

Лабораторная работа 5.1. Развертывание и настройка кластера Hadoop

Цель: ознакомление с процессом установки и настройки распределенных систем, таких как Apache(Arenadata) Hadoop. Изучить основные операции и функциональные возможности системы, что позволит понять принципы работы с данными и распределенными вычислениями. Необходимое ПО:

- Ubuntu 24.04 LTS (22.04, 20.04) или новее.
- Java 8 ил Java11 или новее.
- Apache Spark 3.4.3.
- Python 3.12+.
- pip (менеджер пакетов Python).

Алгоритм выполнения задания в Apache Hadoop

В виртуальной машине Шаг 1-8 пропустить.

Шаг 1. Установка необходимых компонентов.

```
```bash
```

```
sudo apt update
sudo apt install ssh pdsh -y
```
```

Шаг 2. Создание пользователя Hadoop.

```
```bash
```

```
sudo adduser hadoop
sudo usermod -aG sudo hadoop
su - hadoop
```
```

Шаг 3. Настройка SSH.

```
```bash
```

```
ssh-keygen -t rsa -P ""
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
chmod 0600 ~/.ssh/authorized_keys
```
```

Шаг 4. Загрузка и установка Hadoop.

```
```bash
```

```
wget https://downloads.apache.org/hadoop/common/hadoop-3.3.5/hadoop-3.3.5.tar.gz
tar -xvzf hadoop-3.3.5.tar.gz
sudo mv hadoop-3.3.5 /usr/local/hadoop
```
```

Шаг 5. Настройка окружения Hadoop.

Добавьте следующие строки в конец файла ~/.bashrc:

```
```bash
```

```
export HADOOP_HOME=/usr/local/hadoop
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
```

```
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```
```

Применить изменения:

```
```bash
source ~/.bashrc
```
```

Шаг 6. Настройка конфигурационных файлов Hadoop.

а) Отредактируйте \$HADOOP_HOME/etc/hadoop/hadoop-env.sh:

```
```bash
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
```
```

б) Отредактируйте \$HADOOP_HOME/etc/hadoop/core-site.xml:

```
```xml
<configuration>
 <property>
 <name>fs.defaultFS</name>
 <value>hdfs://localhost:9000</value>
 </property>
</configuration>
```
```

в) Отредактируйте \$HADOOP_HOME/etc/hadoop/hdfs-site.xml:

```
```xml
<configuration>
 <property>
 <name>dfs.replication</name>
 <value>1</value>
 </property>
 <property>
 <name>dfs.namenode.name.dir</name>
 <value>/home/hadoop/hdfs/namenode</value>
 </property>
 <property>
 <name>dfs.datanode.data.dir</name>
 <value>/home/hadoop/hdfs/datanode</value>
 </property>
</configuration>
```
```

г) Отредактируйте \$HADOOP_HOME/etc/hadoop/mapred-site.xml:

```
```xml
<configuration>
 <property>
 <name>mapreduce.framework.name</name>
 <value>yarn</value>
 </property>
</configuration>
```
```

е) Отредактируйте \$HADOOP_HOME/etc/hadoop/yarn-site.xml:

```
```xml
<configuration>
 <property>
 <name>yarn.nodemanager.aux-services</name>
 <value>mapreduce_shuffle</value>
 </property>
</configuration>
```
```

Шаг 7. Создание директорий для HDFS.

```
```bash
mkdir -p ~/hdfs/namenode ~/hdfs/datanode
```
```

Шаг 8. Форматирование HDFS.

```
```bash
hdfs namenode -format
```
```

Все дальнейшие действия выполняются пользователем **hadoop**.

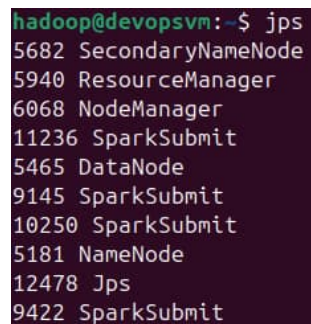
```
```bash
sudo su - hadoop
```
```

Шаг 9. Запуск Hadoop.

```
```bash
start-dfs.sh
start-yarn.sh
```
```

Шаг 10. Проверка работы Hadoop.

