**ИНСТРУКЦИЯ ПО УСТАНОВКЕ**

**RT.STREAMING**

2021

ОГЛАВЛЕНИЕ

[1 УСТАНОВКА KAFKA 3](#_Toc83367715)

[1.1 Шаг 1. Загрузка Kafka 3](#_Toc83367716)

[Шаг 2. Настройка окружения Kafka 3](#_Toc83367717)

[1.2 Шаг 3. Создание топика для хранения событий 3](#_Toc83367718)

[1.3 Шаг 4. Запись событий в топик 4](#_Toc83367719)

[1.4 Шаг 5. Чтение событий 4](#_Toc83367720)

[1.5 Шаг 6. Импорт/экспорт данных в качестве стримов событий через Kafka Connect 5](#_Toc83367721)

[1.6 Шаг 7. Обработка событий через Kafka Streams 5](#_Toc83367722)

[1.7 Шаг 8. Завершение настройки 5](#_Toc83367723)

[2 УСТАНОВКА NIFI 7](#_Toc83367724)

[2.1 Системные требования 7](#_Toc83367725)

[2.2 Установка и запуск NiFi 7](#_Toc83367726)

[3 УСТАНОВКА AIRFLOW 9](#_Toc83367727)

[3.1 Системные требования 9](#_Toc83367728)

[3.2 Утилиты для установки 9](#_Toc83367729)

[3.3 Системные зависимости 9](#_Toc83367730)

[3.4 Скрипты установки 9](#_Toc83367731)

УСТАНОВКА KAFKA

Чтобы развернуть Kafka, выполните шаги, представленные ниже.

Шаг 1. Загрузка Kafka

Загрузите последнюю версию Kafka и распакуйте её:

|  |
| --- |
| $ wget {KAFKA\_URL}/kafka-2.7-2.x86\_64.rpm  $ sudo yum install kafka-2.7-2.x86\_64.rpm |

По умолчанию Kafka будет установлена в каталог /opt/kafka.

Шаг 2. Настройка окружения Kafka

|  |
| --- |
| Примечание. |
| В вашей локальной среде должна быть установлена Java 8+. |

Выполните следующие команды, чтобы запустить все службы в правильном порядке, предварительно перейдя в каталог с Kafka (по умолчанию /opt/kafka):

|  |
| --- |
| # Запустить сервис ZooKeeper  $ bin/zookeeper-server-start.sh config/zookeeper.properties |

Откройте ещё один терминальный сеанс и запустите:

|  |
| --- |
| # Запустить брокерскую службу Kafka  $ bin/kafka-server-start.sh config/server.properties |

После успешного запуска всех служб у вас будет запущена и готова к использованию базовая среда Kafka.

Шаг 3. Создание топика для хранения событий

Kafka — это платформа распределённой потоковой передачи событий, которая позволяет читать, записывать, хранить и обрабатывать события (также называемые записями или сообщениями) на многих машинах.

Примерами событий являются платёжные транзакции, обновления геолокации с мобильных телефонов, заказы на доставку, измерения датчиков с устройств IoT или медицинского оборудования и многое другое. Эти события организованы и хранятся в топиках. Очень упрощённо, топик похож на папку в файловой системе, а события — это файлы в этой папке.

Итак, прежде чем вы сможете записать свои первые события, вы должны создать топик. Откройте ещё один терминальный сеанс и запустите:

|  |
| --- |
| $ bin/kafka-topics.sh --create --topic quickstart-events --bootstrap-server localhost:9092 |

Все инструменты командной строки Kafka имеют до­­полнительные параметры: запустите команду kafka-topics.sh без каких-либо аргументов, чтобы отобразить информацию об использовании команды. Например, он также может показать вам подробную информацию, такую как количество партиций нового топика:

|  |
| --- |
| $ bin/kafka-topics.sh --describe --topic quickstart-events --bootstrap-server localhost:9092  Topic:quickstart-events PartitionCount:1 ReplicationFactor:1 Configs:  Topic: quickstart-events Partition: 0 Leader: 0 Replicas: 0 Isr: 0 |

Шаг 4. Запись событий в топик

Клиент Kafka связывается с брокерами Kafka через сеть для записи (или чтения) событий. После получения брокеры будут хранить события в надёжном и отказоустойчивом виде столько времени, сколько необходимо — даже навсегда.

