



Министерство науки и высшего образования Российской Федерации
Калужский филиал федерального государственного автономного
образовательного учреждения высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

ЛАБОРАТОРНАЯ РАБОТА

«ОСНОВЫ HADOOP. УСТАНОВКА HADOOP. ОСНОВНЫЕ КОМАНДЫ ФАЙЛОВОЙ СИСТЕМЫ HDFS»

по дисциплине: «Технологии обработки больших данных»

Выполнил: студент группы ИУК4-72Б

(Подпись)

Моряков В.Ю.

(И.О. Фамилия)

Проверил:

(Подпись)

Голубева С.Е.

(И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Цель: формирование практических навыков по установке и настройке кластера Hadoop и работе с файловой системой HDFS.

Задачи:

- 1. Изучить основы Hadoop.
- 2. Научиться устанавливать и конфигурировать Hadoop.
- 3. Изучить основные команды для работы с файловой системой HDFS.
- 4. Получить навыки написания программ для работы с HDFS.

Формулировка задания (17 вариант):

Для всех вариантов настроить кластер Hadoop, состоящий из двух серверов, изучить команды HDFS для работы с файлами и выполнить следующие задания:

- 1. Проверить существует ли директория /user/hduser в HDFS, если нет, то создать. Создать директорию /user/hduser/Hadoop
- 2. Создать файл в директории /user/hduser/hadoop, название файла – ваше имя и группа. После создания файла, все, что вы вводите в консоль должно сохраниться в файле. Ввести несколько строк и сохранить.
- 3. Убедиться в существовании файла через web-интерфейс.
- 4. Перенести файл в локальную файловую систему.
- 5. Создать новый текстовый файл в локальной файловой системе. Перенести файл в HDFS. Убедиться в существовании файла через web-интерфейс.
- 6. Просмотреть права доступа на файл. Изменить права доступа к файлу, чтобы только владелец и члены группы имели полный контроль над файлом. Н
- 7. Напишите программу, которая будет сравнивать содержимое двух текстовых файлов в HDFS.

Программа:

- 1. Напишите программу, которая будет сравнивать содержимое двух текстовых файлов в HDFS.

Ход выполнения:

```
WARN[0000] /home/hironoz/BMSTU_FINISH_LINE/docker-hadoop/lab1/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid
potential confusion
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "HADOOP_HOME" variable is not set. Defaulting to a blank string.
[+] Running 4/4
✔ Container lab1-nodemanager-1 Started 0.4s
✔ Container lab1-namenode-1 Started 0.5s
✔ Container lab1-datanode-1 Started 0.4s
✔ Container lab1-resourcemanager-1 Started 0.4s
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
a7d3c991d9c9   apache/hadoop:3 "/usr/local/bin/dumb..." Less than a second ago Up Less than a second 0.0.0.0:9870->9870/tcp, [::]:9870->9870/tcp lab1-namenod
e-1
7bcc9fd9f97db   apache/hadoop:3 "/usr/local/bin/dumb..." Less than a second ago Up Less than a second 0.0.0.0:8088->8088/tcp, [::]:8088->8088/tcp lab1-resourc
emanager-1
a8f6792d7e53   apache/hadoop:3 "/usr/local/bin/dumb..." Less than a second ago Up Less than a second                               lab1-datanod
e-1
3dac64c1d8b3   apache/hadoop:3 "/usr/local/bin/dumb..." Less than a second ago Up Less than a second                               lab1-nodeman
ager-1
bash-4.2$
```

Рисунок 1 Установка hadoop

```
ager-1
bash-4.2$ bash /run_it_docker.sh
Creating HDFS directories...
```

Рисунок 2 Запуск скрипта

```
HDFS automation completed!
=== Создание тестовых файлов ===
Созданы файлы:
-rw-r--r-- 1 hadoop users 30 Oct  8 05:39 /tmp/file1.txt
-rw-r--r-- 1 hadoop users 30 Oct  8 05:39 /tmp/file2.txt
🔍 Сравнение файлов:
  1 /tmp/file1.txt
  2 /tmp/file2.txt
⚠️ Файлы различаются!

♦ Строка 2 отличается:
File1: This is file 1.
File2: This is file 2.
```

Рисунок 3 Результаты работы python программы

```
lab1/run_it_docker.sh X create_files.sh X init-hdfs.sh X mapper_avg_wordt
1 #!/bin/bash
2
3 sleep 1
4
5 echo "Creating HDFS directories..."
6 hdfs dfs -mkdir -p /user/hduser/Hadoop
7
8 FILE_NAME="MoryakovVY_IUK472B.txt"
9 echo -e "Hello!\nThis is a test file.\nHDFS automation with Docker." > /tmp/$FILE_NAME
10 hdfs dfs -put -f /tmp/$FILE_NAME /user/hduser/Hadoop/
11
12 LOCAL_FILE="new_file.txt"
13 echo -e "This is a new file\nContents of the new file" > /tmp/$LOCAL_FILE
14
15 hdfs dfs -put -f /tmp/$LOCAL_FILE /user/hduser/Hadoop/
16
17 hdfs dfs -chmod 770 /user/hduser/Hadoop/$FILE_NAME
18 hdfs dfs -chmod 770 /user/hduser/Hadoop/$LOCAL_FILE
19
20 echo "!!!INPUT!!!\n"
21 hdfs dfs -put -f - /user/hduser/Hadoop/$FILE_NAME
22
23 echo "File created. File content:\n"
24 hdfs dfs -cat /user/hduser/Hadoop/$FILE_NAME
25 hdfs dfs -ls /user/hduser/Hadoop/
26
27 echo "HDFS state after automation:"
28 hdfs dfs -ls -R /user/hduser
29
30 echo "HDFS automation completed!"
```

