

Работа с HDFS

1. Подготовка и вход в систему

Подготовьте файл `Работа с HDFS - отчёт.docx`. Помещайте в него выполняемые команды и вставляйте скриншоты выводимых на экран результатов.

Скачайте архив `bigd_course_hadoop.zip` и распакуйте его.

Откройте файл `docker-compose.yaml` в простом текстовом редакторе, например, в Блокноте. Изучите конфигурацию кластера. Создайте в отчёте таблицу по образцу ниже и запишите в неё информацию об узлах кластера (за исключением IP адресов). При необходимости найдите описание предназначения узлов в Интернете.

Локальное доменное имя (hostname)	IP адрес в сети кластера	Отображаемые порты	Монтируемые тома и файлы хоста	Предназначение
X	X	X	X	X
X	X	X	X	X

Находясь в консоли перейдите в папку `bigd_course_hadoop` и запустите кластер командой

```
| docker-compose up -d
```

Узнайте имя сети Docker'a, созданной для кластера. Для этого выполните команду

```
| docker network ls
```

Так как в конфигурационном файле имя сети не задано явно, её название по умолчанию будет совпадать с именем папки, в котором находится `docker-compose.yaml` с добавкой `_default`.

Выведите на экран информацию о структуре сети

```
| docker network inspect <NETWORK_NAME>
```

Найдите IP адреса узлов и поместите их в таблицу выше.

Среди портов кластера, отображаемых на хост (компьютер, на котором запущен Docker) найдите Hadoop IPC port. Запишите в отчёт номер порта и его предназначение (найдите информацию об этом в Интернете).

Дождитесь, пока все контейнеры завершат процесс инициализации и не получат статус `healthy`. Для этого выполните несколько раз команду

```
| docker ps
```

пока не увидите изменение статуса всех контейнеров.

В выводе команды `doker ps` в последнем столбце `NAMES` узнайте имя контейнера узла имён `<NAME>`. Затем в этом контейнере запустите оболочку командной строки `bash` и перейдите в неё. Для этого выполните в консоли хоста следующую команду:

```
| docker exec -it <NAME> bash
```

Замечание: в нашем примере кластер развернут при помощи Docker-контейнеров на одном компьютере. Поэтому наиболее простой способ обратиться к консоли контейнера - при помощи команды выше запустить в контейнере оболочку командной строки. Для обращения к реальному кластеру, развернутому на удалённых компьютерах, используется протокол SSH (Secure Shell) и одноимённая утилита `ssh`. После авторизации через `ssh` мы точно также попадаем в консоль узла, как это делается в данном задании.

Проверьте, что HDFS работает и доступен, выполнив команду просмотра корневой директории:

```
| hdfs dfs -ls /
```

Если команда выполнилась без ошибок, вы готовы к работе.

2. Базовые операции

2.1 Навигация и создание каталогов

Создайте в корне файловой системе папку с именем `user`.

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

Внутри папки `user` создайте две подпапки: `data` и `results`.

```
| hdfs dfs -mkdir /user/data  
| hdfs dfs -mkdir /user/results
```

В папке `data` создайте подпапку `logs`.

```
| hdfs dfs -mkdir /user/data/logs
```

Для быстрого выполнения команд, находясь в консоли нажмайтe на клавиатуре клавишу "стрелка вверх" чтобы вернуться к предыдущей команде и отредактировать её.

Просмотрите структуру созданных каталогов командами `-ls` и `-ls -R` (рекурсивный просмотр). Сравните отличия.

```
| hdfs dfs -ls /user  
| hdfs dfs -ls -R /user
```

Обратите внимание: в HDFS нет аналога команды перехода в другой каталог `cd` (change dir). Все операции с файлами выполняются с указанием полного пути.

2.2 Копирование файлов с хоста в локальную файловую систему узла

Выходите из консоли контейнера, выполнив следующую команду

```
| exit
```

Находясь в консоли хоста, выполните команду копирования всей папки с рабочими файлами `files` в локальную файловую систему узла имён (вместо `<NAME>` впишите имя контейнера узла имён).

```
| docker cp files <NAME>:/home/user
```

При выполнении этой команды в локальной файловой системе в папке `home` будет создана папка `user` и в неё будет скопировано содержимое папки `files`.

