**Методы анализа данных**

**Практическая работа №5**

**Работа с Apache Kafka**

**Создание темы, производителя и потребителя Kafka**

В этой работе мы будем использовать сервисы Kafka в командной строке для создания темы. Мы также будем использовать Kafka для создания производителей и потребителей и передачи данных через них.

## Настройте Apache Kafka

### Для того, чтобы сократить нагрузку на нашу виртуальную машину, мы остановим HBase и запустим только Apache Kafka.

### Остановите сервисы HBase

sudo stop-hbase.sh

### Примечание: Если HBase и не был запущен, то вы получите сообщение «no hbase master found».

### Перезапустите Kafka и Zookeeper

sudo systemctl stop kafka

sudo systemctl stop zookeeper

sudo systemctl start zookeeper

sudo systemctl status zookeeper

sudo systemctl start kafka

sudo systemctl status kafka

### Убедитесь, что zookeeper и kafka запущены. Если это не так, повторите шаг 1

###### 텍스트이(가) 표시된 사진 자동 생성된 설명

## Создайте тему Kafka

Создайте тему Kafka с названием topic1\_logs, которая будет содержать сообщения, представляющие строки в log файлах.

### Для создания темы используйте команду kafka-topics

#### Для создания темы topic1\_logs, выполните следующий код в терминале:

$kafka-topics --сreate \

--bootstrap-server localhost:9092 \

--replication-factor 1\

--partitions 1\

--topic topic1\_logs

Вы увидите сообщение: «Created topic topic1\_logs».

Примечание: Если ранее вы уже работали с Kafka в вашей среде, вы можете получить сообщение об ошибке, указывающее на то, что этот раздел уже существует. Вы можете не обращать внимания на ошибку.

#### Для отображения всех тем kafka, используйте ключ --list, и убедитесь, что тема, которую мы только что создали, отображена в списке.

$ kafka-topics --list \

--bootstrap-server localhost:9092

* 1. Просмотрите тему topic1\_logs.

$kafka-topics --describe topic1\_logs

--bootstrap-server localhost:9092

1. Создайте потребителей и производителей для темы

### Откройте 2 окна терминала и создайте производителя в одном окне, а потребителя в другом.

#### Из первого терминала запустите команду kafka-console-producer для запуска производителя.

$kafka-console-producer \

--broker-list localhost:9092 \

--topic topic1\_logs

텍스트이(가) 표시된 사진

자동 생성된 설명

#### Обратите внимание, что kafka-console-producer ожидает ввода текста. Текст, введенный здесь, станет сообщением в теме Kafka topic1\_logs.

#### Из второго терминала вызовите команду kafka-console-consumer для создания потребителя

kafka-console-consumer \

--bootstrap-server localhost:9092 \

--topic topic1\_logs \

--from-beginning

텍스트이(가) 표시된 사진

자동 생성된 설명

#### Обратите внимание, что kafka-console-consumer ожидает поступления сообщений в тему kafka topic1\_logs.

### Переименуйте терминалы.

#### В терминале производителя выберите Edit -> Rename Tab и введите имя «Producer».

#### Аналогично, назовите терминал потребителя «Consumer»

###### 텍스트이(가) 표시된 사진 자동 생성된 설명

### Создайте сообщения для темы topic1\_logs.

#### Начните писать что-нибудь в терминале **Producer**

#### Посмотрите в терминал **Consumer**, потребитель должен отобразить сообщения, которые мы написали в другом терминале.

###### 텍스트, 모니터, 스크린샷, 검은색이(가) 표시된 사진 자동 생성된 설명

### Создайте сообщения в пакетном режиме.

#### Из терминала **Producer** остановите производителя нажатием клавиш Ctrl+C

#### Отправьте содержимое файла Alice-in-Wonderlands.txt в тему topic\_logs

cat ~/Data/alice\_in\_wonderland.txt | \

kafka-console-producer \

--broker-list localhost:9092 \

--topic topic1\_logs

#### Что случилось? Все произошло очень быстро. Будем надеяться, вы не моргнули в этот момент. Все содержание книги «Алиса в стране чудес» было передано производителем в виде сообщений, а затем воспринято потребителем.

텍스트, 모니터, 스크린샷, 검은색이(가) 표시된 사진

자동 생성된 설명

### Проведите очистку

#### Остановите производителя и потребителя, нажав Ctrl+C в терминалах.

**Отправка сообщений из Flume в Kafka**

Как мы знаем, часто для создания производителей и потребителей Kafka требуется их программирование с использованием Producer и Consumer API Kafka. Другими словами, Kafka - это не инструмент конечного пользователя типа plug-n-play. Скорее, это требует от пользователей написания кода. Сегодня существует множество общедоступных производителей и потребителей, которые организации могут использовать, поэтому потребность в разработке резко сократилась.

В первые дни было предложено использовать Flume в сочетании с Kafka. Flume поддерживает большое количество источников и сливов. Идея состояла в том, чтобы разработать источник Kafka и приемник Kafka для Flume. Затем они могут быть объединены с существующими источниками и приемниками потока для завершения конвейера потока данных. Например, поток данных Flume, который собирает данные из веб-логов через источник NetCat, источник данных из буферного каталога или источник syslog, а в качестве слова можно назначить Kafka. Это фактически был бы производитель Kafka. Другой поток данных Flume с приемником Kafka для чтения собранных веб-логов и приемником HDFS для его хранения эффективно действовал бы как потребитель веб-логов Kafka.

