**Григорьев Ю.А.**

**Методические указания к лабораторным работам 7,8.**

**Работа с Hadoop и Spark**

Оглавление

[1. Цель и задачи лабораторной работы 1](#_Toc57650629)

[2. Установка Hadoop 1](#_Toc57650630)

[2.1. Скопировать каталог lab78 1](#_Toc57650631)

[2.2. Создание пользователя системы для работы с HADOOP 2](#_Toc57650632)

[2.3. Установить и настроить SSH. 2](#_Toc57650633)

[2.4. Установка HADOOP 3](#_Toc57650634)

[2.5. Настройка HADOOP 4](#_Toc57650635)

[2.6. Форматирование HDFS и запуск Hadoop 8](#_Toc57650636)

[3. Установка, настройка и проверка работы Spark 10](#_Toc57650637)

[3.1. Установка и настройка Spark 10](#_Toc57650638)

[3.2. Проверка работы Hadoop+Spark 11](#_Toc57650639)

[3.3. Подсчёт числа слов в тексте (на примере пьесы Шекспира «Гамлет») 12](#_Toc57650640)

[4. Содержание отчета по лабораторной работе 14](#_Toc57650641)

[5. Источники информации 14](#_Toc57650642)

Внимание!

1. Не забывайте делать копию своей виртуальной машины или среды, в которой вы работаете, на случай восстановления системы после сбоя.

2. Перед выполнением лабораторной работы необходимо полностью прочитать методические указания.

# Цель и задачи лабораторной работы

Целью лабораторной работы является приобретение навыков инсталляции продуктов Apache Hadoop и Apache Spark, поддерживающих технологию MapReduce. Эти системы используются для обработки больших данных (Big Data).

# Установка Hadoop

## Скопировать каталог lab78

1. Запустите виртуальную машину. Откройте терминал (пункт Open Terminal контекстного меню).

2. Создайте каталог lab78 в домашнем каталоге /home/user

mkdir lab78

3. Перепишите 3 файла из яндекс-папки /Файлы/lab78 в /home/user/lab78 своей ВМ. В этом каталоге хранятся 3 файла: архив Hadoop, архив Spark и текстовый файл для анализа:

hadoop-2.9.2.tar.gz,

spark-2.4.6-bin-hadoop2.7.tgz,

gamlet\_en.txt.

## Создание пользователя системы для работы с HADOOP

sudo addgroup hadoop

sudo adduser --ingroup hadoop hduser

Создан пользователь hduser в группе hadoop. При создании пользователя важно запомнить пароль (введите hadoop), остальные настройки не важны.

Для входа в систему под новым пользователем необходимо ввести команду

sudo su hduser

(для входа может быть потребуется ввести пароль для текущей учетной записи)

Теперь пользователю hduser необходимо добавить права для выполнения команд типа от рута (sudo):

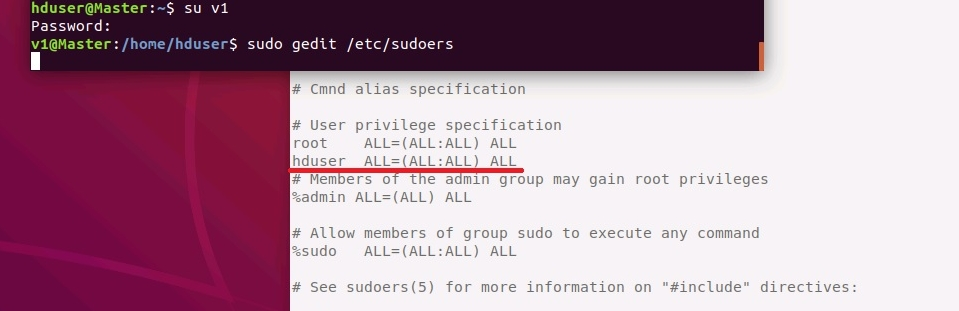
exit перейти к user

sudo su root

nano /etc/sudoers

вставить строку

hduser ALL=(ALL:ALL) ALL



exit перейти к user

## Установить и настроить SSH.

1. Проверить установлен ли SSH-сервера:

dpkg -s openssh-server

Если нет, то установите его: sudo apt-get install openssh-server

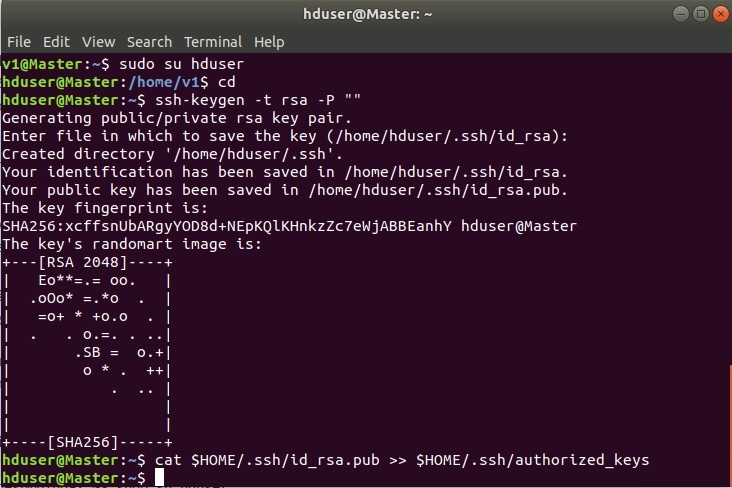
2. Сгенерируйте SSH-ключи (под hduser):