```
```bash
jps
```
```



```
hadoop@devopsvm:~$ jps
5682 SecondaryNameNode
5940 ResourceManager
6068 NodeManager
11236 SparkSubmit
5465 DataNode
9145 SparkSubmit
10250 SparkSubmit
5181 NameNode
12478 Jps
9422 SparkSubmit
```

Вы должны увидеть следующие процессы: NameNode, DataNode, SecondaryNameNode, ResourceManager, NodeManager.

В стандартной конфигурации Hadoop HDFS предоставляет веб-интерфейс, доступный через веб-браузер на порту 9870. Этот интерфейс позволяет просматривать состояние и структуру HDFS, а также выполнять некоторые операции.

Чтобы получить доступ к веб-интерфейсу HDFS, выполните следующие шаги:

- Убедитесь, что Hadoop (в частности, HDFS) запущен.
- Откройте веб-браузер на компьютере, с которого у вас есть сетевой доступ к серверу Hadoop.
- В адресной строке браузера введите:

```

**http://localhost:9870**

...

localhost:9870/dfshealth.html#tab-overview



## Overview 'localhost:9000' (✓active)

Started:	Mon Aug 26 13:07:51 +0300 2024
Version:	3.3.5, r706d88266abcee09ed78fbaa0ad5f74d818ab0e9
Compiled:	Wed Mar 15 18:56:00 +0300 2023 by stevel from branch-3.3.5
Cluster ID:	CID-60a52b68-6139-4947-8731-3c039547a32e
Block Pool ID:	BP-1830111676-127.0.1.1-1724666841903

- Если обращаетесь к Hadoop с другого компьютера, замените "localhost" на IP-адрес или имя хоста сервера, на котором запущен Hadoop.
- Нажмите Enter, должны увидеть веб-интерфейс HDFS.

Через этот веб-интерфейс вы сможете просматривать структуру директорий HDFS, проверять состояние и здоровье узлов, просматривать логи и выполнять другие административные задачи.

На веб-интерфейсе YARN вы сможете увидеть информацию о запущенных приложениях, статусе узлов и других метриках кластера. Это удобный способ управления и мониторинга задач.

**http://localhost:8088**

Шаг 11. Работа с экономическими данными

а) Создайте директорию в HDFS:

```
```bash
```

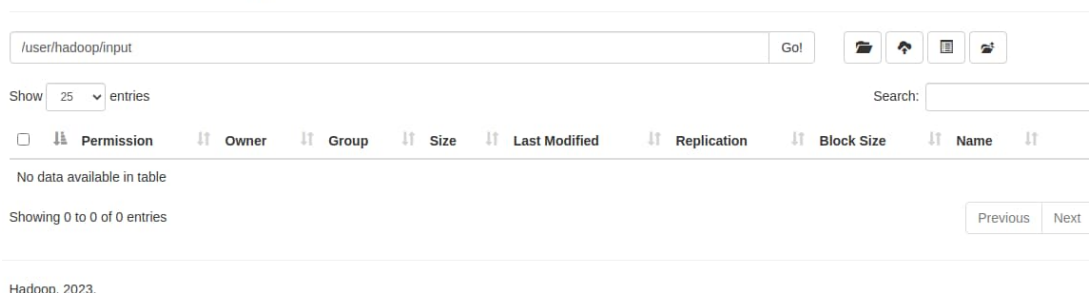
```
hdfs dfs -mkdir /user
```

```
hdfs dfs -mkdir /user/hadoop
```

```
hdfs dfs -mkdir /user/hadoop/input
```

```
```
```

## Browse Directory



Проведем расчет экономических показателей на примере открытых экономических данных. Будем использовать данные о ВВП стран мира от Всемирного банка.

Шаг 12. Подготовка данных.

Скачайте данные о ВВП стран мира

([https://raw.githubusercontent.com/BosenkoTM/Distributed\\_systems/main/practice/2024/lw\\_01/GDP.csv](https://raw.githubusercontent.com/BosenkoTM/Distributed_systems/main/practice/2024/lw_01/GDP.csv)).

[https://github.com/BosenkoTM/Distributed\\_systems/tree/main/practice/2024/lw\\_01](https://github.com/BosenkoTM/Distributed_systems/tree/main/practice/2024/lw_01)

Сохраните файл как **GDP.csv**.

