**Федеральное агентство связи**

**Ордена трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**



**Практическая работа № 1**

**По дисциплине**

**Введение в большие данные**

Группа: МБД2431

ФИО: Киреев Артём Александрович

**Москва, 2025**

**Цель работы: получить навыки работы с файловой системой HDFS.**

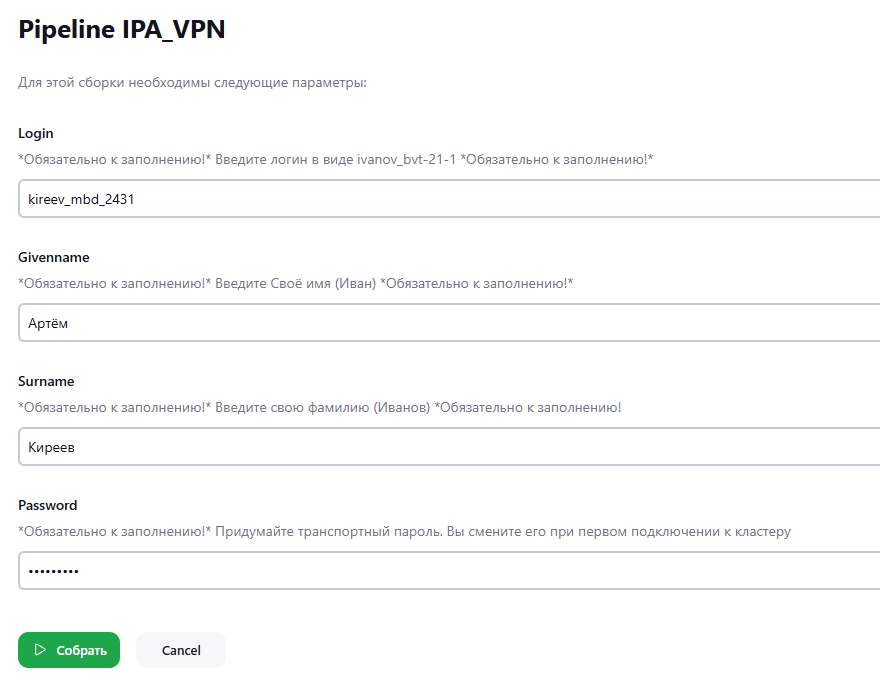


Рисунок 1 - Создание сборки Pieline

Скачиваем клиент для создания OpenVPN-подключения и загружаем \*.ovpn-файл с почты и установите VPN-соединение.

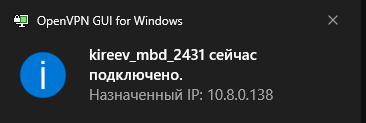


Рисунок 2 - Подключение по OpenVPN

Залогиньтесь на хост по протоколу ssh

ssh kireev\_mbd\_2431@192.168.0.5

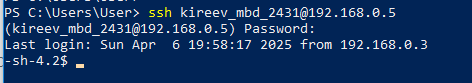


Рисунок 3 - Подключение по ssh

Опробуем консольные утилиты для работы с кластером через HDFS Shell

Напишите команды, с помощью которых выполняли действия и скриншоты с результатами.

1. Создать локально на сервере тестовый файл

echo "test text" >> test

1. Увеличить размер файла, чтобы он превышал размер одного блока HDFS fallocate -l 333M test

