile’а

Мы видим некоторые новые атрибуты, автоматически созданные системой в результате операции разделения. Теперь существует атрибут fragment.count, который соответствует количеству новых FlowFile’ов, созданных из исходного FlowFile’а из входного потока. Новый атрибут fragment.index соответствует индексному номеру текущего FlowFile’а внутри фрагментов. FlowFile, показанный ниже, является первым фрагментом. Когда мы проверяем содержимое, мы видим, что оно соответствует первому элементу строки, который мы ввели в пользовательское текстовое поле для Lab2-Good-GenerateFlowFile.

Запустим обработчик Lab2-ExtractText и дадим ему поработать, пока вся очередь FlowFile’ов не обработается и не передастся дальше по потоку к одной из двух воронок.

Откроем FlowFile’ы в очереди matched. Похожим образом посмотрим на FlowFile’ы из очереди unmatched.

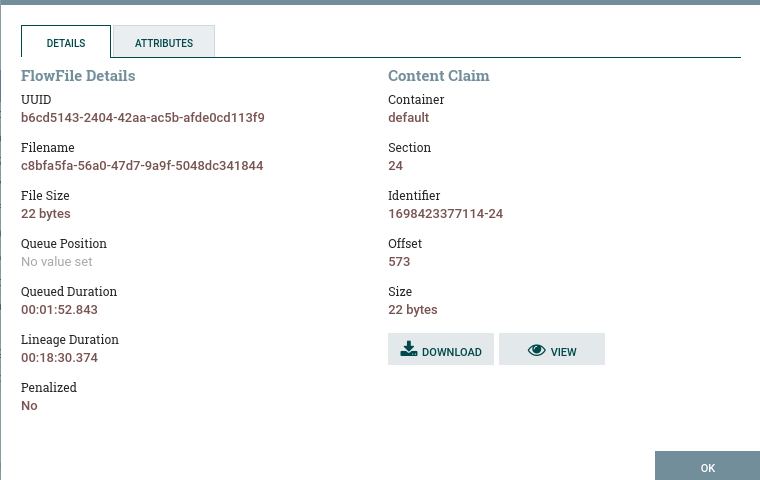


Рисунок 63 - Сведения о FlowFile’е в очереди matched

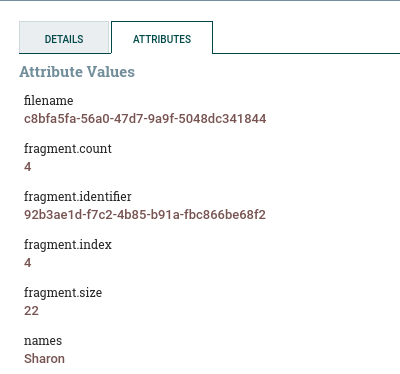


Рисунок 64 – Атрибуты FlowFile в очереди matched

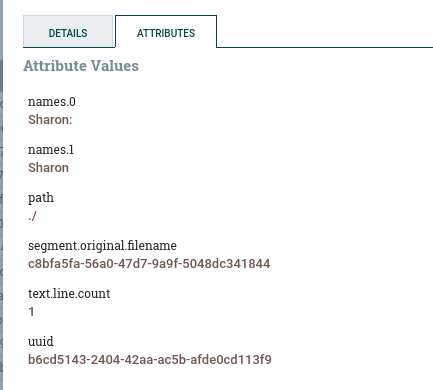


Рисунок 65 – Атрибуты FlowFile в очереди matched

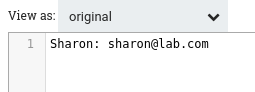


Рисунок 66 – Содержимое FlowFile в очереди matched

Мы видим, что все FlowFile’ы, созданные из Lab2-Good-GenerateFlowFile, оказались в очереди matched, в то время как все FlowFile’ы из Lab2-Bad-GenerateFlowFile находятся в очереди unmatched. Это связано с регулярным выражением, которое мы использовали в поле names текстового обработчика Lab2-Extract. Регулярное выражение соответствует всем символам до появления двоеточия. Весь текст, введенный в Lab2-Good-GenerateFlowFile, был в виде пары «ключ:значение» с двоеточием между ключом и значением. Заметим, что в unmatched атрибут names не был создан.



Рисунок 67 – Сведения о FlowFile’е в очереди unmatched

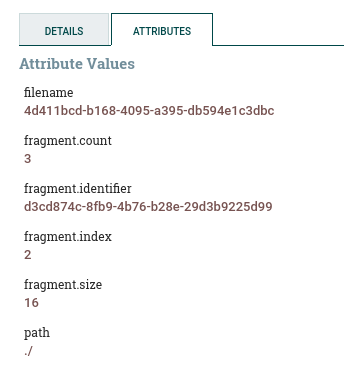
. 

Рисунок 68 – Атрибуты FlowFile в очереди unmatched

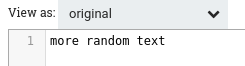


Рисунок 69 – Содержимое FlowFile в очереди unmatched

Очистим очереди, прежде чем мы закончим эту лабораторную работу. Сделаем это как для отношения matched, так и для отношения unmatched.

# Создание и использование шаблонов

В этой лабораторной работе мы узнаем, как создавать шаблоны из существующих потоков данных. Шаблоны — это способ сохранения логики работы часто используемых потоков данных. У многих разработчиков NiFi есть библиотеки шаблонов, которые они могут повторно использовать для создания более сложных потоков данных.

Добавим на корневой холст группу обработки и назовем ее Lab 4. Скопируем весь поток данных, созданный в группе Lab 3 и вставим его в группу Lab 4. Убедимся в том, что там нет запущенных обработчиков и все очереди для всех отношений очищены.

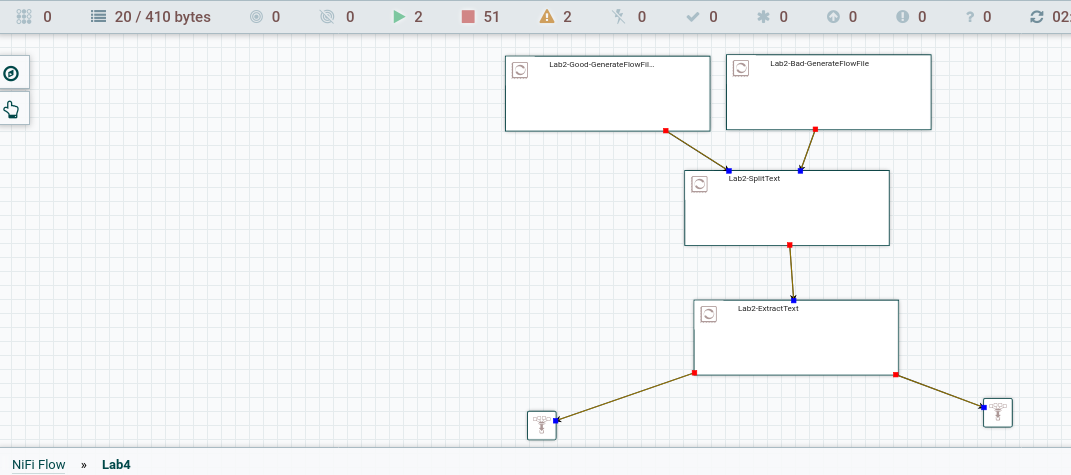


Рисунок 70 – Копирование на холст Lab 4 потока данных

Удалим обработчики, не являющиеся частью шаблона. Удалим обработчик Lab2-ExtractText. Возникнет ошибка.

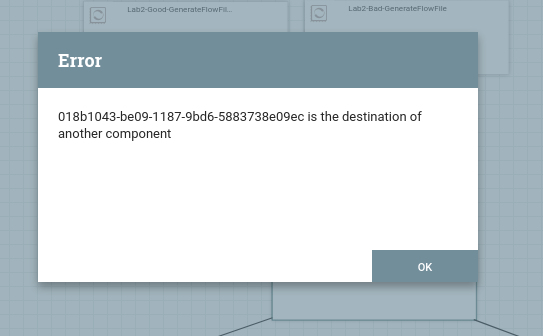


Рисунок 71 – Ошибка при удалении

NiFi жалуется на то, что обработчик является местом назначения другого компонента. NiFi пока не может удалить этот обработчик, так как это приведет к тому, что вышестоящий обработчик потеряет одно из своих назначений. Чтобы удалить обработчик, мы должны удалить все входящие в него связи. Это не относится к связям, исходящим от обработчика.

Удалим связь от Lab2-SplitText к Lab2-ExtractText. Теперь удалим обработчик Lab2-ExtractText и удалим обе воронки.

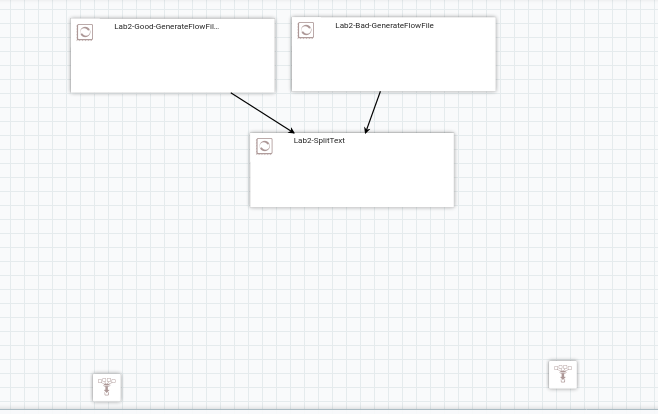


Рисунок 72 – Удаление Lab2-ExtractText

Поскольку мы собираемся сохранить поток данных в виде шаблона, мы не хотим, чтобы имена обработчиков были слишком конкретными. При этом в нашем случае все обработчики имеют префикс Lab2.

Переименуем обработчики.



Рисунок 73 – Переименование обработчиков

Изменим UUIDTextGenFlowFile. В предыдущей работе этот обработчик генерировал 3 произвольные строки текста. Мы немного улучшим этот обработчик, чтобы он генерировал случайный текст с уникальным идентификатором UUID.

Удалим предыдущий Custom Text и напишите в этом поле ${UUID()}

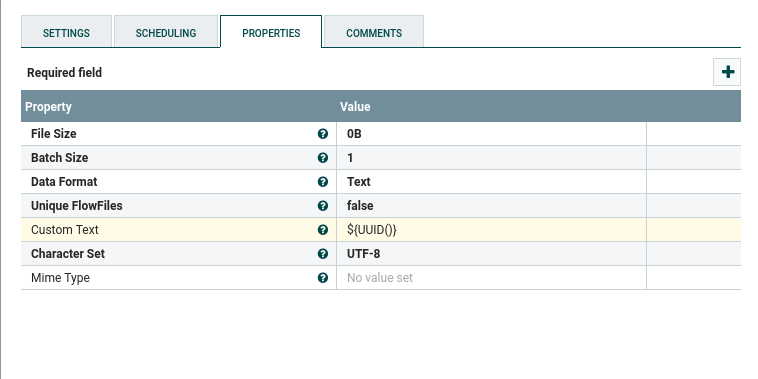


Рисунок 74 – Изменение обработчика

Теперь мы готовы сохранить три обработчика в качестве шаблона. Этот небольшой поток данных может имитировать источник потоковых данных, в котором фактические данные типа «ключ:значение» смешанный со случайным шумом.

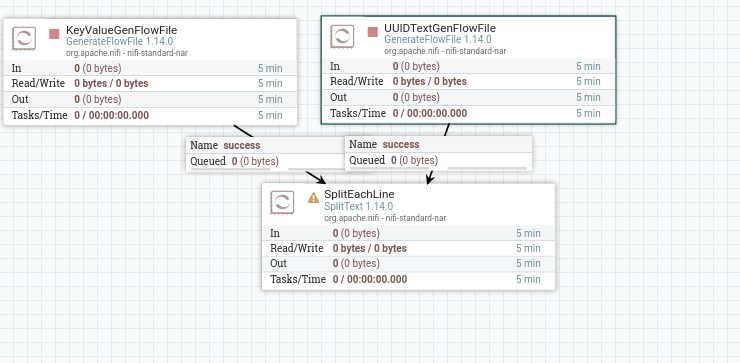


Рисунок 75 – Заготовка для шаблона

Выберем все компоненты, включая три обработчика и две связи. Выберем значок Create Template на панели операций. Откроется окно Create Template. Назовем шаблон KV Datasource и нажмем Create.