В этой работе мы создадим производителя Kafka, используя Flume и слив Kafka внутри него. Он будет собирать данные из источника потоковой передачи, который помещает файл в буферный каталог. Мы удостоверимся, что производитель Kafka работает должным образом, прочитав раздел с потребителем консоли (Console consumer) Kafka.

## Настройте Apache Kafka

Как и в прошлый раз, чтобы снизить нагрузку на виртуальную машину, мы остановим HBase и запустим только Apache Kafka

### Остановим HBase (опять же, он может быть не запущен)

sudo stop-hbase.sh

### Перезапустим Kafka и Zookeeper

sudo systemctl stop kafka

sudo systemctl stop zookeeper

sudo systemctl start zookeeper

sudo systemctl status zookeeper

sudo systemctl start kafka

sudo systemctl status kafka

### Убедитесь, что zookeeper и kafka запущены. Если это не так, повторите предыдущий шаг.

###### 텍스트이(가) 표시된 사진 자동 생성된 설명

## Создайте конвейер Flume для сбора текста в потоковом режиме

### Перейдите в папку /home/student/Labs/C3U4 в терминале, а затем создайте новую папку и назовите ее spooldir:

cd ~/Labs/C3U4  
mkdir spooldir

### Создадим файл конфигурации агента spooldir.conf и заполним его.

#### Настроим SpoolDir в качестве источника:

#### Обратитесь к следующей инструкции для настройки источника из буферной директории (Spool Directory Source): <https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html#spooling-directory-source>

#### Установите в качестве spool-директории /home/student/Labs/C3U4/spool

#### Создадим 2 канала типа memory:

#### Обратитесь к следующей инструкции для настройки канала с памятью: <https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html#memory-channel>

#### Настройте емкость (capacity) на 10000

#### Настройте максимальное число событий на транзакцию (transactionCapacity) на 100

#### Создадим слив Kafka и Logger:

#### Обратитесь к следующей инструкции для настройки слива Kafka: <https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html#kafka-sink>

#### Установите тип слива на org.apache.flume.sink.kafka.KafkaSink

#### Установите тему Kafka в сливе на stream\_text

#### Установите список серверов начальной загрузки (bootstrap) на localhost:9092

#### Установите количество обрабатываемых сообщений в пакете (flumeBatchSize) на 5

#### Обратитесь к следующей инструкции для настройки Logger Sink <https://flume.apache.org/releases/content/1.9.0/FlumeUserGuide.html#logger-sink>

#### Установите тип на logger

#### Подсоединим источник к обоим каналам и к каждому каналу по сливу.

## Создайте консольного потребителя Kafka

### Откройте новый терминал и назовите его «Kafka Consumer», как мы делали это на шаге 3.2 предыдущей работы

### Запустите консольного потребителя Kafka

#### Выполните команду kafka-console-consumer со следующими параметрами:

#### Установите значение topic на stream\_text

#### Запустите потребителя так, чтобы он читал сначала (--from-beginning)

## Протестируйте конвейер данных

### Перейдите в /home/student/Labs/C3U4

### Создайте каталог и назовите его spool. Если каталог существует с предыдущих работ, полностью удалите каталог и создайте его заново. Чтобы быстро удалить содержимое и каталог, вы можете использовать следующую команду:

rm -rf ./spool

### Запустите производителя Flafka через агента Flume

#### Откройте новый терминал и измените имя терминала на «Kafka Producer», как мы сделали на шаге 3.2 предыдущей работы

#### Запустите поток данных Flume, что мы создали выше

flume-ng agent --conf $FLUME\_HOME/conf \

--conf-file /home/student/Labs/C3U4/spooldir.conf \

--name <имя\_агента> -Dflume.root.logger=INFO,console

### Запустите симулятор потокового источника

#### Откройте новый терминал и назовите его «Streaming Source»

#### Перейдите в /home/student/Labs/C3U4

#### Выполните программу spool\_stream.py (этот файл уже есть в папке C3U4) с помощью интерпретатора python. Следующая команда принимает 5000 символов из alice\_in\_wonderland.txt , создает временный файл для их хранения, а затем перемещает файл в каталог spool. Программа создает один файл каждые 5 секунд.

python ./spool\_stream.py ./spool 5000 ~/Data/alice\_in\_wonderland.txt

텍스트이(가) 표시된 사진

자동 생성된 설명

#### Выходные данные с трех терминалов должны выглядеть так, как указано выше. В терминале Kafka Producer регистратор отображает файл из каталога spool, который он отправляет в раздел Kafka. На экран Kafka Consumer будет выводиться каждые 5 секунд новое содержимое.

#### Примечание: Существует вероятность того, что Kafka Producer может выйти из строя во время работы. Это связано с тем, что сценарий перемещения и копирования занимает слишком много времени. Вы можете игнорировать это.

#### В терминале источника потоковой передачи нажмите Ctrl+C, чтобы остановить симулятор потоковой передачи.

#### Остановите производителя Flafka.

#### Остановите потребителя Kafka.