Запустите консоль клиента поставщика, чтобы записать несколько событий в созданный топик. По умолчанию каждая вводимая строка приводит к записи отдельного события в топик.

|  |
| --- |
| $ bin/kafka-console-producer.sh --topic quickstart-events --bootstrap-server localhost:9092  This is my first event  This is my second event |

Вы можете остановить клиента производителя с помощью Ctrl-C в любое время.

Шаг 5. Чтение событий

Откройте другой терминальный сеанс и запустите консоль клиента потребителя, чтобы прочитать только что созданные события:

|  |
| --- |
| $ bin/kafka-console-consumer.sh --topic quickstart-events --from-beginning --bootstrap-server localhost:9092  This is my first event  This is my second event |

Вы можете остановить клиента потребителя с помощью Ctrl-C в любое время.

Не стесняйтесь экспериментировать: например, переключитесь обратно на свой терминал поставщика (предыдущий шаг), чтобы записать дополнительные события, и посмотрите, как события сразу же отображаются в терминале потребителя.

Поскольку события надёжно хранятся в Kafka, они могут быть прочитаны сколько угодно раз и любым количеством потребителей. В этом легко убедиться, открыв ещё один сеанс терминала и повторно запустив предыдущую команду.

Шаг 6. Импорт/экспорт данных в качестве стримов событий через Kafka Connect

Вероятно, у вас есть много данных в существующих системах, таких как реляционные базы данных или традиционные системы обмена сообщениями, а также множество приложений, которые уже используют эти системы. Kafka Connect позволяет непрерывно получать данные из внешних систем в Kafka и наоборот. Таким образом, очень легко интегрировать существующие системы с Kafka. Чтобы сделать этот процесс ещё проще, доступны сотни таких коннекторов.

Шаг 7. Обработка событий через Kafka Streams

Как только данные будут сохранены в Kafka в качестве событий, вы можете обрабатывать их с помощью клиентской библиотеки Kafka Streams для Java/Scala. Она позволяет имплементировать критически важные приложения и микросервисы реального времени, где входные и/или выходные данные хранятся в топиках Kafka. Kafka Streams сочетает в себе простоту написания и развёртывания стандартных приложений Java и Scala на стороне клиента с преимуществами серверной кластерной технологии Kafka, что делает эти приложения высокомасштабируемыми, эластичными, отказоустойчивыми и распределёнными. Библиотека поддерживает однократную обработку, операции и агрегаты с отслеживанием статусов, управление окнами, объединения, обработку на основе времени события и многое другое.

Ниже в примере представлена имплементация популярного алгоритма WordCount:

|  |
| --- |
| KStream<String, String> textLines = builder.stream("quickstart-events");  KTable<String, Long> wordCounts = textLines  .flatMapValues(line -> Arrays.asList(line.toLowerCase().split(" ")))  .groupBy((keyIgnored, word) -> word)  .count();  wordCounts.toStream().to("output-topic", Produced.with(Serdes.String(), Serdes.Long())); |

Шаг 8. Завершение настройки

Выполните следующие шаги:

1. Остановите поставщиков и клиентов потребителей с помощью Ctrl-C, если вы ещё этого не сделали.

Остановите брокера Kafka с помощью Ctrl-C.

Наконец, остановите сервер ZooKeeper с помощью Ctrl-C.

Если вы также хотите удалить любые данные вашей локальной среды Kafka, включая любые события, которые вы создали в процессе, выполните команду:

|  |
| --- |
| $ rm -rf /tmp/kafka-logs /tmp/zookeeper |

УСТАНОВКА NIFI

Системные требования

NiFi может работать на таком простом устройстве, как ноутбук, но его также можно кластеризовать на многих серверах корпоративного класса. Следовательно, количество необходимого оборудования и памяти будет зависеть от размера и характера задействованного потока данных. Данные хранятся на диске, пока NiFi их обрабатывает. Таким образом, NiFi необходимо выделить достаточное дисковое пространство для различных репозиториев, в частности, content-репозитория, flowfile-репозитория и provenance-репозитория. NiFi имеет следующие минимальные системные требования:

1. Требуется Java 8 или Java 11.