Команда `docker cp` работает следующим образом. Если копируется папка (в команде выше - это `files`) и при этом целевая папка не существует (папка `user`) то эта папка будет создана и содержание исходной папки будет помещено в новую папку (всё что есть в папке `files` окажется во вновь созданной папке `/home/user`). Но если целевая папка уже существует, то в неё будет скопировано не содержание исходной папки, а сама папка. Чтобы с этим разобраться, выполнить команду копирования повторно. Просматривая локальную файловую систему узла имён (см. команды ниже), вы увидите, что файлы из `files` оказались в папке `/home/user` (результат выполнения первой команды), а ещё в папке `/home/user` появилась папка `files` с теми же самыми файлами (результат повторного выполнения той же самой команды).

Снова запустите консоль в контейнере узла имён и перейдите в неё

```
| docker exec -it <NAME> bash
```

Посмотрите содержимое папки `/home/user` и убедитесь, что там появились файлы из папки `files` хоста. Для этого нужно выполнить команду просмотра папки в Linux:

```
| ls /home/user
```

Теперь просмотрите содержание этой папки, добавив к команде `ls` опцию `-lh`.

```
| ls -lh /home/user
```

Замечание: опция `l` - от слова "long", т.е. "длинная" информация о файлах, опция `h` означает "human readable", т.е., формат, удобный для человеческого восприятия, благодаря ей длина файла будет показана не в байтах, а в более удобных килобайтах и мегабайтах.

Вы увидите расширенную информацию об находящихся в этой папке файлах, в частности их размеры. Запишите в файл отчёта размер файла `access_log.gz`.

Обратите внимание: сейчас Вы работаете с локальной файловой системой узла имён, а не с файловой системой кластера HDFS.

Файл `access_log.gz` имеет расширение `.gz`. Это значит, что он запакован при помощи архиватора `gzip`, обычно используемого в `Linux`.

Распакуйте файл `access_log.gz` следующей командой:

```
| gzip -d /home/user/access_log.gz
```

Снова посмотрите содержимое папки `/home/user`

```
| ls -lh /home/user
```

Вы должны увидеть распакованный файл `access_log`. Обратите внимание, что исходный запакованный файл `access_log.gz` был удалён. Это особенность работы архиватора `gzip` - по умолчанию он не сохраняет исходные файлы. Запишите в отчёт размер распакованного файла.

После того как файлы подготовлены в локальной файловой системе, их можно скопировать в HDFS. Файлы при этом будут разделены на блоки, которые будут размещены на узлах данных кластера.

2.3 Копирование файлов из локальной файловой системы узла в HDFS

Копирование файлов из локальной файловой системы в HDFS выполняется при помощи команды `put`. Скопируйте в HDFS три файла, приведённые ниже.

```
| hdfs dfs -put /home/user/canberra_weather.csv /user/data  
hdfs dfs -put /home/user/exams.csv /user/data  
hdfs dfs -put /home/user/access_log /user/data/logs
```

Выведите рекурсивно содержание своей рабочей папки `user` в файловой системе HDFS.

```
| hdfs dfs -ls -R -h /user
```

Обратите внимание на дополнительную опцию `-h`. Она отвечает за то, чтобы показать размер файлов в удобных единицах измерения. Сравните результаты работы этой команды с этой опцией и без неё.

2.4 Просмотр содержимого файлов в HDFS

Выведите на экран содержимое файла `exams.csv` из HDFS на экран:

```
| hdfs dfs -cat /user/data/exams.csv
```

Этот файл содержит проходные балы ЕГЭ по разным предметам одного из университетов. Так как файл небольшой, при помощи команды `-cat` мы смогли просмотреть всё его содержание.