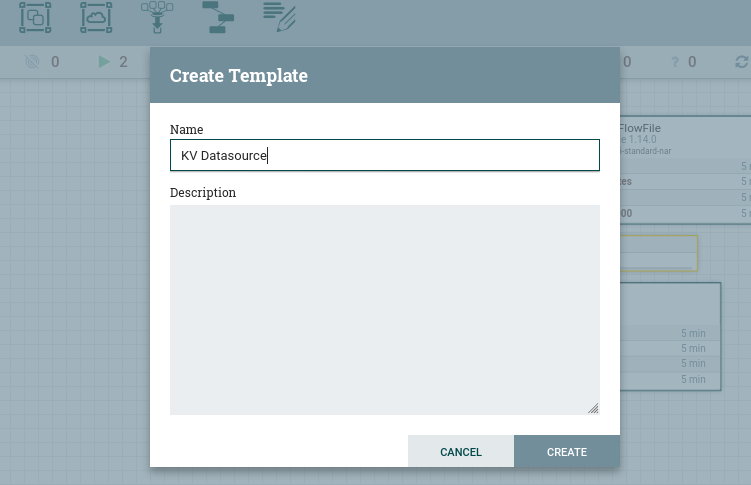


Рисунок 76 – Создание шаблона

Создадим новый поток данных, используя шаблон, который мы только что создали. Выберем значок Templates на панели компонентов и перетащим ее указателем мыши на холст. Во всплывающем меню выберем шаблон KV Datasource, который мы только что сохранили. Копия потока данных будет помещена на холст.

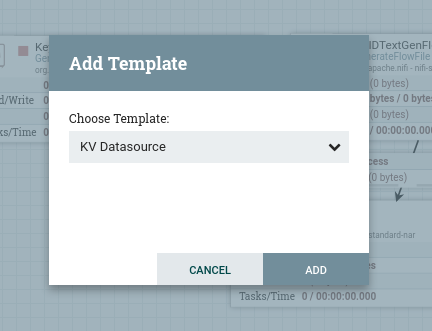


Рисунок 77 – Добавление шаблона

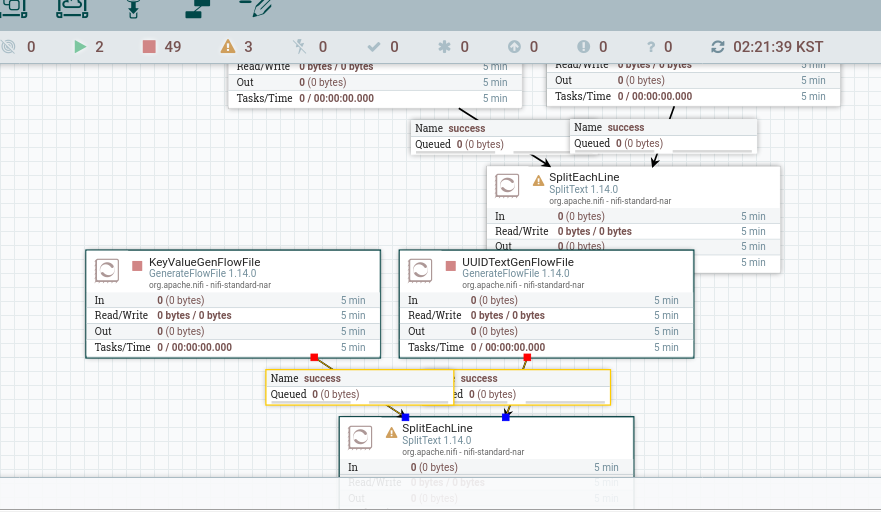


Рисунок 78 – Добавленный шаблон

Мы можем экспортировать шаблоны в виде XML-файлов для переноса в другие кластеры NiFi или просто для постоянного хранения в виде физического файла. Откройте Главное меню в правом верхнем углу экрана. Выберите пункт Templates из этого меню. Появится новое всплывающее окно со всеми сохраненными шаблонами в текущем кластере NiFi. В крайнем правом углу находится значок загрузки и корзины. Используем значок корзины, чтобы удалить сохраненные шаблоны. Используем значок загрузки, чтобы экспортировать выбранный шаблон в виде XML-файла.



Рисунок 79 – Список сохраненных шаблонов

Откроем терминал и перейдем в каталог Downloads. Перечислим (команда ls) содержимое каталога, и мы увидим сохраненный XML-файл. мы можем скопировать этот файл и использовать его в другом кластере NiFi.

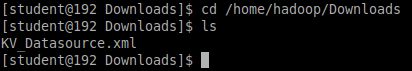


Рисунок 80 – Сохраненный шаблон в локальной системе

Откроем KV\_Datasource.xml с помощью редактора.

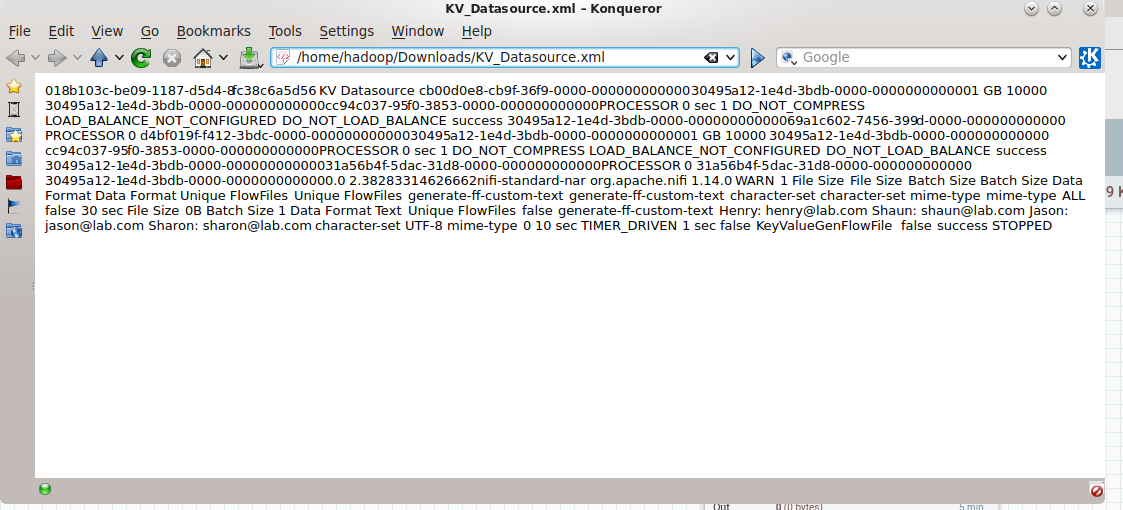


Рисунок 81 – XML-файл

Удалим сохраненный шаблон из текущего кластера, чтобы мы могли импортировать его на следующем шаге. Если шаблон с таким именем уже существует в кластере, мы не сможем его импортировать.

Нажмем на значок «Upload Template» в панели операций. Выберем значок лупы, чтобы просмотреть файлы в вашем каталоге. Выберем каталог Downloads на левой вкладке и оттуда выберем файл KV\_Datasource.xml, который мы только что сохранили на предыдущем шаге

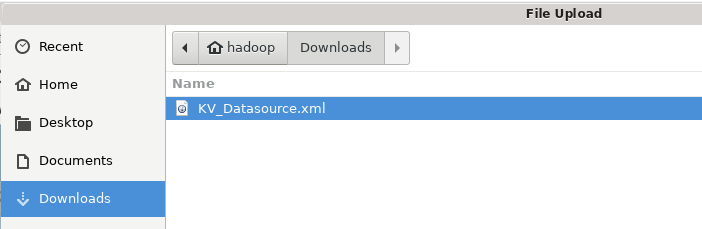


Рисунок 82 – Загрузка шаблона

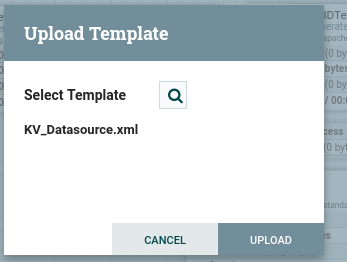


Рисунок 83 – Загрузка шаблона

После выбора XML-файла нажмем на кнопку Upload, чтобы завершить операцию. Всплывающее окно Success подтвердит успешное выполнение операции.

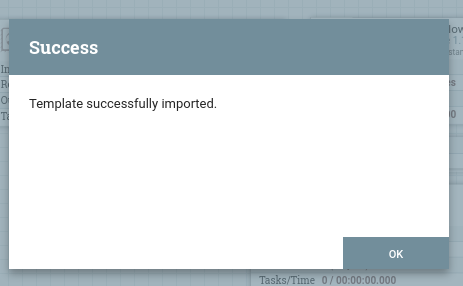


Рисунок 84 – Завершение загрузки шаблона

Вернемся в главное меню для подтверждения того, что шаблон импортирован и является частью NiFi кластера. Проверим шаблон, повторив операции с шага 2.1, чтобы создать копию исходного потока данных KV\_Data на холсте.

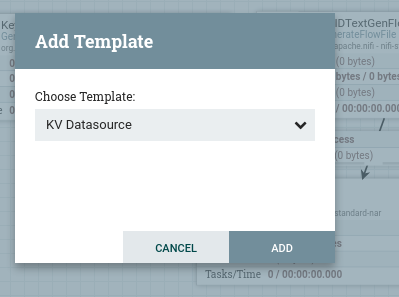


Рисунок 85 – Выбор шаблона

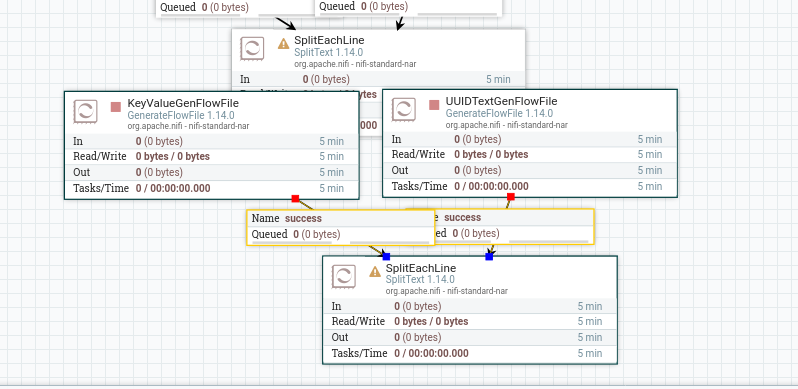


Рисунок 86 – Выбор шаблона

Часто бывает полезно создавать резервные копии всего вашего холста со всеми потоками данных в нем.

Выберем все компоненты холсте. Перейдем в корневой холст, используя цепочку навигации на экране слева снизу. Выберем все компоненты, нажав Ctrl+A. Создадим шаблон из выбранных компонентов.



Рисунок 87 – Выбор компонентов на корневом холсте

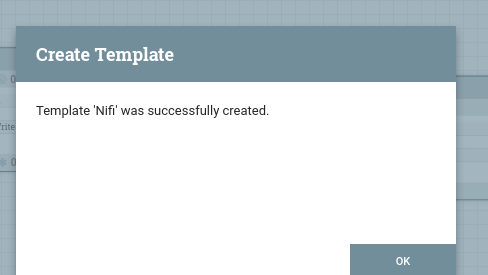


Рисунок 88 – Создание резервной копии

# Использование групп обработки

В этой лабораторной работе мы объединим использование шаблонов и групп обработки, чтобы упростить разработку потоков данных. До сих пор мы использовали только группы обработки для организации проектов в разные каталоги. Теперь мы будем использовать их для соединения меньших логических потоков данных в более крупные.

Для начала, настроим новую группу обработки Lab 5 так же, как мы делали это ранее.

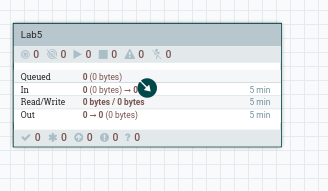


Рисунок 89 – Создание группы Lab 5

Мы создадим поток данных, в котором мы возьмем источник данных, объединим содержимое, сожмем его и сохраним на локальном диске.