Шаг 13. Загрузка данных в HDFS.

```
```bash
```

```
wget
```

```
https://raw.githubusercontent.com/BosenkoTM/Distributed_systems/main/practice/2024/lw_01/GDP.csv
```

```
hdfs dfs -mkdir /user/hadoop/economic_data
```

```
hdfs dfs -put GDP.csv /user/hadoop/economic_data/
```

```
```
```

**предоставить другим пользователям доступ на изменения данных**

```
hdfs dfs -chmod 777 /user/hadoop/economic_data
```

Обработка данных с помощью MapReduce или Spark

Шаг 14. Запустите Spark.

```
```bash
```

```
spark-shell
```

```
```
```

Шаг 15. Загрузите данные и выполните расчеты.

```
```scala
```

```
val data = spark.read.option("header", "true").csv("file:///home/hadoop/GDP.csv")
```

Проверьте правильность названия столбца

Вывести схему DataFrame для проверки:

```
```scala
```

```
data.printSchema()
```

```
root
|-- Country: string (nullable = true)
|-- Year: string (nullable = true)
|-- GDP: string (nullable = true)
|-- Urban_population: string (nullable = true)
|-- Industry: string (nullable = true)
|-- Business: string (nullable = true)
|-- Mining: string (nullable = true)
|-- Manufacturing: string (nullable = true)
|-- Electricity_supply: string (nullable = true)
|-- Water_supply: string (nullable = true)
|-- Construction: string (nullable = true)
|-- Retail_trade: string (nullable = true)
|-- Transportation: string (nullable = true)
|-- Accommodation: string (nullable = true)
|-- Information: string (nullable = true)
|-- Financial: string (nullable = true)
|-- Real estate : string (nullable = true)
|-- Professional_scientific: string (nullable = true)
|-- Administrative: string (nullable = true)
|-- Education: string (nullable = true)
|-- Human_health: string (nullable = true)
|-- Arts: string (nullable = true)
|-- Other: string (nullable = true)
```

Если в схеме видите, что правильное название столбца, например, ``GDP``, замените GDR на правильное имя:

```
val result = data.selectExpr("avg(GDP) as avg_GDR")
```

Вычисляем среднее значение GDR

```
val result = data.selectExpr("avg(GDR) as avg_GDR")
```

```
scala> val data = spark.read.option("header", "true").csv("file:///home/hadoop/GDP.csv")
data: org.apache.spark.sql.DataFrame = [Country: string, Year: string ... 21 more fields]

scala> val result = data.selectExpr("avg(GDP) as avg_GDR")
result: org.apache.spark.sql.DataFrame = [avg_GDR: double]
```

// Сохраняем результат в CSV файл

```
result.write.option("header", "true").csv("/home/hadoop/output/avg_GDR.csv")
```

Выходим из Scala.

```
```scala
```

```
:q
```

В Ubuntu 24.04 Scala сохраняет результаты в файле **part-00000-*.csv**, каталог будет определен последним адресом в пути при сохранении, то есть **avg_GDR.csv**.

Шаг 16. Переименовать полученный результат **part-00000-*.csv** в Ubuntu **avg.csv**

```
```bash
```

```
mv part-00000-*.csv avg.csv
```

```
```
```

Шаг 17. **Переносим данные в HDFS.** Загрузите экономические данные в HDFS:

```
```bash
```

```
hdfs dfs -put /home/hadoop/output/avg.csv /user/hadoop/input/
```

```
```
```

Проверьте, что данные загружены.

```
```bash
```

```
hdfs dfs -ls /user/hadoop/input/
```

```
```
```

/user/hadoop/input

Go!

Show

25

entries

Search:

| <input type="checkbox"/> | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name | |
|--------------------------|------------|--------|------------|------|---------------|-------------|------------|---------|--|
| <input type="checkbox"/> | -rw-r--r-- | hadoop | supergroup | 27 B | Aug 26 15:42 | 1 | 128 MB | avg.csv | |

Чтобы остановить Hadoop 3 в Ubuntu, выполните следующие шаги.

Сначала остановите YARN (если он запущен).

```
```bash
```

```
stop-yarn.sh
```

Затем остановите HDFS.

```
```bash
```

```
stop-dfs.sh
```

Если вы используете MapReduce JobHistory Server, остановите его:

```
mr-jobhistory-daemon.sh stop historyserver.
```

Для полной остановки всех Hadoop-демонов можно использовать команду.