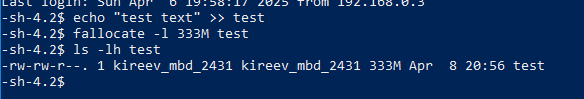


Рисунок 4 – Создать и увеличить размер файла

1. Создать новую директорию в hdfs по пути /data/<ваш\_логин>

hdfs dfs -mkdir /data/kireev\_mbd\_2431

1. Создать новую директорию в hdfs по пути /data/<ваш\_логин>/test\_dir

hdfs dfs -mkdir /data/kireev\_mbd\_2431/test\_dir

hdfs dfs -ls /data/kireev\_mbd\_2431

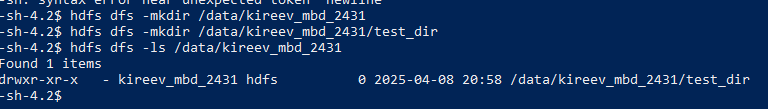


Рисунок 5 - Создание директории в hdfs

1. Положить в HDFS файл test по пути /data/<ваш\_логин>/test\_dir/test

hdfs dfs -put test /data/kireev\_mbd\_2431/test\_dir/test

hdfs dfs -ls /data/kireev\_mbd\_2431/test\_dir

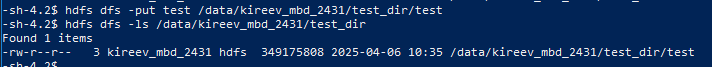


Рисунок 6 - Перекладывание файла

1. Скопировать директорию /data/<ваш\_логин>/test\_dir в /data/<ваш\_логин>/test\_dir\_1

hdfs dfs -cp /data/kireev\_mbd\_2431/test\_dir /data/kireev\_mbd/2431/test\_dir\_1

hdfs dfs -ls /data/kireev\_mbd\_2431

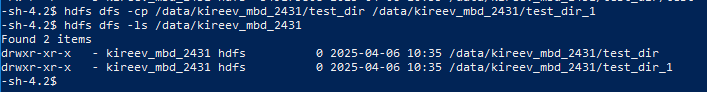


Рисунок 7 - Скопировать директорию

1. Удалить файл test из директории test\_dir\_1 без сохранения файла в корзине.

hdfs dfs -rm -skipTrash /data/kireev\_mbd\_2431/test\_dir\_1/test

hdfs dfs -ls /data/kireev\_mbd\_2431/test\_dir\_1

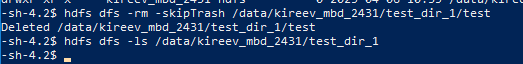


Рисунок 8 - Удалить файл test из директории

1. Просмотреть размер любой директории

hdfs dfs -du -h /data/kireev\_mbd\_2431

hdfs dfs -du -h /data/kireev\_mbd\_2431/test\_dir





Рисунок 9 - Просмотреть размер любой директории

1. Посмотреть, как файл /data/<ваш\_логин>/test\_dir/test хранится на файловой системе (см. команду hdfs fsck).

hdfs fsck /data/kireev\_mbd\_2431/test\_dir/test -files -blocks -locations

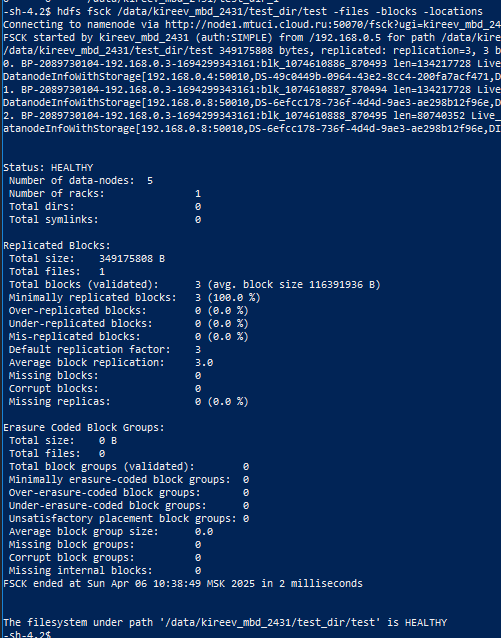


Рисунок 10 - Хранение файла на файловой системе

1. Какой фактор репликации установлен на кластере?

Default replication factor: 3 в выводе команды hdfs fsck.

1. Сколько блоков составляют файл?

Total blocks (validated): 3 в выводе команды hdfs fsck.