Создадим источник данных. Добавим обработчик GenerateFlowFile на холст Lab 5. Назовем его Lab5-GenerateFlowFile. Настроим его на срабатывание каждые 2 секунды. Изменим свойство File Size на 5000 KB

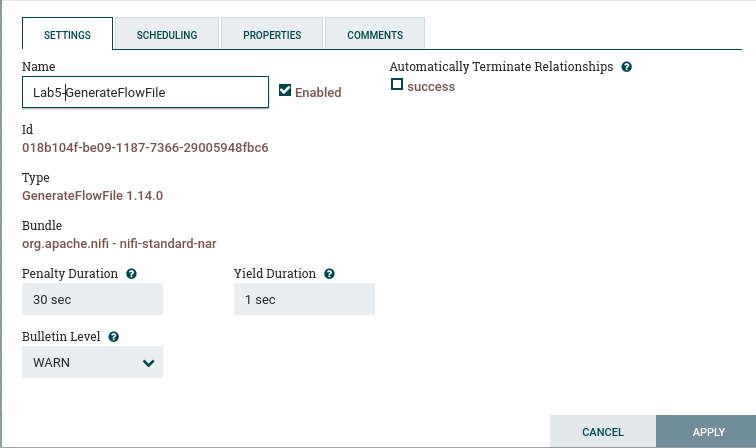


Рисунок 90 – Именование обработчика

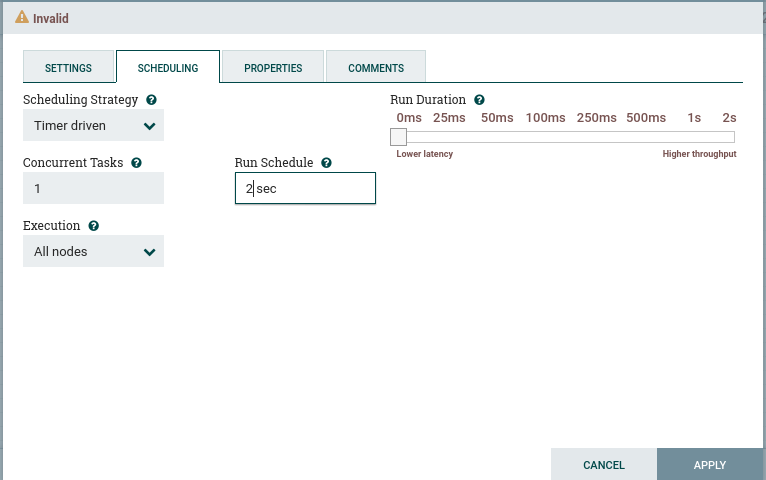


Рисунок 91 – Настройка времени срабатывания

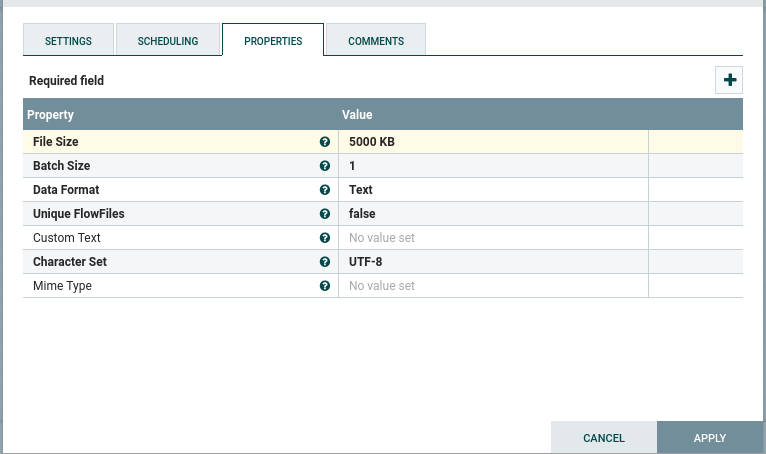


Рисунок 92 – Настройка свойств

Создадим поток данных Merge-Compress-Save. Добавим обработчик MergeContent. Настроим отношения failure и original, как автозавершаемые. В Properties установим Merge Format (формат слияния), как TAR. Изменим Minimum Group Size (минимальный размер группы) на 30 MB.

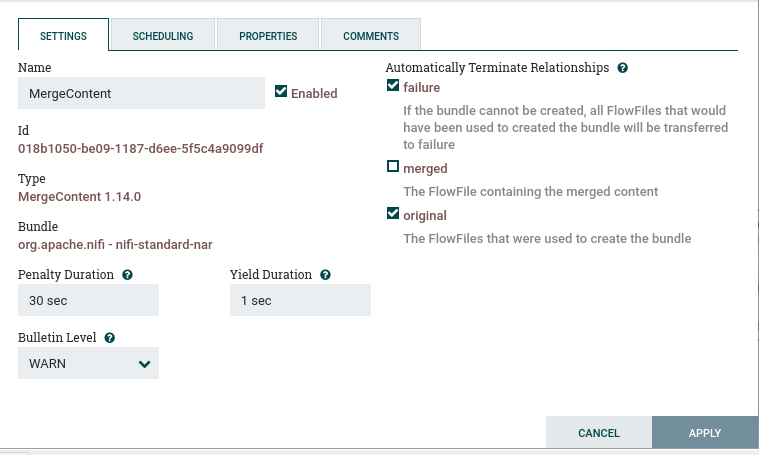


Рисунок 93 – Настройка обработчика

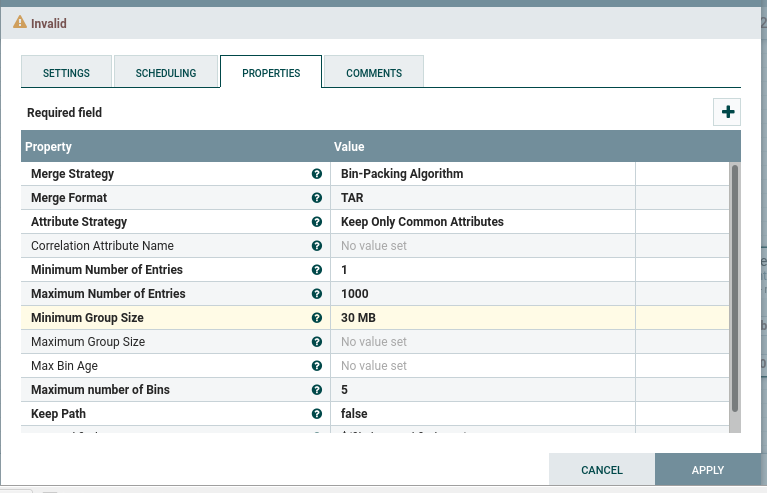


Рисунок 94 – Настройка свойств обработчика

По сути, мы будем ждать, пока размер слитого содержимого не станет 30 МБ, а затем упакуем его в файл формата tar.

Добавим обработчик CompressContent. В Properties убедимся, что Mode установлен compress. Изменим Compression Format (формат сжатия) на gzip.

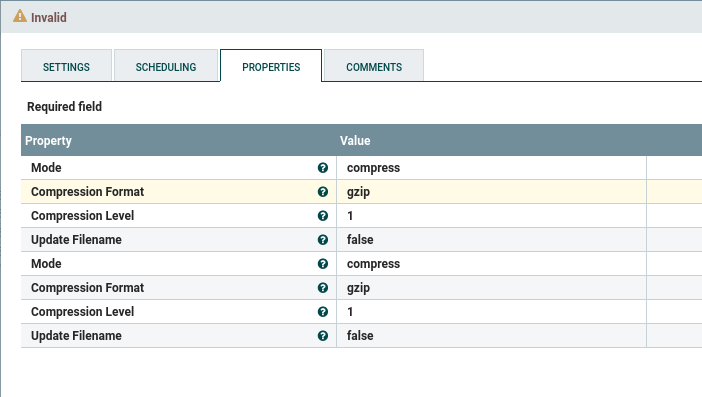


Рисунок 95 – Настройка свойств обработчика

Добавим обработчик PutFile. Настроим автозавершение отношения success. После сохранения файла, нам не нужно хранить FlowFile’ы. В Properties, изменим Directory (директорию) на /tmp/nifi/lab5. Изменим Conflict Resolution Strategy (правило решения конфликтов) на replace (замена). Убедимся, что в Create Missing Directories (создавать несуществующие папки) установлено true. Установим Owner (владелец) на student. Установим Group (группа) на student.

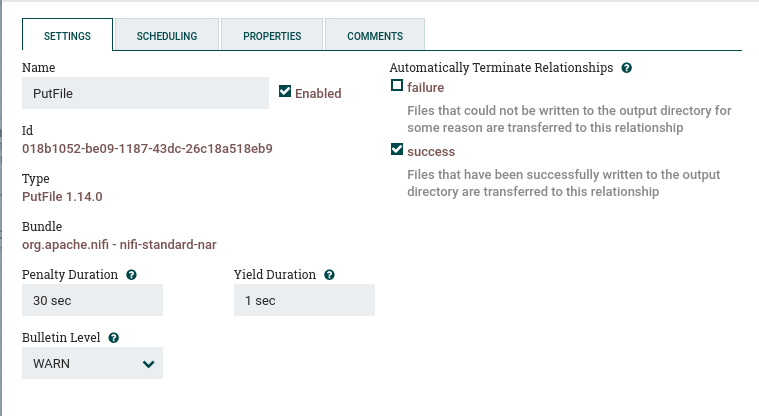


Рисунок 96 – Настройка обработчика

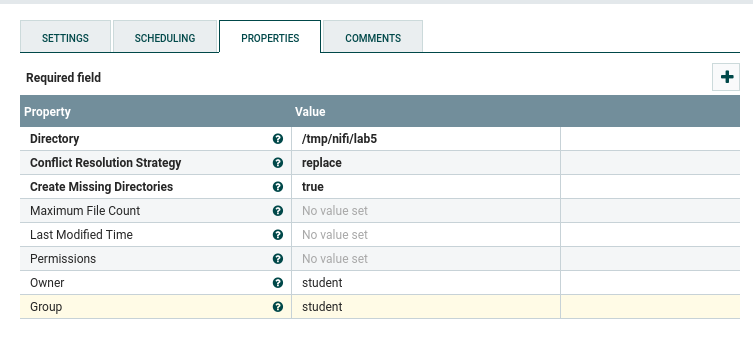


Рисунок 97 – Настройка свойств обработчика

Подключим обработчик Lab5-GenerateFlowFile к обработчику MergeContent в отношении success.

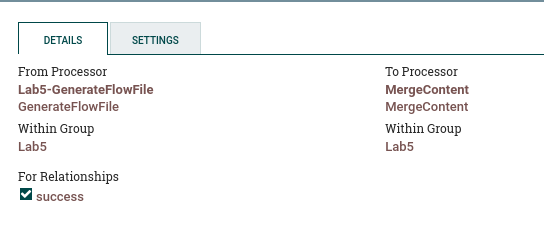


Рисунок 98 – Соединение обработчиков

Подключим обработчик MergeContent к обработчику CompressContent в отношении merged.

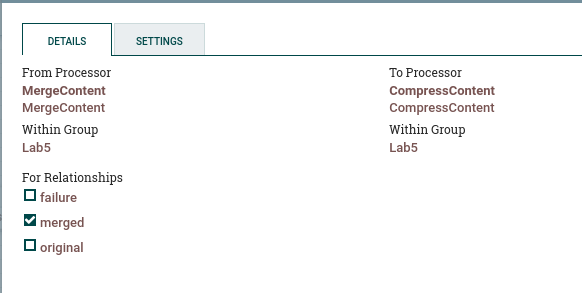


Рисунок 99 – Соединение обработчиков

Подключим обработчик CompressContent к обработчику PutFile в отношении success.

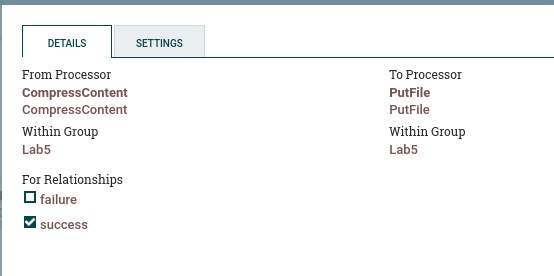


Рисунок 100 – Соединение обработчиков

Добавим воронку и соединим ее с обработчиком CompressContent в отношении failure.

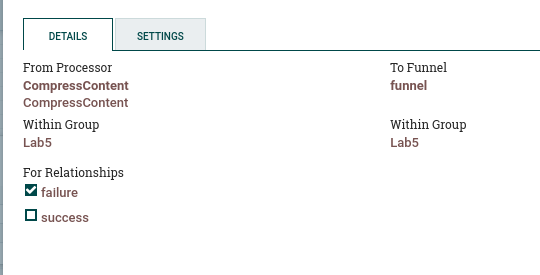


Рисунок 101 – Соединение обработчиков

Добавим воронку и подсоединим к ней обработчик PutFile в отношении failure.

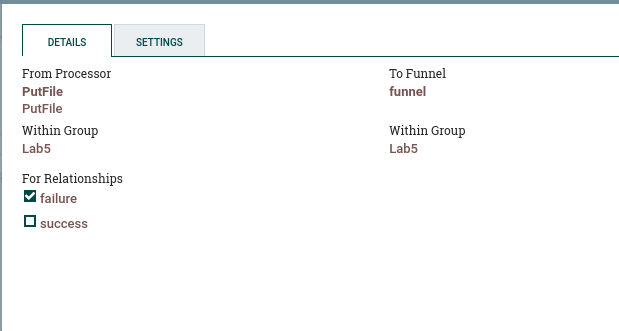


Рисунок 102 – Соединение обработчиков

Поток данных выглядит так, как показано ниже.