Теперь также выведите на экран файл `canberra_weather.csv`

```
| hdfs dfs -cat /user/data/canberra_weather.csv
```

Этот файл содержит значительно больше информации и на экране мы видим только его конец. В таких случаях можно показать на экране только начало файла при помощи `-head`:

```
| hdfs dfs -head /user/data/canberra_weather.csv
```

Также можно показать только последнюю часть файла - командой `-tail`:

```
| hdfs dfs -tail /user/data/logs/access_log
```

3. Операции внутри HDFS

3.1 Копирование и перемещение файлов внутри HDFS

Создайте копию файла `exams.csv` внутри папки `data` с именем `exams_copy.csv`:

```
| hdfs dfs -cp /user/data/exams.csv /user/data/exams_copy.csv
```

Переместите файл `exams_copy.csv` в папку `results` с новым именем `processed_data.csv`.

```
| hdfs dfs -mv /user/data/exams_copy.csv /user/results/processed_data.csv
```

Замечание. За переименование и перемещение отвечает одна и та же команда `-mv`. Если нужно только переименовать файл, то в качестве целевой указывается та же самая папка.

Проверьте содержимое обеих папок (`data` и `results`), чтобы убедиться в успешности операций.

```
| hdfs dfs -ls /user/data  
hdfs dfs -ls /user/results
```

3.2 Удаление файлов и каталогов

Удалите файл `exams.csv` (а также файл `exams_copy.csv`, если он остался) из папки `data`:

```
| hdfs dfs -rm /user/data/exams.csv
```

Создайте новую папку `new_results` в папке `user`.

```
| hdfs dfs -mkdir /user/new_results
```

Пока папка пустая, её можно удалить командой `-rmdir`. Проверьте это, удалив вновь созданную папку:

```
| hdfs dfs -rmdir /user/new_results
```

Теперь попробуйте таким же образом удалить непустую папку `results`. Это должно привести к ошибке.

```
| hdfs dfs -rmdir /user/results
```

Команда `-rmdir` удаляет только пустую папку. Для удаления папки в месте с файлами используется комбинация команд `-rm -R`. Опция `-R` означает recursive, т.е., удалить рекурсивно папку вместе со своим содержимым:

```
| hdfs dfs -rm -R /user/results
```

4. Управление правами доступа

Получите детальную информацию о файле `canberra_weather.csv` при помощи команды `-ls -h`.

Самостоятельно найдите в Интернете описание того, что выводит эта команда и поместите в отчёт имя владельца файла и группу.

Самостоятельно, при помощи Интернета, разберитесь, какую информацию о правах доступа к файлам выводит команда `-ls`. Поместите в отчёт информацию о том, каким правами обладает владелец файла, группа и другие пользователи.

Самостоятельно при помощи Интернета, выясните как при помощи команды `-chmod` задавать права на чтение и запись владельцу файла, группе и другим. Можно прочитать информацию о команде Linux `chmod`, которая работает по тем же принципам.

Измените права доступа на файл `canberra_weather.csv`, сделав его доступным только для чтения владельцу. Группа и остальные не должны иметь никаких прав. Добавьте в отчёт скриншот вывода команды `-ls -h`, демонстрирующий правильно заданные права. Стока с описанием прав доступа должна выглядеть следующим образом: `-r-----`.

Верните стандартные права обратно: владелец может читать и писать, группа и остальные только читать: `-rw-r--r--`. Не забудьте поместить скриншот в отчёт.

5. Анализ распределения данных в HDFS

5.1 Распределение блоков по узлам данных

Получите информацию о блоках файла `access_log`:

```
| hdfs fsck /user/data/logs/access_log
```

Запишите в отчёт ответы на следующие вопросы:

- Сколько узлов данных доступно в кластере?
- Какой размер файла в байтах?
- На сколько блоков разделен файл?
- Какой средний размер блока?
- Чему равен коэффициент репликации по умолчанию?
- Чему равна среднее число реплик блоков?
- Сколько блоков повреждено?

Получите информацию об этом файле через веб-интерфейс. Для этого выполните следующие действия:

- Откройте веб-браузер и перейдите по адресу <http://localhost:9870>
- Перейдите в меню "Datanodes" чтобы увидеть список всех DataNode и информацию о них
- Используйте меню "Utilities" -> "Browse the file system" для навигации по файловой системе
- Найдите файл `access_log` и кликните по его имени.

Создайте в отчёте следующую таблицу. Число строк должно быть равно числу блоков.

Запишите в отчёт для каждого блока его ID и через запятую имена узлов данных, на которых находятся его данные в столбец "Имена узлов до остановки".