Когда выполняется команду hdfs fsck /data/<ваш\_логин>/test\_dir/test -blocks -files -locations, вы получаете подробную информацию о каждом блоке, из которого состоит файл

0. BP-2089730104-192.168.0.3-1694299343161:blk\_1074610886\_870493 len=134217728 Live\_repl=3 [DatanodeInfoWithStorage[192.168.0.8:50010,DS-6efcc178-736f-4d4d-9ae3-ae298b12f96e,DISK], DatanodeInfoWithStorage[192.168.0.4:50010,DS-49c0449b-0964-43e2-8cc4-200fa7acf471,DISK], DatanodeInfoWithStorage[192.168.0.5:50010,DS-78f1e807-efc1-4cc0-875d-44182c85393b,DISK]]

3. Заполните таблицу для данных первого блока Вашего тестового файла

Таблица 1. Для данных первого блока

|  |  |
| --- | --- |
| 0. | Номер блока по порядку |
| BP-2089730104-192.168.0.3-1694299343161 | Идентификатор block pull |
| blk\_1074610886 | Идентификатор блока |
| 870493 | Generation stamp. |
| 134217728 | Объем блока |
| 3 | Количество живых реплик блока |
| 192.168.0.8:50010 | IP-адрес и порт, по которому доступен блок |
| DS-6efcc178-736f-4d4d-9ae3-ae298b12f96e | Data Storage ID идентификатор ноды (Если у ноды изменится hostname или IP-адрес, нода всё равно будет идентифицироваться внутри HDFS) |
| DISK | Способ хранения блока (Может также хранится в S3-хранилище) |

4. Скопируйте результат работы команды для любого из блоков, составляющих ваш тестовый файл. Какие данные мы получили?

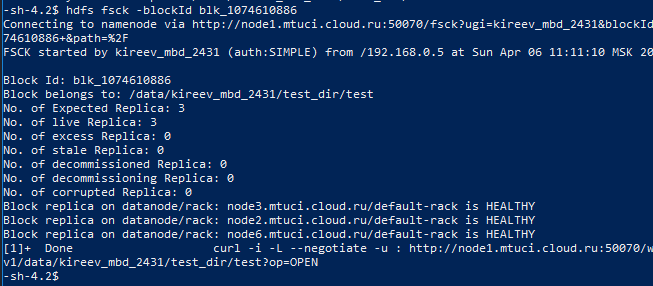


Рисунок 11 - Результат работы команды для любого из блоков

Блок blk\_1074610886 находится в хорошем состоянии. Все реплики блока доступны, не повреждены и находятся на разных DataNodes.

Фактор репликации для этого блока установлен на 3, и все реплики в порядке.

Реплики блока распределены по трем разным DataNodes (node3, node2, node6), что обеспечивает высокую доступность и надежность данных.

5. Выполним вывод информации о файле /data/<ваш\_логин>/test\_dir/test:

curl -i --negotiate -u : http://node1.mtuci.cloud.ru:50070/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir/test?op=GETFILESTATUS

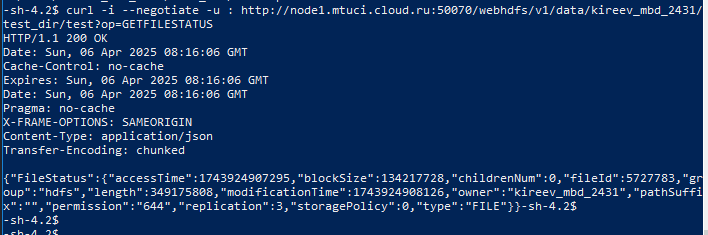


Рисунок 12 - Вывод информации о файле

6. Какая информация выводится в результате работы команды?

Таблица 2. Информация о состоянии и характеристиках файла в HDFS.