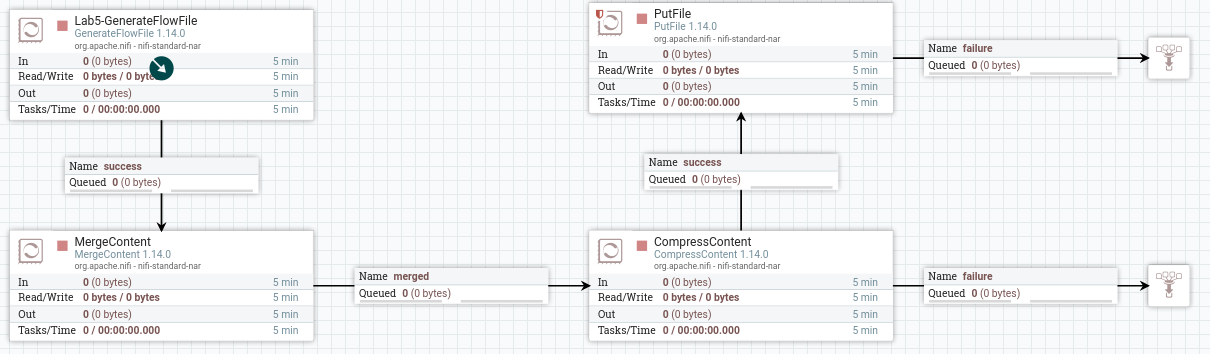


Рисунок 103 – Итоговый холст

Запустим обработчик Lab5-GenerateFlowFile примерно на 30 секунд, пока FlowFile’ы не будут поставлены в очередь ниже по потоку. Остановим обработчик, а затем откроем очередь и проверим FlowFile’ы. Обратим внимание на размер файла.

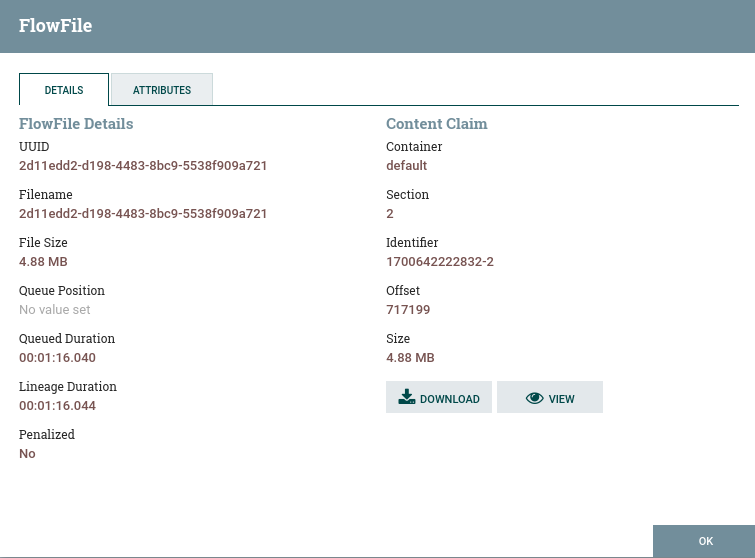


Рисунок 104 – Свойства FlowFile’а

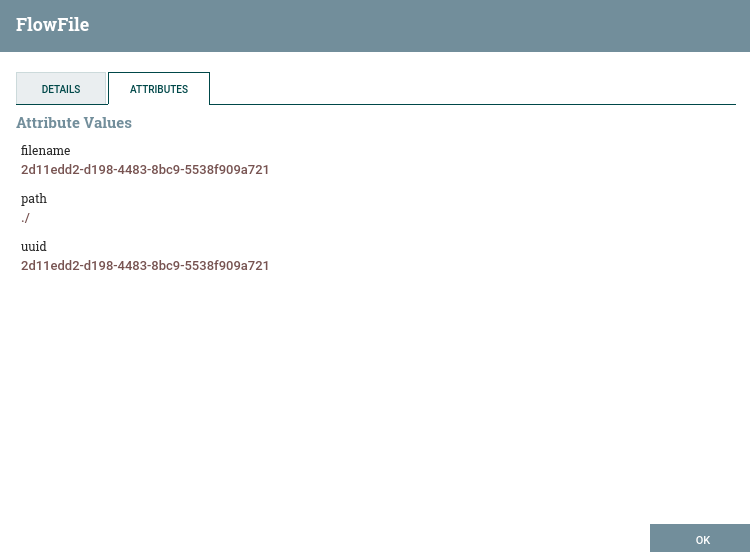


Рисунок 105 – Атрибуты FlowFile’а

Запустим обработчик MergeContent примерно на 15 секунд, пока FlowFile’ы не будут поставлены в очередь ниже по потоку. Остановим обработчик, а затем откроем очередь и проверим FlowFile’ы. Обратим внимание на размер файла. На вкладке Attributes посмотрим на новые атрибуты, созданные этим обработчиком.

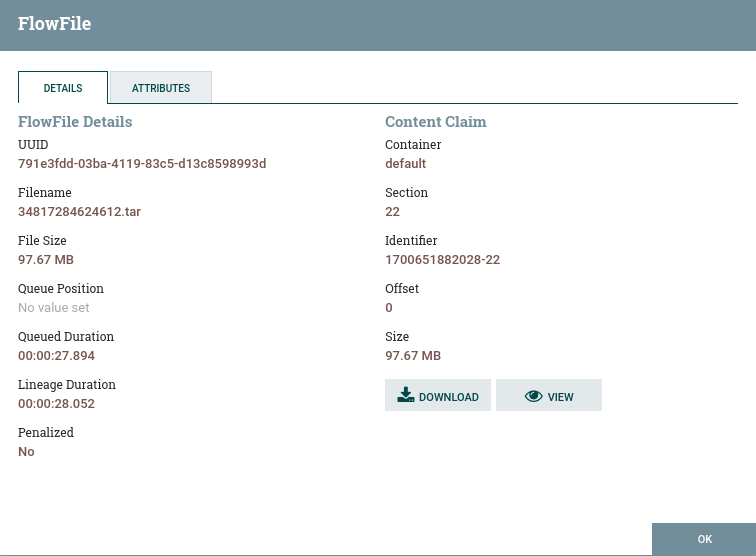


Рисунок 106 – Свойства FlowFile’а

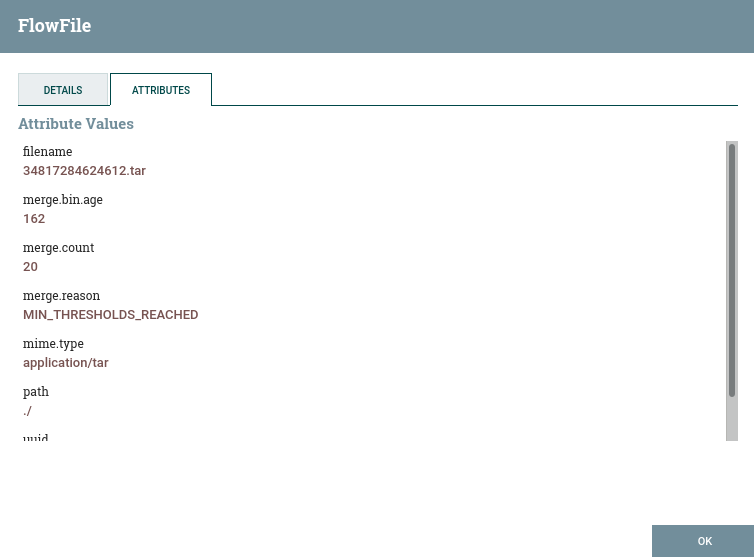


Рисунок 107 – Атрибуты FlowFile’а

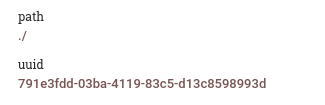


Рисунок 108 - Атрибуты FlowFile’а

Атрибут merge.count показывает, сколько FlowFile’ов было объединено вместе. У нас было около 20 файлов потока. Атрибут merge.reason гласит, что был достигнут минимальный порог. Если прошло определенное количество времени (Max Bin Age), обработчик упакует все FlowFile’ы, собранные до сих пор, и отправит вниз по потоку.

Выполним CompressContent до тех пор, пока FlowFile’ы не будут поставлены в очередь ниже по потоку. Остановим обработчик. Сначала взглянем на статистику за 5 минут. Здесь показан 1 входной файл. Чтение/запись 97,67 МБ/80,71 МБ и 1 выходной файл. Это указывает на то, что этот обработчик обработал 1 поступающий FlowFile и отправил 1 FlowFile. Объем операций чтения/записи для завершения этого процесса показан на рисунке.

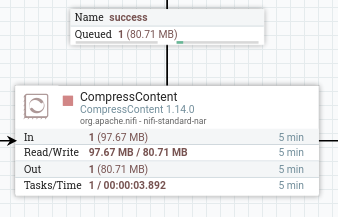


Рисунок 109 – Работа CompressContent

Откроем очередь и проверим FlowFile’ы. Обратим внимание, что размер файла соответствует выводам, показанным из приведенной выше статистике за 5 минут. Кроме того, на вкладке Attributes mime.type был изменен с tar на zip, поскольку он был сжат с помощью gzip.

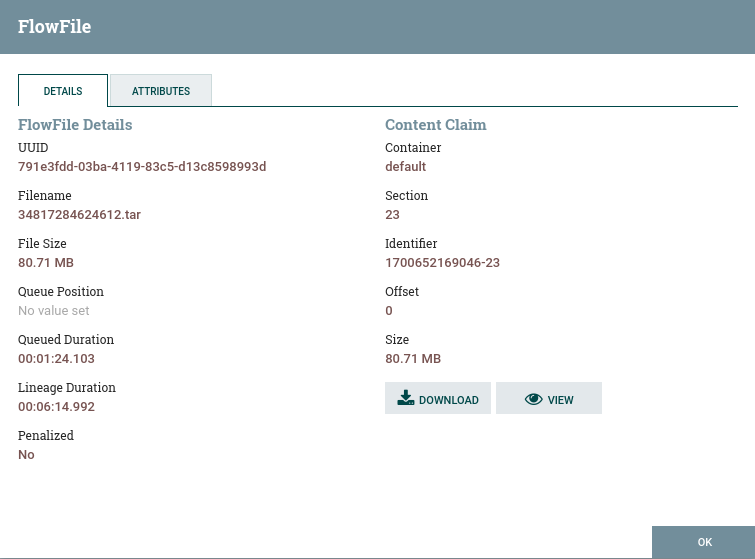


Рисунок 110 – Свойства FlowFile’а



Рисунок 111 – Атрибуты FlowFile’а



Рисунок 112 – Атрибуты FlowFile’а

Запустим обработчик PutFile и дадим ему поработать до тех пор, пока очередь не будет очищена. Проверим, что сбоя не было, убедившись, что очередь failure в воронке остается пустой.

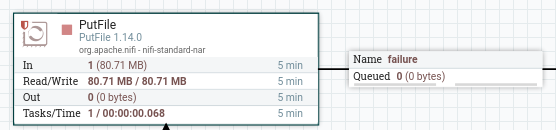


Рисунок 113 – Работа PutFile

Откроем новый терминал и перейдем в папку /tmp/nifi/lab5. Получим список файлов папки:

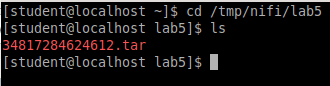


Рисунок 114 – Файл в локальной системе

Распакуем файл и снова перечислим содержимое каталог. Мы должны увидеть то же количество файлов, что и merge.count из приведенного выше шага. В нашем случае это было 8.

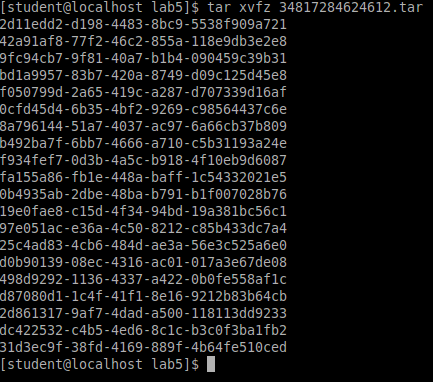


Рисунок 115 – Распаковка архива

Используем команду more (less) для просмотра содержимого любого из этих файлов.



Рисунок 116 – Просмотр содержимого файла

Создадим поток данных с группами обработки. Часть нашего потока данных, связанная с объединением, сжатием и сохранением (Merge-Compress-Save), кажется чем-то, что мы часто можем использовать. Теперь мы создадим из него группу обработки и будем использовать ее как часть потока данных.