```
```bash
```

```
stop-all.sh
```

Проверьте, что все процессы Hadoop остановлены.

```
```bash
```

```
jps
```

Эта команда покажет список запущенных Java-процессов. Убедитесь, что в списке нет процессов, связанных с Hadoop (например, NameNode, DataNode, ResourceManager и т.д.).

Если какие-то процессы остались, вы можете остановить их вручную с помощью команды kill.

```
```bash
```

```
kill -9 <PID>
```

где <PID> - идентификатор процесса, который вы хотите остановить.

```
ghdthbv
```

```
hdfs getconf -confKey fs.defaultFS
```

### **Задание для самостоятельной работы**

1. Загрузите данные по акциям другой компании (например, Microsoft - MSFT).
2. Выполните аналогичный анализ для новых данных.
3. Сравните результаты анализа двух компаний.
4. Напишите Spark-приложение, которое находит дни с максимальным объемом торгов для обеих компаний.

### **Отчет**

Студенты должны подготовить отчет, включающий:

1. Описание процесса установки и настройки Hadoop.
2. Листинг выполненных команд и их результаты.
3. Анализ полученных результатов.
4. Код и результаты выполнения задания для самостоятельной работы.
5. Выводы о функциональности и возможностях Hadoop для анализа экономических данных.

### **Постановка задачи**

Проанализировать экономические данные, содержащиеся в вашем файле, который находится в файловой системе Hadoop (HDFS). Задача заключается в извлечении, обработке, и анализе данных с целью выявления закономерностей, тенденций, и создания визуализаций на основе предоставленных данных.

Действия, которые требуется выполнить:

#### **1. Подключение к Hadoop и загрузка данных.**

- Подключиться к HDFS и убедиться, что файл доступен по пути `hdfs://localhost:9000/user4/hadoop/economic_data/BAII_ФАЙЛ.csv`
- Использовать PySpark или Pandas для загрузки данных из HDFS в DataFrame, который можно будет использовать для анализа.

#### **2. Исследование и очистка данных.**

- Проверить структуру данных и типы столбцов (например, с помощью `printSchema()` для PySpark или `describe()` для Pandas).
- Убедиться, что все данные корректны, и преобразовать необходимые столбцы в числовые форматы, если они изначально представлены в виде строк.
- Проверить данные на наличие пропущенных или некорректных значений, удалить или заполнить такие значения в зависимости от ситуации.

#### **3. Анализ данных.**

- Провести базовый статистический анализ данных:

- Вычислить средние значения, медианы, минимумы и максимумы для экономических параметров.

- Проанализировать и выявить тенденции.

- Построить временные ряды, чтобы понять, как изменялась их экономика с течением времени.

#### **4. Визуализация данных.**

- Построить графики (например, графики временных рядов).

- Построить диаграммы для сравнения экономических показателей.

#### **5. Сохранение и экспорт результатов.**

- Сохранить результаты анализа и визуализации в формате CSV или изображений.

- Сохранить обработанные данные (например, данные только для отдельных стран) обратно в HDFS, чтобы другие команды могли использовать их для дальнейшего анализа.

- Создать отчет, включающий ключевые выводы и визуализации, для представления результатов анализа заинтересованным сторонам.

#### **6. Автоматизация процесса (опционально).**

- Создать скрипт или Jupyter Notebook, который автоматизирует процесс загрузки, анализа и визуализации данных для упрощения дальнейших исследований и повторного использования кода.

#### **Ожидаемый результат.**

В результате выполнения этих действий будут получены обработанные и проанализированные данные. Визуализации и отчет позволят наглядно представить данные и выводы, выявить тенденции и взаимосвязи.

#### **Варианты заданий**

Вариант выбирается согласно номеру студента в списке группы:

1. Установка Apache Hadoop на одном узле и выполнение простой задачи на подсчет строк в файле.

Данные: Исторические данные по акциям Сбербанка (SBER) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет средней цены закрытия, группировка по месяцам.

2. Установка Apache(Arenadata) Hadoop и выполнение задачи на копирование файлов в HDFS.

Данные: Исторические данные по акциям Газпрома (GAZP) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2019 год, расчет максимальной цены открытия, группировка по кварталам.

3. Установка Apache Hadoop и выполнение задачи на сортировку данных.

Данные: Исторические данные по акциям Лукойла (LKOH) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, группировка по годам.

4. Настройка Apache Hadoop.

Данные: Исторические данные по акциям Яндекса (YNDX) с сайта Московской биржи (moex.com)



Операции: Фильтрация данных за 2021 год, расчет средней цены закрытия, тренд анализа.