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Значение** | **Описание** |
| accessTime | 1743924907295 | Время последнего доступа к файлу в миллисекундах с 1 января 1970 года |
| blockSize | 134217728 | Размер блока в байтах (128 MB) |
| childrenNum | 0 | Количество дочерних элементов (для файла всегда 0) |
| fileId | 5727783 | Уникальный идентификатор файла |
| group | hdfs | Группа, которой принадлежит файл |
| length | 349175808 | Размер файла в байтах (около 333 MB) |
| modificationTime | 1743924908126 | Время последней модификации файла в миллисекундах с 1 января 1970 года |
| owner | kireev\_mbd\_2431 | Владелец файла |
| pathSuffix | "" | Суффикс пути (пустой в этом случае) |
| permission | 644 | Разрешения файла (читать и записывать для владельца, читать для группы и других) |
| replication | 3 | Фактор репликации файла |
| storagePolicy | 0 | Политика хранения (0 - по умолчанию) |
| type | FILE | Тип элемента (файл) |

7. Теперь попробуем прочитать первые 10 символов тестового файла, используя тот же синтаксис команды:

curl -i --negotiate -u : http://node1.mtuci.cloud.ru:50070/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir/test?op=OPEN&length=10

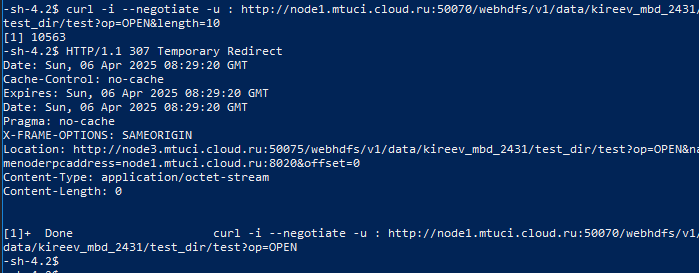


Рисунок 13 - Первые 10 символов тестового файла

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Значение** | **Описание** |
| sber-node | node1.mtuci.cloud.ru | Адрес ноды, на которой установлен клиент HDFS |
| Порт подключения к REST API | 50070 | Порт, используемый для подключения к REST API |
| --negotiate | --negotiate | Включает SPNEGO в curl для аутентификации Kerberos |
| /webhdfs/v1 | /webhdfs/v1 | Адрес Web API |
| /data/<ваш\_логин>/test\_dir/test | /data/kireev\_mbd\_2431/test\_dir/test | Путь к нужному файлу в HDFS |
| op= | OPEN | Производимое действие (открыть файл) |
| length= | 10 | Желаемая длина считывания символов (10 символов) |

8. Почему мы не получили требуемых данных?

NameNode перенаправляет к DataNode, на котором хранится запрашиваемый блок данных

9. Проанализируйте ссылку раздела location из ответа сервера

Перенаправление: NameNode перенаправляет вас к DataNode, на котором хранится запрашиваемый блок данных. http://node3.mtuci.cloud.ru:50075/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir/test?op=OPEN&namenoderpcaddress=node1.mtuci.cloud.ru:8020&offset=0

Это позволяет оптимизировать доступ к данным, минуя NameNode после первоначального запроса.

10. Для того, чтобы получить желаемые данные мы можем добавить в curl-запрос флаг -L (location). Эта опция заставит Curl повторить запрос для нового адреса, который мы получили ранее в Location.

curl -i -L --negotiate -u : http://node1.mtuci.cloud.ru:50070/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir/test?op=OPEN&length=10