Убедимся, что все обработчики остановлены, а все очереди пусты. Удалим отношение между обработчиками Lab5-GenerateFlowFile и MergeContent. Уберем Lab5-GenerateFlowFile подальше, чтобы мы могли легко выбрать остальные обработчики и связи.

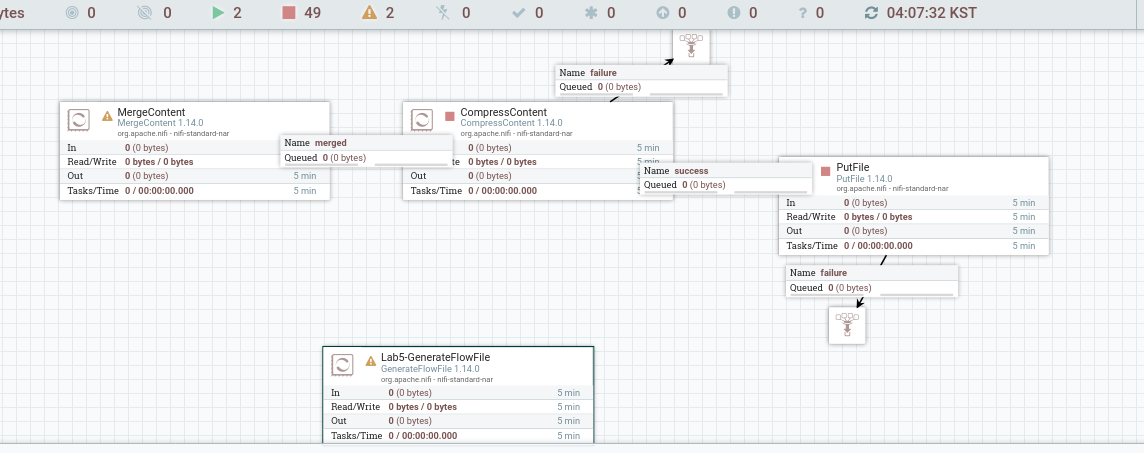


Рисунок 117 – Обработка холста

Создадим группу обработки из остальных обработчиков. Выберем все компоненты, кроме Lab5-GenerateFlowFile, вызовем контекстное меню. Выберем Group

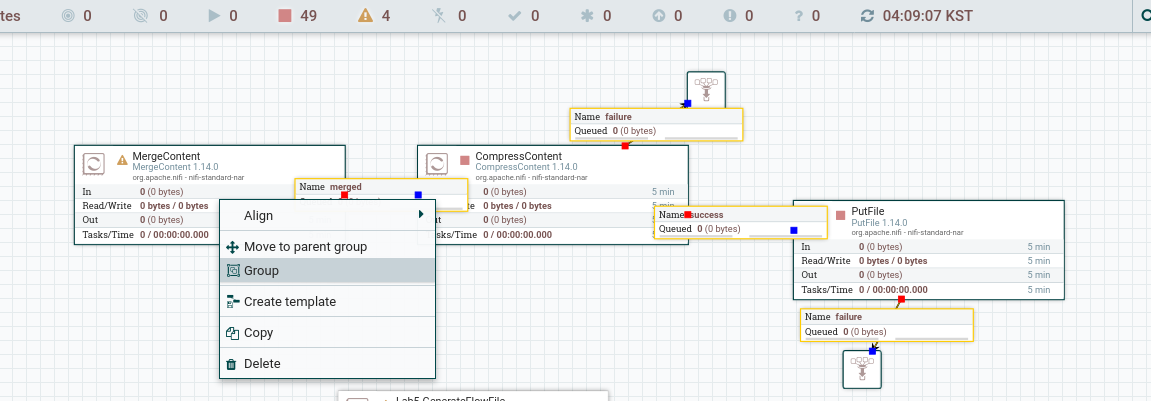


Рисунок 118 – Создание группы обработки

Введем Merge-Compress-Save в качестве Processor Group Name (имя группы обработки).

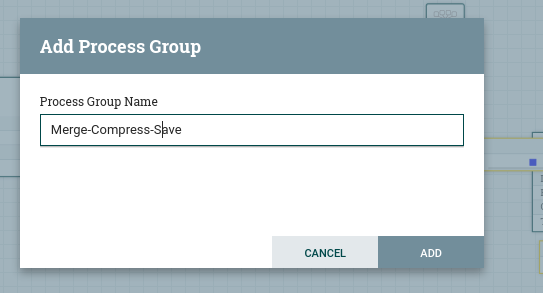


Рисунок 119 – Именование группы

Нажмем на кнопку Add, чтобы создать группу обработки. Все выбранные компоненты теперь будут вставлены во вновь созданную группу обработки.

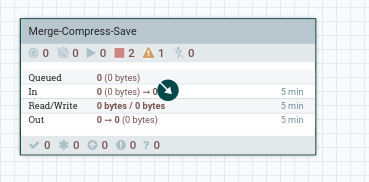


Рисунок 120 – Созданная группа

Дважды щелкнем по группе, чтобы войти в нее. Заметим, что появился желтый значок на обработчике MergeContent. Наведем на него указатель мыши для того, чтобы прочесть сообщение.

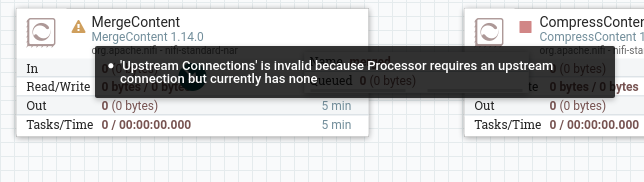


Рисунок 121 – Сообщение об ошибке

Обработчик MergeContent требует наличия следующего по потоку обработчика, но сейчас его нет. Раньше им был Lab5-GenerateFlowfile, но мы его отключили. Добавим Input Port к группе обработки, с помощью выбора соответствующего значка на панели компонентов. Назовем его Get-Datasource.

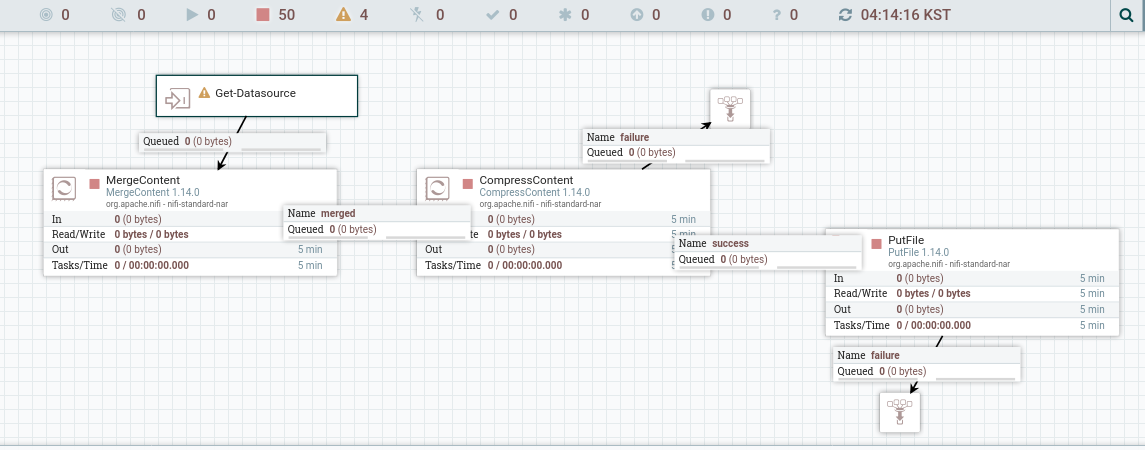


Рисунок 122 – Подключение Get-Datasource

#### Подключим обработчик Lab5-GenerateFlowFile к группе обработки Merge-Compress-Save в отношении success. Убедимся, что в To Input с правой стороны стоит значение Get-Datasource.

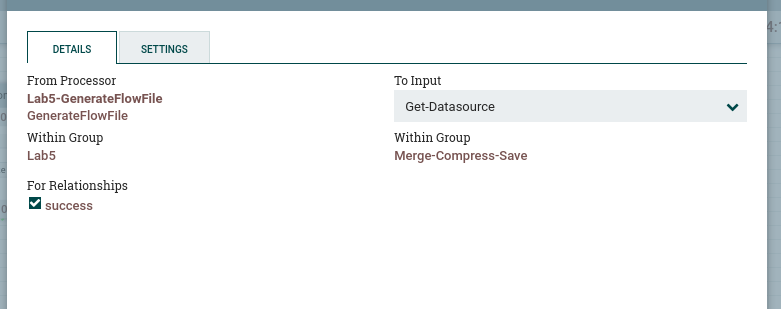


Рисунок 123 – Соединение обработчиков

Наш поток данных должен выглядеть так, как показано ниже.

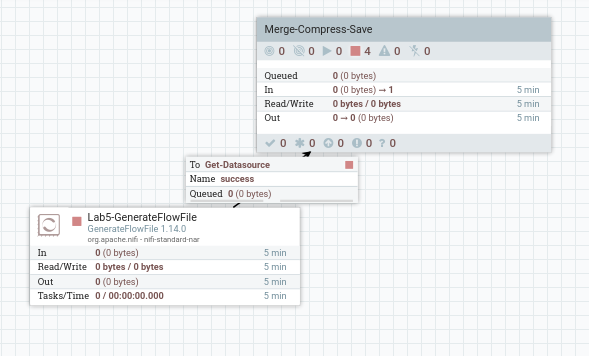


Рисунок 124 – Итоговый поток данных

Из терминала перейдем обратно в /tmp/nifi/lab5. Мы будем тестировать новый поток данных. Чтобы убедиться, что мы добавляем новые файлы, удалим все файлы в каталоге.



Рисунок 125 – Очищение папки

Вернемся к холсту Lab 5 и запустим все обработчики, включая группу обработки. Дадим потоку данных поработать около минуты, а затем остановим его. Мы должны увидеть активность в 5-минутной статистике как для обработчика Lab5-GenerateFlowFile, так и для группы обработки Merge-Compress-Save.

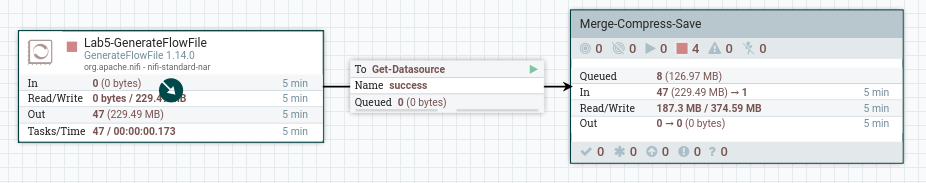


Рисунок 126 – Работа потока данных

Вернемся в терминал, чтобы проверить, что новые файлы созданы.

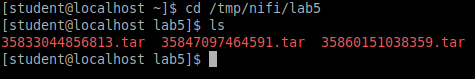


Рисунок 127 – Результат работы потока данных

# Настройка обратного давления для связей

В этой лабораторной работе мы будем исследовать обратное давление. Иногда обработчик может испытывать задержки, приводящие к накоплению очереди из входящего потока. Если очередь становится слишком большой, мы можем настроить обратное давление таким образом, чтобы вышестоящий обработчик перестал отправлять новые потоки данных вниз по потоку. Это, в свою очередь, может привести к активации рекурсивного обратного давления. Другими словами, поскольку вышестоящий обработчик прекратил обработку FlowFile’ов, его вышестоящая очередь, в свою очередь, начнет накапливаться. Это накопление может вызвать другую политику обратного давления, в результате чего вышестоящий обработчик или несколько вышестоящих обработчиков также прекратят обработку.

Создаем группу обработки и назовем ее Lab 6.

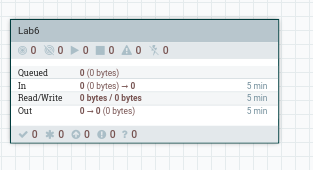


Рисунок 128 – Создание группы

В работе 5 мы создали очень полезный поток данных. Поток данных был преобразован в группу обработки с именем Merge-Compress-Save. Слияние и сжатие содержимого с последующим его сохранением — это часто встречающийся сценарий. Вернемся в лабораторную работу 5 и создадим шаблон из этой группы обработки. Назовем его Merge-Compress-Save.