sudo su hduser

cd

ssh-keygen -t rsa -P "" на приглашение Enter file ... нажмите Enter

cat $HOME/.ssh/id\_rsa.pub >> $HOME/.ssh/authorized\_keys



## Установка HADOOP

1. Распаковать архив.

cd /home/user/lab78

sudo tar -xzvf hadoop-2.9.2.tar.gz распаковать архив

2. Переместить распакованный каталог hadoop-2.9.2 в рабочий каталог /usr/local/hadoop с помощью команды mv. Следует отметить, что команда mv создаёт каталог hadoop-2.9.2 в /usr/local/hadoop:

sudo su root

mkdir /usr/local/hadoop

sudo mv hadoop-2.9.2 /usr/local/hadoop

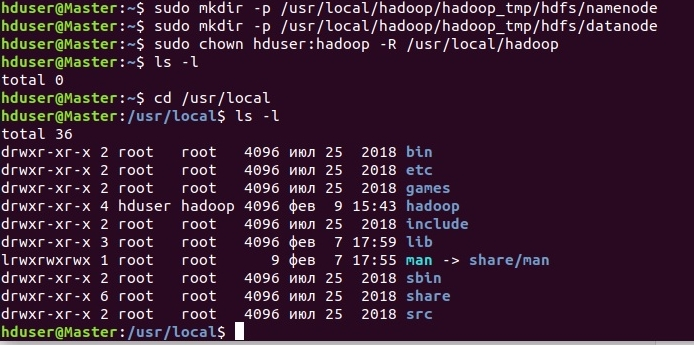
3. Создадим рабочие каталоги для HDFS (NameNode и DataNode) и назначим владельца каталога hadoop (пользователь hduser):

sudo mkdir -p /usr/local/hadoop/hadoop\_tmp/hdfs/namenode

sudo mkdir -p /usr/local/hadoop/hadoop\_tmp/hdfs/datanode

sudo chown hduser:hadoop -R /usr/local/hadoop/

exit вернуться к hduser



4. Настроить переменные окружения:

cd

sudo nano .bashrc

Вставить в конец файла следующие строки

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64

export HADOOP\_HOME=/usr/local/hadoop/hadoop-2.9.2

export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

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 HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"

Примечание. Проверить наличие /usr/lib/jvm/java-1.8.0-openjdk-amd64 :

update-java-alternatives -l

Если нет, то установить

sudo apt install openjdk-8-jdk

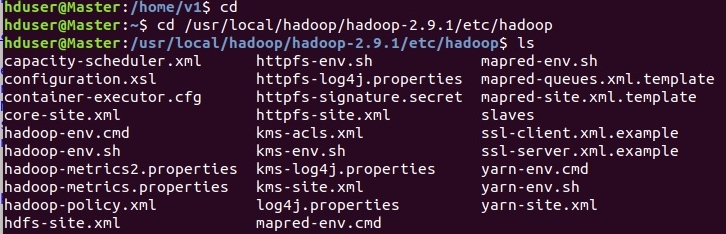
Проверить:

update-java-alternatives -l

## Настройка HADOOP

Все конфигурационные файлы находятся по следующему пути (в нашей конфигурации): /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/