Рисунок 4 Скрипт для работы с HDFS

```

Creating HDFS directories...
!!!INPUT!!!\n
This is my custom input
File created. File content:\n
This is my custom input
Found 2 items
-rw-r--r--    1 hadoop supergroup          24 2025-10-09 07:50 /user/hduser/Hadoop
/MoryakovVY_IUK472B.txt
-rwxrwx---    1 hadoop supergroup          44 2025-10-09 07:50 /user/hduser/Hadoop
/new_file.txt
HDFS state after automation:
drwxr-xr-x    - hadoop supergroup           0 2025-10-09 07:50 /user/hduser/Hadoop
-rw-r--r--    1 hadoop supergroup          24 2025-10-09 07:50 /user/hduser/Hadoop
/MoryakovVY_IUK472B.txt
-rwxrwx---    1 hadoop supergroup          44 2025-10-09 07:50 /user/hduser/Hadoop
/new_file.txt
HDFS automation completed!
=== Создание тестовых файлов ===
put: `/tmp/file2.txt': No such file or directory
Созданы файлы:
-rw-r--r--    1 hadoop users 30 Oct  9 07:50 /tmp/_file2.txt
-rw-r--r--    1 hadoop users 30 Oct  9 07:50 /tmp/file1.txt

```

Рисунок 5 Результат работы hdfs-скрипта

Browse Directory

Show entries

Search:

| <input type="checkbox"/> | Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name | |
|--------------------------|----------------------------|------------------------|----------------------------|------|---------------|-------------------|------------|--|--|
| <input type="checkbox"/> | -rw-r--r-- | hadoop | supergroup | 6 B | Oct 08 10:31 | 1 | 128 MB | MoryakovVY_IUK472B.txt | |
| <input type="checkbox"/> | -rwxrwx--- | hadoop | supergroup | 44 B | Oct 08 10:31 | 1 | 128 MB | new_file.txt | |

Showing 1 to 2 of 2 entries

Hadoop, 2023.

Рисунок 6 Результаты работы в графическом интерфейсе

Листинги программ

compare_hdfs_files.py

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-

```

```

import subprocess
import sys

```

```

def read_hdfs_file(path):
    try:
        output = subprocess.check_output(
            ["hdfs", "dfs", "-cat", path],
            stderr=subprocess.STDOUT
        )

        if isinstance(output, bytes):
            output = output.decode('utf-8')
        return output.splitlines()
    except subprocess.CalledProcessError:
        print "Ошибка: невозможно прочитать {}".format(path)
        sys.exit(1)

def main():
    if len(sys.argv) != 3:
        print "Использование: python compare_hdfs_files.py <HDFS_file1> <HDFS_file2>"
        sys.exit(1)

    file1, file2 = sys.argv[1], sys.argv[2]

    print "Сравнение файлов:\n 1 {} \n 2 {}".format(file1, file2)

    lines1 = read_hdfs_file(file1)
    lines2 = read_hdfs_file(file2)

    if lines1 == lines2:
        print "Файлы идентичны"
    else:
        print "Файлы различаются!\n"
        max_len = max(len(lines1), len(lines2))
        for i in range(max_len):
            line1 = lines1[i] if i < len(lines1) else "<no line>"
            line2 = lines2[i] if i < len(lines2) else "<no line>"
            if line1 != line2:
                print "Строка {} отличается:\n File1: {} \n File2: {}".format(i+1, line1, line2)

if __name__ == "__main__":
    main()

```

Скрипты для выполнения:

```

create_files.sh
HDFS_DIR="/user/hduser/compare_test"

echo "=== Создание тестовых файлов ==="

FILE1="/tmp/file1.txt"
FILE2="/tmp/file2.txt"

echo -e "Hello Hadoop!\nThis is file 1." > $FILE1

```

```

echo -e "Hello Hadoop!\nThis is file 2." > $FILE2

hdfs dfs -mkdir -p /tmp
hdfs dfs -put /tmp/file1.txt /tmp/file1.txt
hdfs dfs -put /tmp/file2.txt /tmp/file2.txt

echo "Созданы файлы:"
ls -l $FILE1 $FILE2

init-hdfs.sh

#!/bin/bash

sleep 1

echo "Creating HDFS directories..."
hdfs dfs -mkdir -p /user/hduser/Hadoop

FILE_NAME="MoryakovVY_IUK472B.txt"
echo -e "Hello!\nThis is a test file.\nHDFS automation with Docker." > /tmp/$FILE_NAME
hdfs dfs -put -f /tmp/$FILE_NAME /user/hduser/Hadoop/

LOCAL_FILE="new_file.txt"
echo -e "This is a new file\nContents of the new file" > /tmp/$LOCAL_FILE

hdfs dfs -put -f /tmp/$LOCAL_FILE /user/hduser/Hadoop/

# Set access permissions
hdfs dfs -chmod 770 /user/hduser/Hadoop/$FILE_NAME
hdfs dfs -chmod 770 /user/hduser/Hadoop/$LOCAL_FILE

echo "!!!INPUT!!!\n"
hdfs dfs -put -f - /user/hduser/Hadoop/$FILE_NAME

echo "File created. File content:\n"
hdfs dfs -cat /user/hduser/Hadoop/$FILE_NAME
hdfs dfs -ls /user/hduser/Hadoop/

# Check the created directories and files
echo "HDFS state after automation:"
hdfs dfs -ls -R /user/hduser

echo "HDFS automation completed!"

```

Вывод: в ходе лабораторной работы были получены практические навыки по работе с hadoop и python.