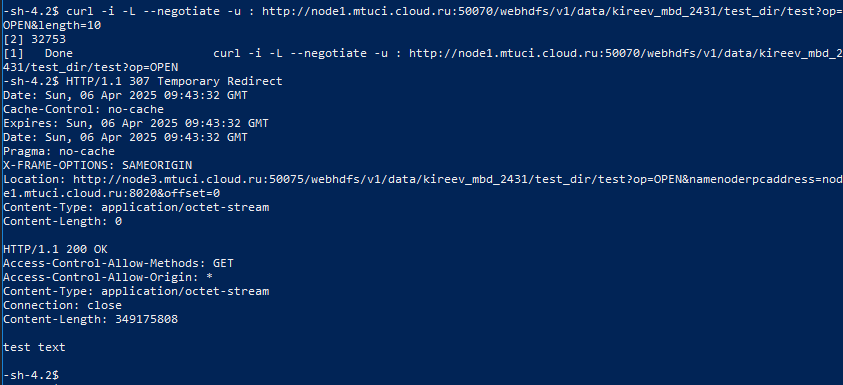


Рисунок 14 - Вывод верной информации

11. Скопируйте файл /test из /data/<ваш\_логин>/test\_dir в test\_dir\_1 и удалите его /data/<ваш\_логин>/test\_dir\_1/test с помощью curl-команды. Приложите скриншоты.

curl -i --negotiate -u : -X PUT \ “http://node1.mtuci.cloud.ru:50070/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir\_1/test?op=CREATE&overwrite=true”

curl -i --negotiate -u : -X DELETE "http://192.168.0.3:50070/webhdfs/v1/data/kireev\_mbd\_2431/test\_dir\_1/test?op=DELETE&recursive=true"