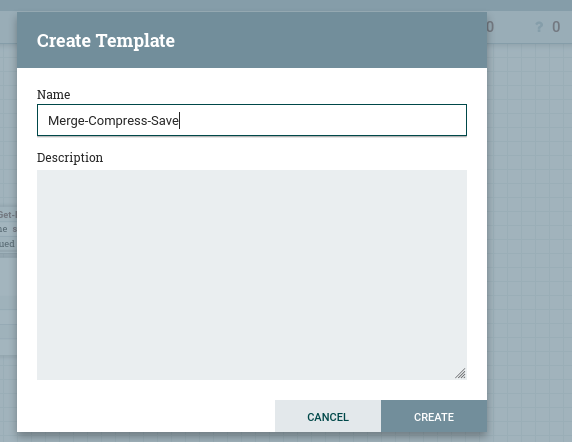


Рисунок 129 – Создание шаблона

* 1. В лабораторной работе 5 мы сохранили другой шаблон, который генерирует пары «ключ:значение». Создадим ещё один поток данных из шаблона KV Datasource. Холст показан на рисунке ниже

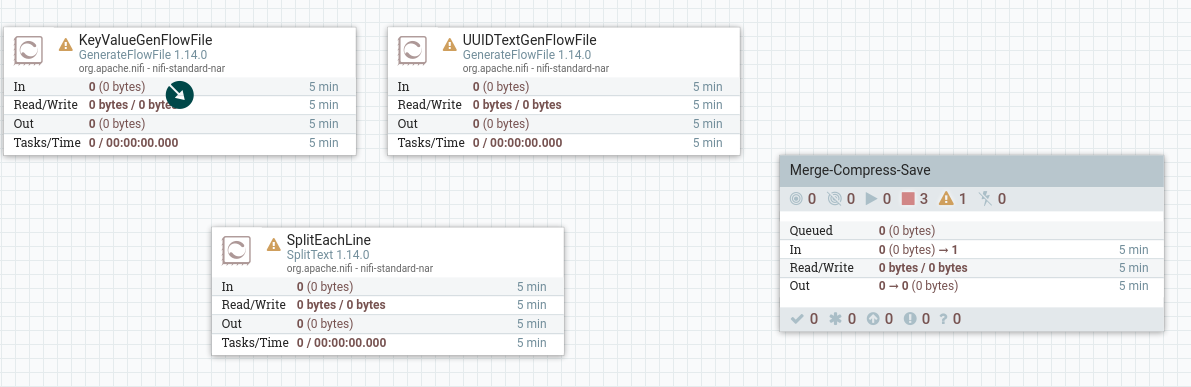


Рисунок 130 – Импорт шаблона KV Datasource

Группы обработки значительно упрощают процесс передачи данных. Теперь, когда мы знаем, как создавать группы обработки, возьмем три обработчика из шаблона KV Datasource и создадим новую группу обработки. Назовем ее KV Datasource. Наш поток данных должен выглядеть так, как показано ниже.

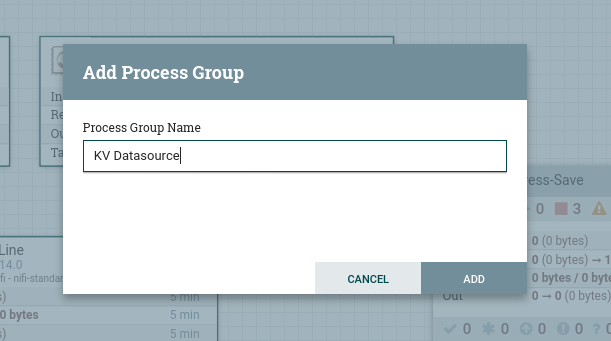


Рисунок 131 – Создание группы



Рисунок 132 – Итоговый холст

Изменим группу обработки KV Datasource. Войдем в группу KV Datasource. Изменим Run Schedule у обработчика KeyValueGenFlowFile на 1 секунду.

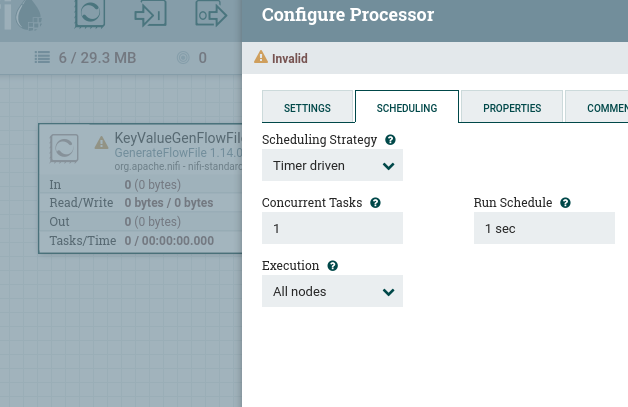


Рисунок 133 – Настройка времени срабатывания

Для этой лабораторной работы нам не понадобится UUIDTextGenFlowFile. Удалим его.

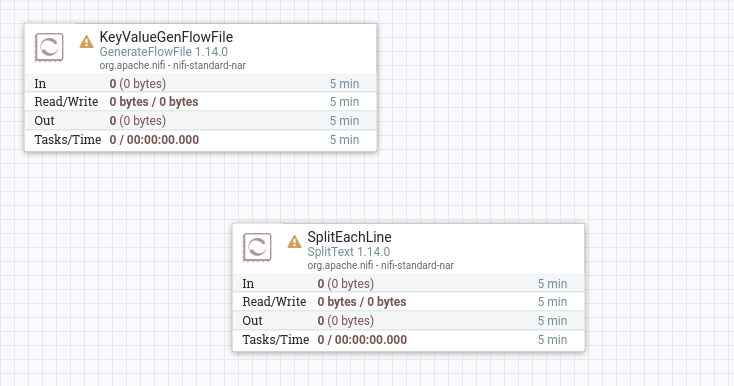


Рисунок 134 – Удаление обработчика

Скопируем обработчик Lab2-ExtractText из Lab2**.**

Добавим выходной порт и назовем его «From KV»

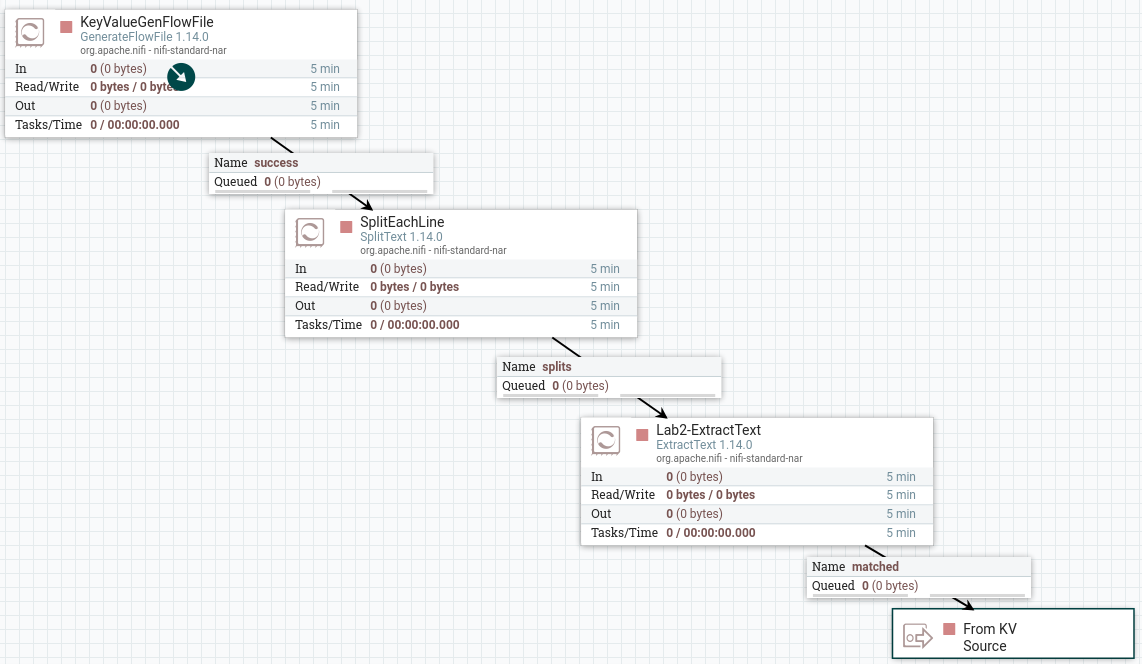


Рисунок 135 – Итоговый холст для группы KV Datasource

Настройте обратное давление

Щелкнем правой кнопкой мыши на соединении между Lab2-Extract Text и SplitEachLine, чтобы вызвать контекстное меню. Выберем пункт Configure. На вкладке Settings изменим значение Back Pressure Object Threshold на 100

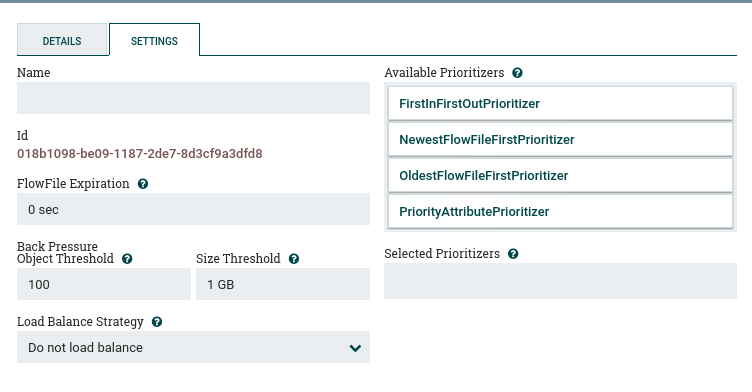


Рисунок 136 – Настройка обратного давления

Таким же образом установим значение Back Pressure Object Threshold на 30 между KeyValueGenFlowFile и SplitEachLine. Это максимальное число FlowFile’ов, что может быть помещено в очередь до срабатывания обратного давления.

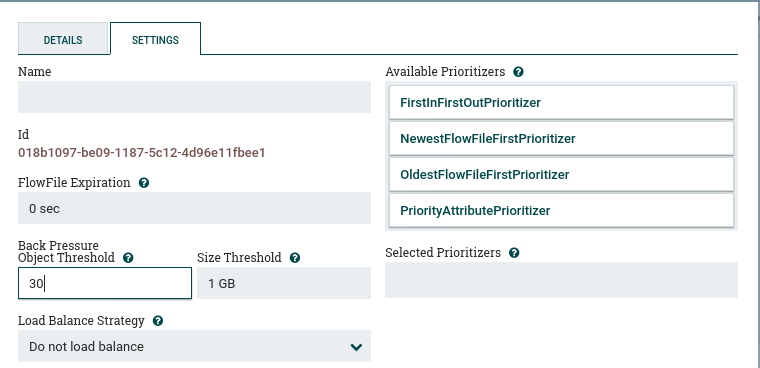


Рисунок 137 – Настройка обратного давления

Протестируем настройки политики обратного давления по количеству. Запустим KeyValueGenFlowFile. Обратим внимание на нисходящую очередь к обработчику SplitEachLine. По мере того, как все больше и больше файлов потока выстраиваются в очередь, она будет указывать на то, что буфер заполняется. Очередь имеет цветовую маркировку, чтобы показать, какая часть буфера заполнена до того, как будет применено обратное давление. Зеленый цвет означает 0-60%, желтый - 61-85%, а красный - 86-100%.

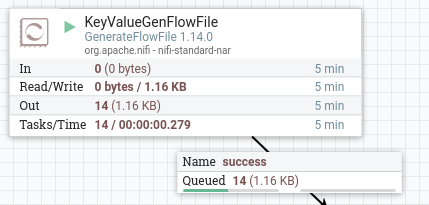


Рисунок 138 – Работа политики обратного давления

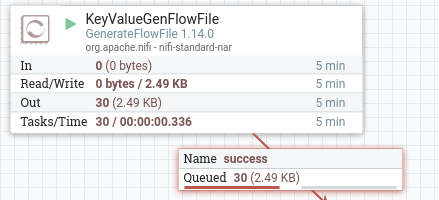


Рисунок 139 – Работа политики обратного давления

Остановим все обработчики и очистим очереди. Теперь запустим обработчики KeyValueGenFlowFile и SplitEachLine. Понаблюдаем за очередями.

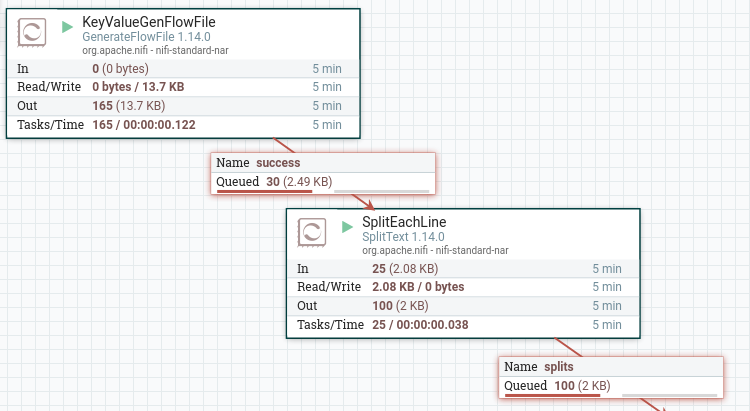


Рисунок 140 – Работа политики обратного давления

Обратим внимание, что справа от текущего, по-видимому, есть еще один пороговый маркер. Текущий маркер указывает пороговое значение для количества FlowFile’ов. Пока еще неактивный маркер справа предназначен для отображения пороговых значений размера.