ID блока	Имена узлов до остановки	Имена узлов после остановки
x	x	x
x	x	x

5.2 Моделирование сбоя узла данных и наблюдение за восстановлением

Определите какие узлы данных работают в кластере:

```
| hdfs dfsadmin -report
```

Проанализируйте вывод этой команды. Для каждого узла данных создайте в отчёте таблицу следующего вида:

Параметр	До остановки узла	После остановки узла
Имя узла (hostname)	x	x
IP-адрес и порт	x	x
Объём памяти, занятой HDFS (DFS Used)	x	x
Количество блоков, сохранённых на узле	x	x
Статус узла (работает/остановлен)	x	x

Заполните столбцы "До остановки узла" информацией, полученной на основании вывода выполненной команды.

Найдите в отчёте список узлов, на которых находятся реплики нулевого блока файла `access_log`. Выйдите из консоли узла имён при помощи команды `exit` и, находясь в командной строке своего компьютера выполните команду остановки одного из этих узлов:

```
| docker stop datanodeN
```

Подождите несколько минут проверьте статус кластера:

```
| hdfs dfsadmin -report
```

Замечание: по умолчанию узел имён перепроверяет доступность узлов данных с интервалом 10 минут. Но для экономии времени учебный кластер, с которым Вы работаете, настроен на более частую проверку.

Обратите внимание на изменение количества работающих узлов данных. В таблицах с описаниями узлов данных заполните вторые столбцы "После остановки узла".

Используя веб-интерфейс, получите информацию, на каких узлах расположены блоки файла `access_log`. Если Вы увидите, что блок имеет только две копии, подождите некоторое время, пока не закончится восстановление репликации. Запишите имена узлов каждого из блоков в таблицу выше, в столбец "Имена узлов после остановки".

5.3 Моделирование ввода в работу новых узлов

Чтобы работа балансировщика узлов данных стала более наглядной, мы остановим ещё одни узел данных, скопируем на кластер ещё один файл, а потом запустим оба остановленных узла. Наша цель увидеть, как при появлении новых узлов данных, происходит автоматический перенос блоков на новые узлы.

Выходите из консоли узла имён, остановите ещё один узел данных и вернитесь в консоль.

Затем используя веб-интерфейс дождитесь, когда кластер зарегистрирует отключение ещё одного узла (меню "DataNodes"), затем покажет, что теперь блоки файла `access_log` имеют только по две реплики.

Теперь добавьте в HDFS ещё одну копию файла `access_log` под именем `access_log2`:

```
| hdfs dfs -put /home/user/access_log /user/data/logs/access_log2
```

При помощи веб-интерфейса получите информацию, на каких узлах находятся блоки файла `access_log2` и поместите это в таблицу ниже, в столбец "Имена узлов до восстановления".

ID блока	Имена узлов до восстановления	Имена узлов после восстановления
x	x	x
x	x	x

Теперь выйдете из консоли узла имён и из командной строки своего компьютера запустите оба ранее остановленных узла:

```
| docker start datanodeN1  
docker start datanodeN2
```

Снова войдите в консоль узла имён и проверьте статус кластера. Убедитесь, что все узлы имеют статус работающих.

```
| hdfs dfsadmin -report
```

Теперь дождитесь, пока репликация узлов файла `access_log2` будет восстановлена, т.е., каждый блок будет иметь по три копии на разных узлах.

Запишите имена узлов, на которых оказались копии блоков в таблицу выше, в столбец "Имена узлов после восстановления".

После того как выполнение задания окончено, выйдите из консоли узла имён и остановите кластер

```
| docker compose down
```

Для очистки долговременной памяти своего компьютера, удалите тома файловых систем узлов кластера командой ниже.

```
| docker volume prune --all
```

ВАЖНО: все локальные файловые системы узлов кластера, а также все файлы в HDFS будут удалены! Не делайте этого, пока не отчитаетесь по заданию преподавателю!

ВАЖНО: скорее всего после выполнения этого задания Вы будете выполнять следующее, которое тоже посвящено работе с этим кластером. Поэтому лучше пока воздержаться от удаления томов кластера.