hdfs dfs -ls /data/kireev\_mbd\_2431/test\_dir\_1/

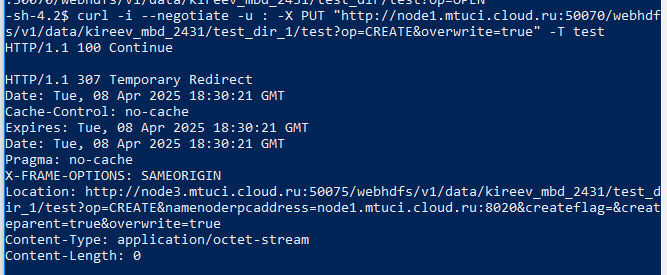


Рисунок 15 - Копирование файла

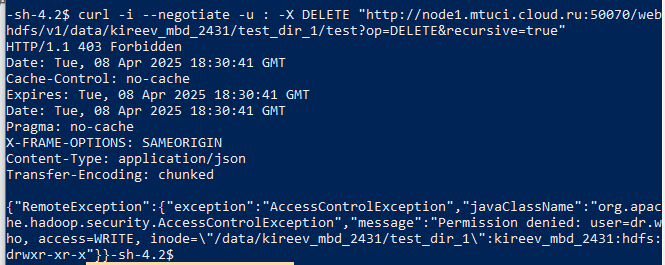


Рисунок 16 - Удаление файла

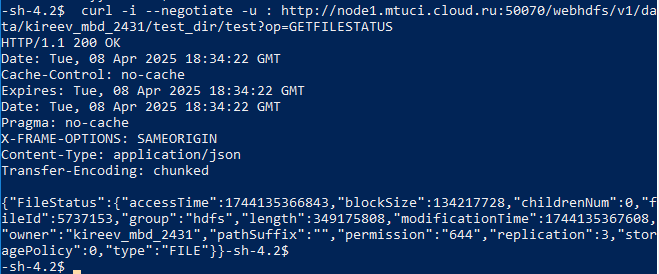


Рисунок 17 - Проверка удаления

**Работа с UI**

12. Залогиньтесь в Hadoop Manager Ambari http://192.168.0.3:8080/

Логин и пароль для входа – monitor/monitor

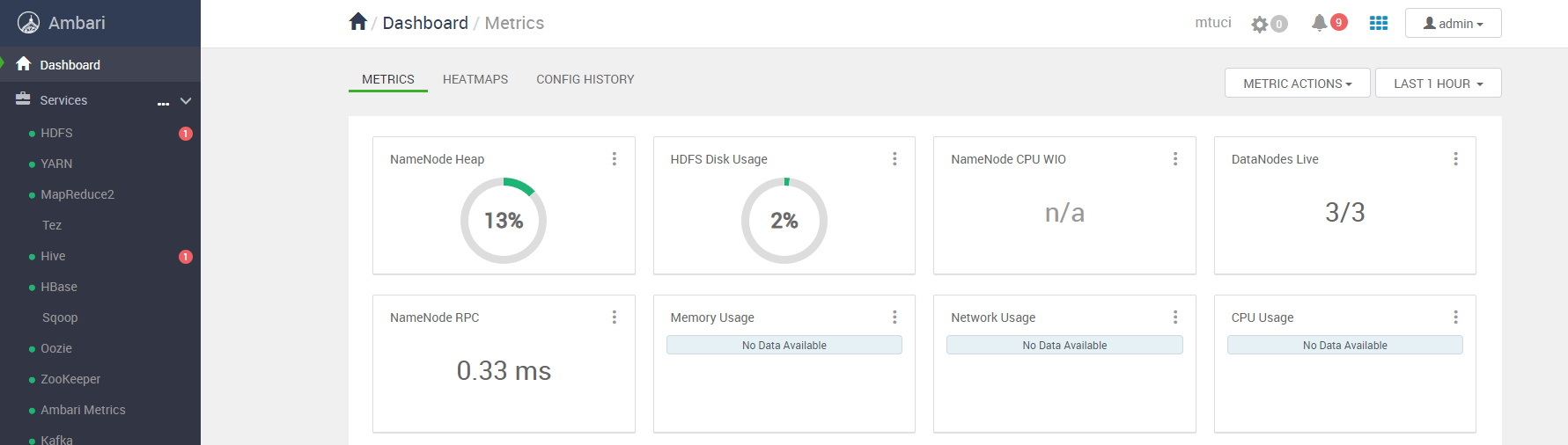


Рисунок 18 – Интерфейс Hadoop

Через Ambari осуществляется настройка и мониторинг кластера Hadoop.

13. Войдите в NameNode UI http://192.168.0.3:50070/

(Адрес можно увидеть в разделе Services->HDFS->QuickLinks)

В интерфейсе собрана наиболее важная информация о состоянии HDFS

14. Зайдите в Utilities->Browse FileSystem.

15. Найдите созданный вами файл в директории /data.

16. Нажмите на название файла и внимательно изучите информацию о блоках данных.

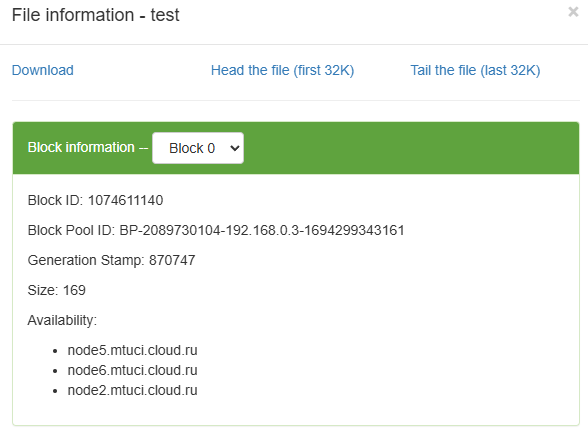


Рисунок 20 - Информация о блоках данных

17. Ответьте на вопрос: какой объём стандартного блока HDFS исходя из показателя Size?

Ответ: 169 байт

18. Есть ли блок, который отличается по размеру от остальных?

Ответ: Да, последний блок файла может быть меньше стандартного размера блока, если размер файла не кратен стандартному размеру блока.

19. Сколько пространства занимает последний блок файла в локальной файловой системе?

Ответ: 169 байт в локальной файловой системе

20. Подумайте, при каких обстоятельствах количество реплик блоков файла может превышать стандартный фактор репликации?

Ответ: если администратор кластера изменил стандартный фактор репликации. Если файл был помечены для хранения с более высоким уровнем репликации (например, для критически важных данных). В случае ошибок или неисправностей в кластере, HDFS может автоматически увеличить количество реплик для обеспечения надежности данных.