Протестируем настройки политики обратного давления по размеру. Остановим все обработчики и очистим все очереди. Настроим связь splits и изменим Size Threshold на 1000 B.

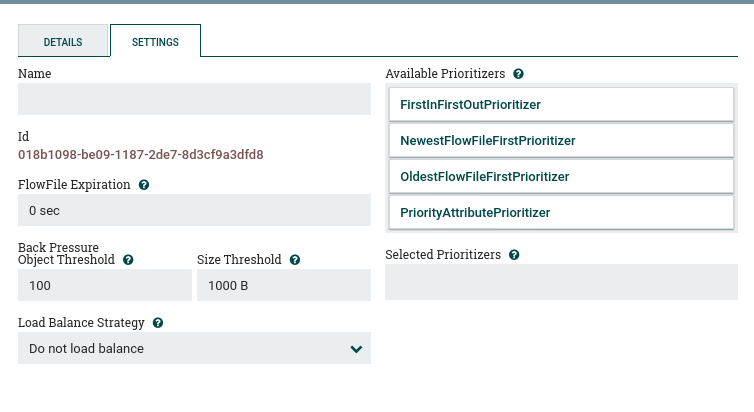


Рисунок 141 – Настройка обратного давления по размеру

Настроим связь success и изменим Size Threshold на 3000 B.

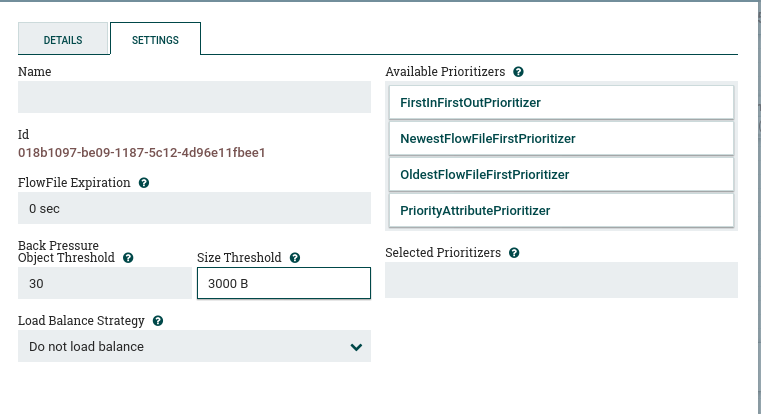


Рисунок 142 - Настройка обратного давления по размеру

Перезапустим обработчики KeyValueGenFlowFile и SplitEachLine и снова понаблюдаем за очередями.

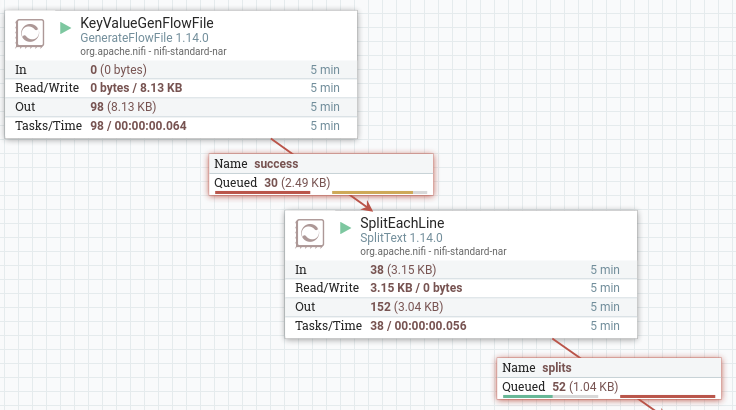


Рисунок 143 – Работа политики обратного давления по размеру

Политика обратного давления имеет два параметра. Существует настройка общего количества FlowFile’ов, поставленных в очередь, и общий размер FlowFile’ов, поставленных в очередь. Цветовые маркеры обозначают количество файлов потока слева и размер FlowFile’ов справа. Политика обратного давления срабатывает всякий раз, когда достигается любой из этих пороговых значений. Обратим внимание, что в очередях пороги размера достигается до порога количества FlowFile’ов.

Настроим поток данных с политикой обратного давления. Остановим все обработчики и очистим все очереди. Переместимся на холст Lab 6. Соединим группу обработки KV DataSource и группу Merge-Compress-Save. Наш поток данных показан на рисунке ниже.

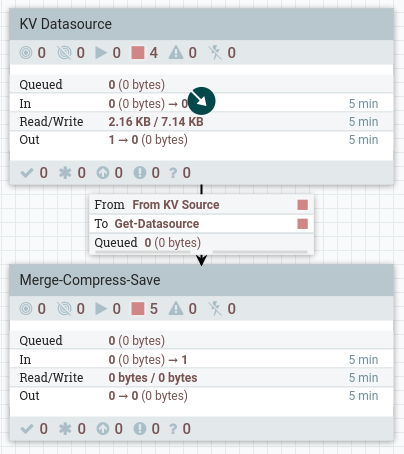


Рисунок 144 – Соединение групп обработки

Запустим обе группы обработки. Понаблюдаем за очередями. Обратим внимание, что очереди в KV Datasource продолжают заполняться, и срабатывает политика обратного давления. Также обратитм внимание, что размер выходных файлов значительно меньше, чем раньше.



Рисунок 145 – Работа политики обратного давления

Мы настроилиобработчик Merge Content в качестве обработчика получателя в группе обработки Merge-Compress-Save. Этот обработчик пытается объединить несколько FlowFile’ов до тех пор, пока не будут выполнены критерии для запуска. В предыдущей работе мы специально настроили все так, чтобы оно срабатывало после сбора около 30 МБ данных. Однако теперь, когда данные поступает из KV Datasource, FlowFile’ы значительно уменьшились. Таким образом, вместо запуска при минимальном размере группы (Minimum Group Size) он запускается при максимальном количестве записей (Maximum Number of Entries) в 1000 FlowFile’ов. Это приводит к тому, что выходные файлы становятся значительно меньше. Это также приводит к тому, что вышестоящие обработчики в KV Source запускают свои политики обратного давления, поскольку FlowFile’ы накапливаются в ожидании объединения их содержимого.



Рисунок 146 – Содержимое на локальной директории

# Работа с Hadoop в NiFi

В этой лабораторной работе мы изменим группу обработки Merge-Compress-Save и шаблон. До сих пор мы сохраняли данные только на локальном диске Linux. Теперь мы добавим HDFS в качестве дополнительного места хранения.

Создадим группу обработки Lab 7.

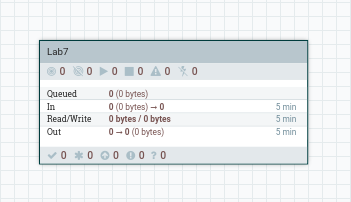


Рисунок 147 – Создание группы

Скопируем группу обработки Merge-Compress-Save путем добавления ее шаблона на холст. Войдите в группу Merge-Compress-Save. Удалим связь между CompressContent и PutFile. Удалим обработчик PutFile.

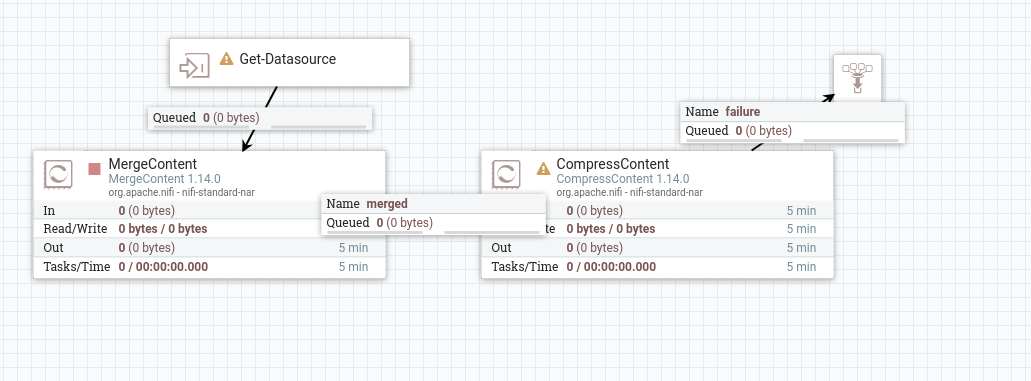


Рисунок 148 – Изменение группы Merge-Compress-Save

Добавим обработчик PutHDFS и настроим его.



Рисунок 149 – Добавление PutHDFS

Hadoop Configuration Resources (Ресурсы настроек Hadoop): Это место, где хранитятся конфигурационные XML-файлы Hadoop. Откроем терминал и войдем в систему под именем hadoop. Введем следующие команды, чтобы перейти к нужному каталогу.

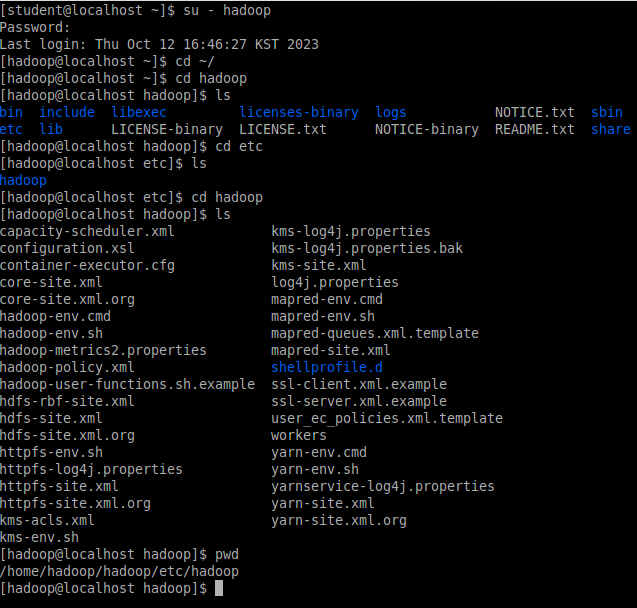


Рисунок 150 – Получение пути к конфигурационным XML-файлам Hadoop

Нас интересуют core-site.xml и hdfs-site.xml

В Hadoop Configuration Resources введем следующие два пути в одной строке, разделяя их запятой:

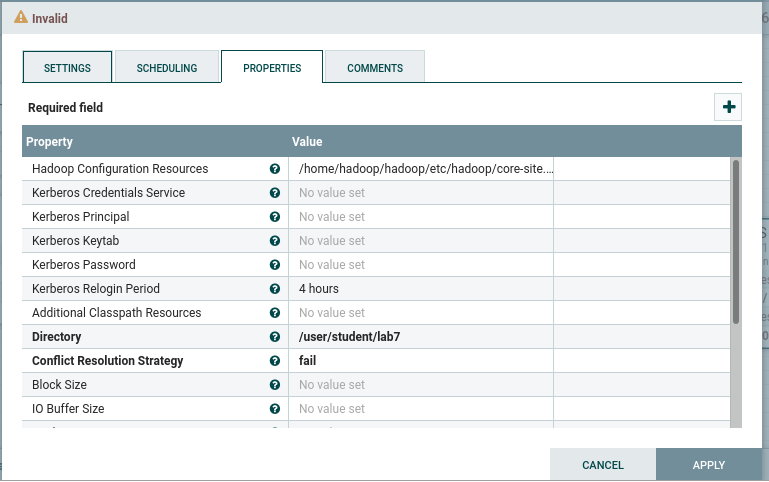


Рисунок 151 – Настройка обработчика

Directory (Директория):/user/student/lab7. Conflict Resolution Strategy (Правило разрешения конфликтов):fail. Remote Owner (Удаленный владелец):student. Remote Group (Удаленная группа):student.

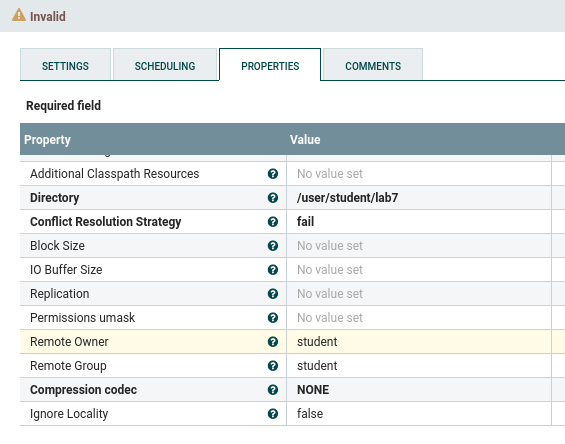


Рисунок 152 – Настройка обработчика

Сделаем отношение success автозавершаемым.