5. Настройка кластерного режима для Apache(Arenadata) Hadoop на 2 узлах и проверка работоспособности.

Данные: Исторические данные по акциям Роснефти (ROSN) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, группировка по месяцам.

6. Установка Установка Apache Hadoop и выполнение задачи на агрегацию данных.

Данные: Исторические данные по акциям Норильского никеля (GMKN) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены закрытия, группировка по кварталам.

7. Установка и настройка Установка Apache Hadoop для работы с внешним источником данных (например, S3, MySQL).

Данные: Исторические данные по акциям ВТБ (VTBR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 10 лет, расчет коэффициента вариации цены закрытия, тренд

8. Установка Apache(Arenadata) Hadoop и выполнение задачи на объединение файлов в HDFS.

Данные: Исторические данные по акциям Магнита (MGNT) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, группировка по годам.

9. Установка и настройка Apache Hadoop.

Данные: Исторические данные по акциям Полюса (PLZL) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2019 год, расчет средней цены закрытия, корреляция с объемом торгов.

10. Настройка Apache Hadoop.

Данные: Исторические данные по акциям МТС (MTSS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 2 года, расчет максимальной цены закрытия, тренд анализа.

11. Установка Apache(Arenadata) Hadoop и выполнение задачи на создание и удаление каталогов в HDFS.

Данные: Исторические данные по акциям Татнефти (TATN) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, группировка по месяцам.

12. Установка Apache Hadoop.

Данные: Исторические данные по акциям Сургутнефтегаз (SNGS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены открытия, тренд анализа.

13. Установка Apache Hadoop.

Данные: Исторические данные по акциям Мечела (MTLR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, группировка по кварталам.

14. Установка Apache(Arenadata) Hadoop и выполнение задачи на распределение файлов между узлами.

Данные: Исторические данные по акциям Интер РАО (IRAO) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, корреляция с объемом торгов.

15. Установка Apache Hadoop и выполнение задачи на анализ текстовых данных.

Данные: Исторические данные по акциям Аэрофлота (AFLT) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 2 года, расчет максимальной цены закрытия, тренд анализа.

16. Установка Apache Hadoop.

Данные: Исторические данные по акциям Системы (AFKS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2019 год, расчет средней цены закрытия, группировка по месяцам.

17. Установка Apache(Arenadata) Hadoop и выполнение задачи на создание и просмотр логов системы.

Данные: Исторические данные по акциям ФосАгро (PHOR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 5 лет, расчет минимальной цены закрытия, тренд анализа.

18. Установка Apache Hadoop и выполнение задачи на фильтрацию данных.

Данные: Исторические данные по акциям Алросы (ALRS) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2020 год, расчет стандартного отклонения цены закрытия, группировка

19. Установка Apache Hadoop и выполнение задачи на распределенную обработку данных.

Данные: Исторические данные по акциям Русала (RUAL) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 3 года, расчет медианной цены закрытия, корреляция с объемом торгов

20. Установка Apache Hadoop выполнение задачи на работу с JSON-файлами.

Данные: Исторические данные по акциям Мосбиржи (MOEX) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2018 год, расчет средней цены открытия, группировка по годам.

21. Установка Apache Hadoop.

Данные: Исторические данные по акциям РУСГИДРО (HYDR) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 10 лет, расчет коэффициента вариации цены закрытия, тренд анализа.

22. Установка Apache(Arenadata) Hadoop и выполнение задачи на создание резервной копии данных.

Данные: Исторические данные по акциям Россетей (RSTI) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2021 год, расчет максимальной цены закрытия, корреляция с объемом торгов.

23. Установка Apache Hadoop и выполнение задачи на вычисление статистических параметров данных.

Данные: Исторические данные по акциям X5 Retail Group (FIVE) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 2 года, расчет средней цены закрытия, группировка по месяцам.

24. Установка Apache Hadoop.

Данные: Исторические данные по акциям ТМК (TRMK) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за 2019 год, расчет минимальной цены закрытия, тренд анализа.

25. Установка Apache(Arenadata) Hadoop и выполнение задачи на слияние данных из нескольких источников в HDFS.

Данные: Исторические данные по акциям М.Видео (MVID) с сайта Московской биржи (moex.com)

Операции: Фильтрация данных за последние 5 лет, расчет стандартного отклонения цены закрытия, группировка по кварталам.