Поддерживаемые операционные системы:

Linux;

Unix;

Windows;

macOS.

Поддерживаемые веб-браузеры:

Microsoft Edge: текущий и (текущий - 1);

Mozilla FireFox: текущий и (текущий - 1);

Google Chrome: текущий и (текущий - 1);

Safari: текущий и (текущий - 1).

Установка и запуск NiFi

1. Для РЕД ОС:

Скачайте файл с дистрибутивом;

Выполните:

|  |
| --- |
| sudo yum install ApacheNiFi-1.1.12-1.el7\_1.x86\_64.rpm |

Внесите необходимые изменения в файлы, находящиеся в каталоге <installdir>/conf.

Как минимум, рекомендуется отредактировать файл nifi.properties и ввести пароль для nifi.sensitive.props.key, а также установить порт для web-интерфейса nifi.web.http.port и nifi.web.https.port;

В каталоге <installdir>/bin выполните следующие команды, набрав ./nifi.sh <command>:

* start — запускает NiFi в фоновом режиме;
* stop — останавливает NiFi, работающий в фоновом
* status — показывает текущий статус NiFi;
* run — запускает NiFi на переднем плане и ждёт, пока Ctrl-C инициирует завершение работы NiFi;
* install — устанавливает NiFi в качестве сервиса, которым затем можно управлять через:

service nifi start;

service nifi stop;

service nifi status.

Для Windows:

Распакуйте в желаемый каталог установки;

Внесите необходимые изменения в файлы, находящиеся в каталоге <installdir>/conf.

Как минимум, рекомендуется отредактировать файл nifi.properties и ввести пароль для nifi.sensitive.props.key;

Перейдите в каталог <installdir>/bin;

Дважды щёлкните run-nifi.bat. Это запустит NiFi на переднем плане и будет ждать, пока Ctrl-C инициирует выключение NiFi;

Чтобы увидеть текущий статус NiFi, дважды щёлкните status-nifi.bat;

При первом запуске NiFi создаются следующие файлы и каталоги:

content\_repository;

database\_repository;

flowfile\_repository;

originance\_repository;

каталог work;

каталог logs;

в каталоге conf создаётся файл flow.xml.gz.

УСТАНОВКА AIRFLOW

В данном разделе описывается установка с использованием пакета apache-airflow, опубликованного в PyPI, но некоторая информация может быть полезна и во время установки с другими инструментами.

Системные требования

Airflow протестирован с помощью:

Python: 3.6, 3.7, 3.8;

Базы данных:

PostgreSQL: 9.6, 10, 11, 12, 13;

MySQL: 5.7, 8;

SQLite: 3.15.0+;

Kubernetes: 1.18.15 1.19.7 1.20.2.

Утилиты для установки

Для установки потребуется [Docker Compose](https://docs.docker.com/compose/install/) v1.29.1 и Docker CE.

Системные зависимости

Это полностью готовый образ docker.

Скрипты установки

Для разворачивания экземпляра Airflow потребуется:

1. Создать каталог для Airflow, например, airflow.
2. В каталоге в консоли выполнить:

|  |
| --- |
| mkdir -p ./dags ./logs ./plugins  echo -e "AIRFLOW\_UID=$(id -u)\nAIRFLOW\_GID=0" > .env |

1. Подключиться к docker registry repo.data.rt.ru/demo-registry, используя логин и пароль и команду docker login.
2. Скачать файл docker-compose.yaml c (URL) в каталог из п. 1.
3. Выполнить в каталоге из п. 1:

|  |
| --- |
| docker-compose up airflow-init |

1. Выполнить для запуска Airflow:

|  |
| --- |
| docker-compose up |

Web-интерфейс будет доступен по адресу http://{host-ip}:8080/.

Пользователь и пароль по умолчанию — airflow.