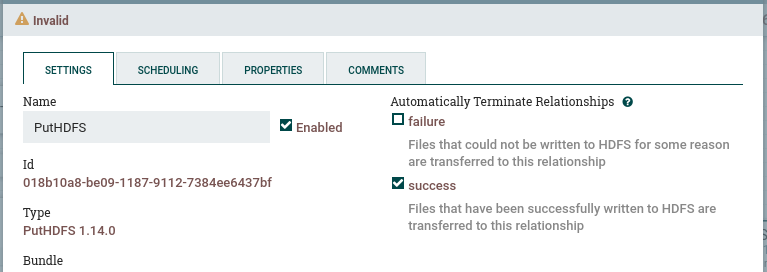


Рисунок 153 – Настройка обработчика

Подключим обработчик CompressContent к обработчику PutHDFS в отношении success

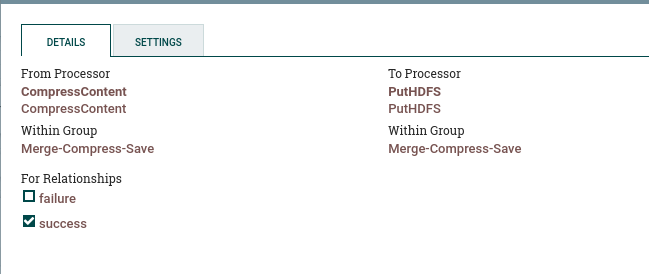


Рисунок 154 – Соединение обработчиков

Подключим PutHDFS к воронке в отношении failure

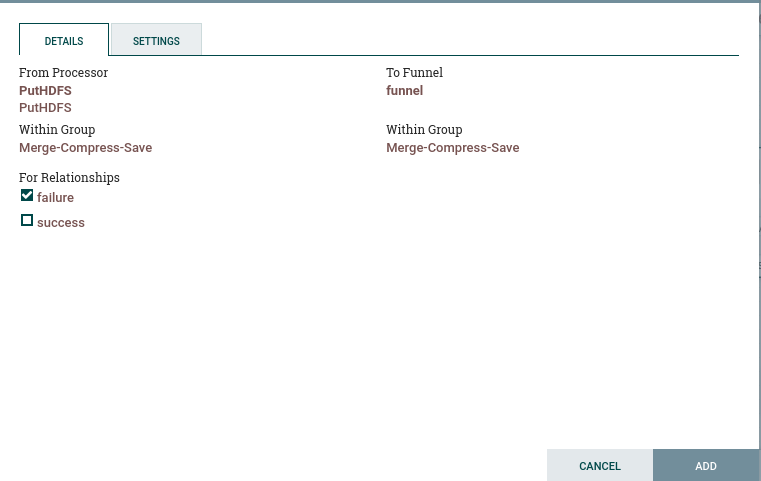


Рисунок 155 – Соединение обработчиков

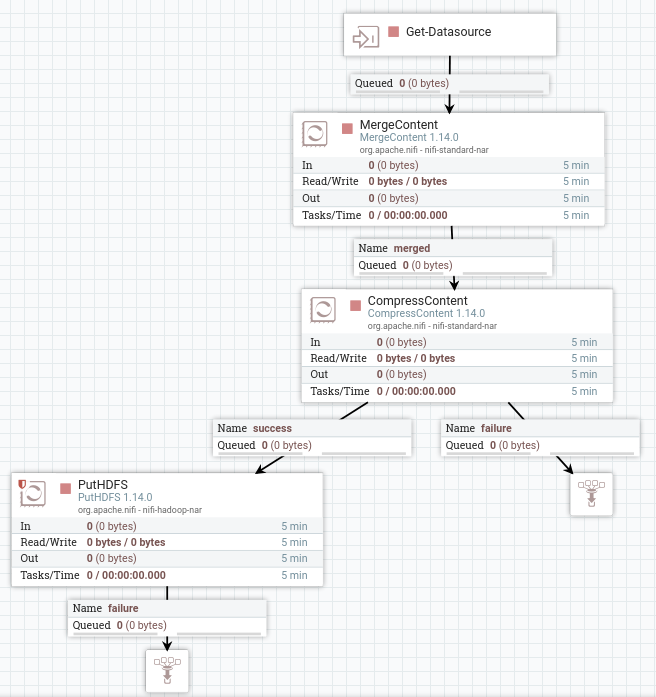


Рисунок 156 – Итоговый холст Merge-Compress-Save

Изменим группу обработки Merge-Compress-Save и сохраним ее в качестве шаблона.

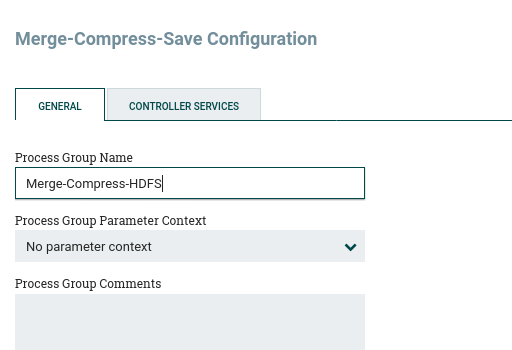


Рисунок 157 – Изменение шаблона

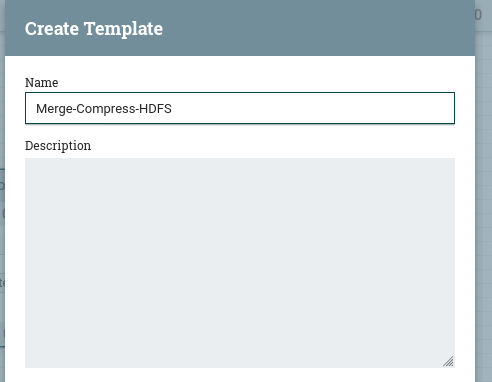


Рисунок 158 – Создание шаблона

Добавим новый источник данных. Скопируем Lab5-GenerateFlowfile и вставим его в группу обработки Lab 7. Подключим источник данных к группе обработки Мerge-Compress-HDFS.

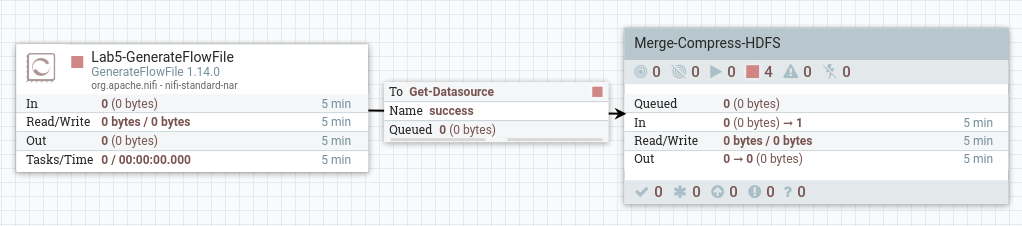


Рисунок 159 – Итоговый холст Lab 7

Запустим Lab5-GenerateFlowFile и Merge-Compress-HDFS. Есть ошибка в обработчике PutHDFS, появилась новая сводка. Наведем мышь на значок сводки, чтобы посмотреть сообщение.

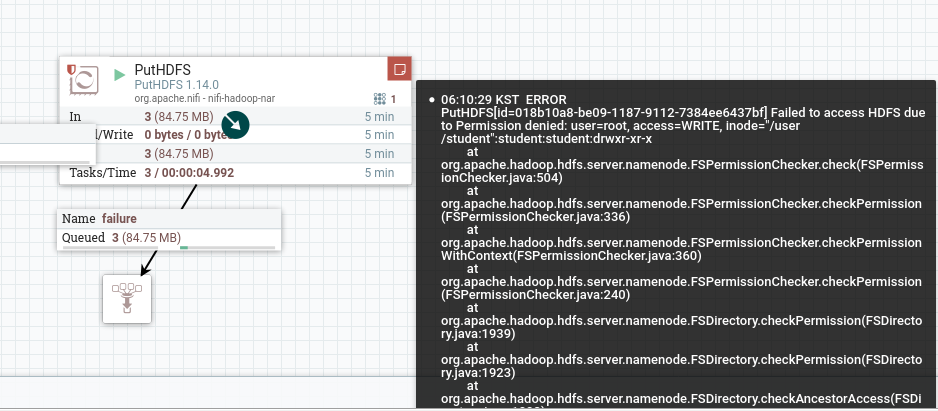


Рисунок 160 – Ошибка в работе

Это произошло потому, что NiFi в настоящее время работает от имени пользователя root в Linux. В то время как пользователь root имеет полный контроль над всеми локальными каталогами и файлами, это не относится к файловой системе HDFS. ERROR (Ошибка) возникает из-за того, что пользователь root пытается сохранить файл в каталог HDFS, принадлежащий пользователю student.

К счастью, есть способ выкрутиться. Просмотрим информацию, предоставленную настройками Remote Owner и Remote Group. В нем говорится, что пользователь, запускающий NiFi, должен иметь права суперпользователя HDFS.

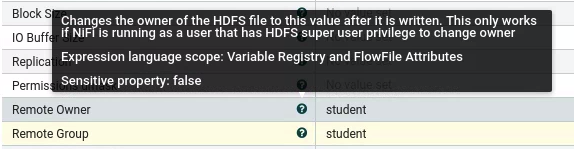


Рисунок 161 – Справка по настройкам

Почините обработчик PutHDFS установкой пользователя root в качестве суперпользователя HDFS.

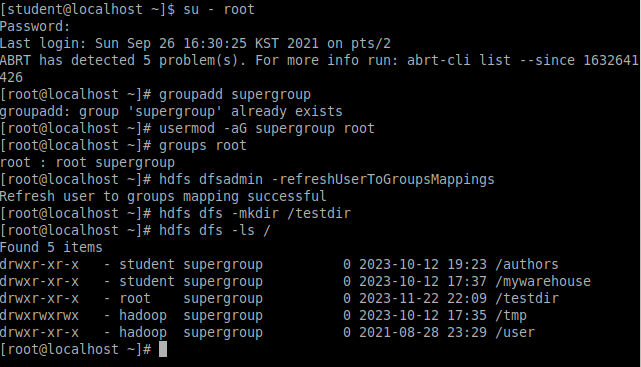


Рисунок 162 – Создание суперпользователя HDFS

Перезапустим поток данных и убедимся, что проблема решена. Посмотрим 5-минутную статистику у обработчиков, в особенности у PutHDFS, чтобы увидеть, что он может проводить операции Read и Write.

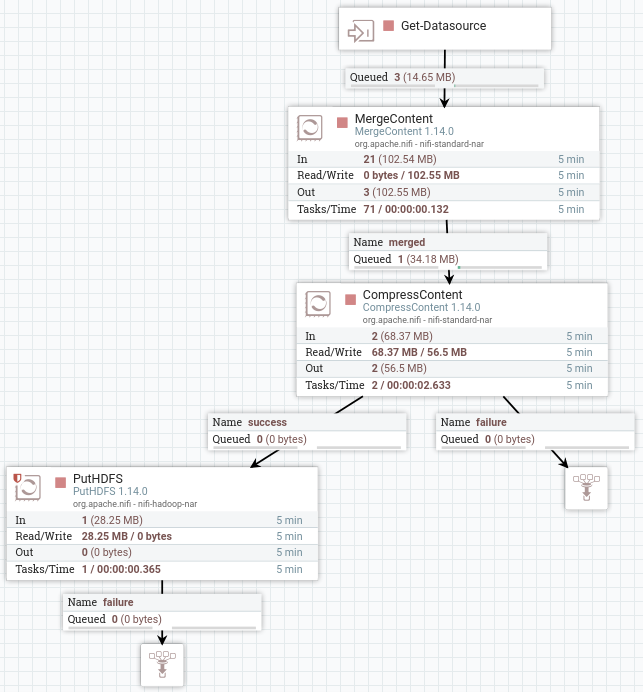


Рисунок 163 – Работа потока данных

Просмотрим содержимое директории назначения. Убедимся, что файлы сохранены.

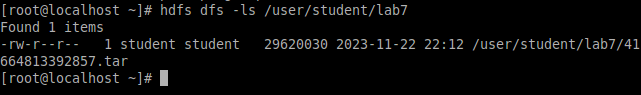


Рисунок 164 – Сохраненные файлы

Остановим все обработчики и очистим очереди.