На рисунке вместо hadoop-2.9.1 надо читать hadoop-2.9.2



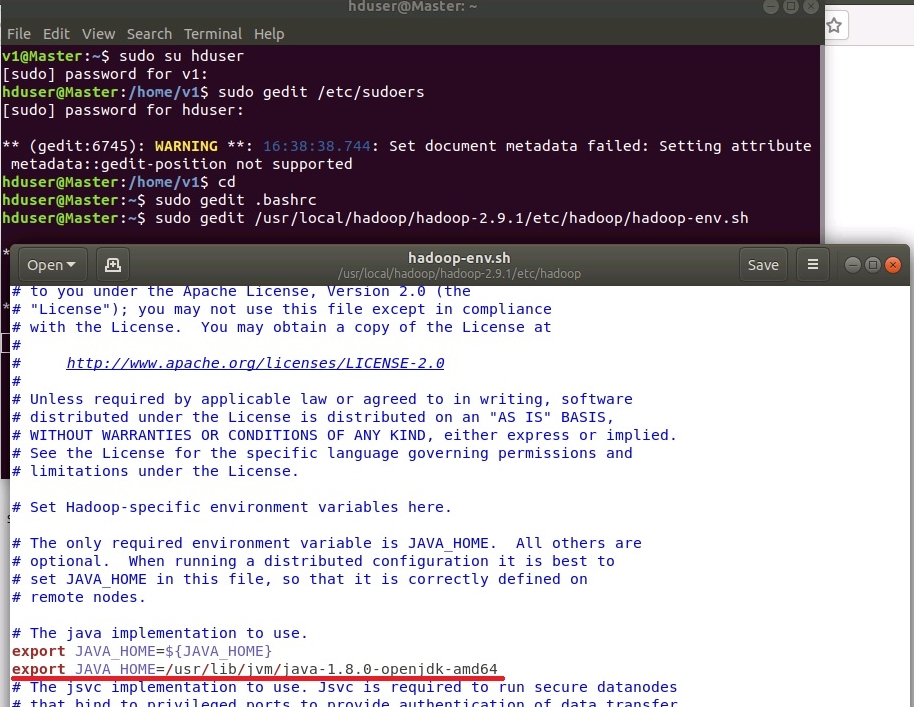
1. Настройка hadoop-env.sh

Добавить переменную JAVA\_HOME:

sudo nano /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/hadoop-env.sh

добавить строку

export JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64



2. Настройка файла core-site.xml

Для этого открыть файл в редакторе с помощью команды

sudo nano /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/core-site.xml

И прописать следующие строки:

<configuration>

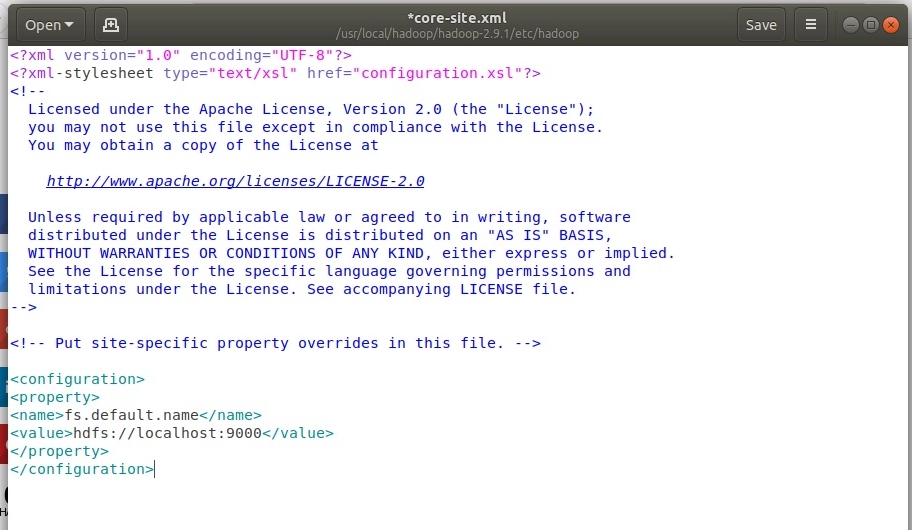
<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>



Здесь указан путь к основной машине (NameNode) файловой системы HDFS, где localhost только для локальной установки HADOOP (в распределенной версии тут либо IP-адрес, либо DNS-имя); 9000 - порт, может быть иным, но лучше не менять.

3. Настройка файла hdfs-site.xml

Для этого открыть файл в редакторе с помощью команды

sudo nano /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/hdfs-site.xml

И добавить строки

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop/hadoop\_tmp/hdfs/namenode</value>

</property>

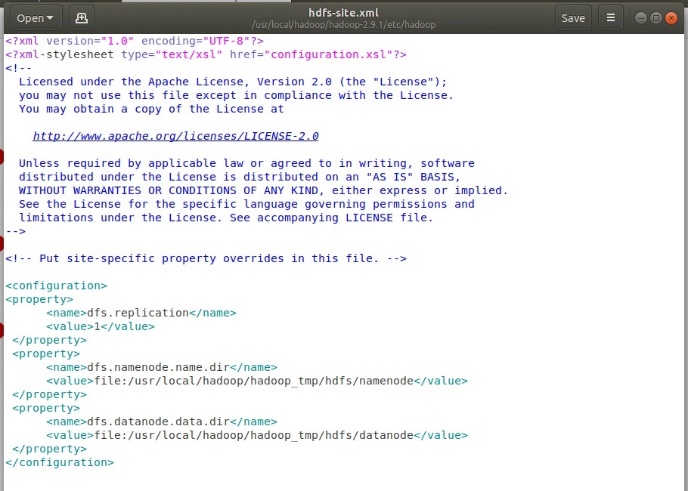
<property>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop/hadoop\_tmp/hdfs/datanode</value>

</property>

</configuration>



Параметры: dfs.replication - количество реплик файлов в HDFS, NameNode - это своеобразная таблица дескрипторов в HDFS (данные о частях и репликах), DataNode - это сами данные.

4. Настройка yarn-site.xml.

Для этого открыть файл в редакторе с помощью команды

sudo nano /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/yarn-site.xml

Добавить строки:

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

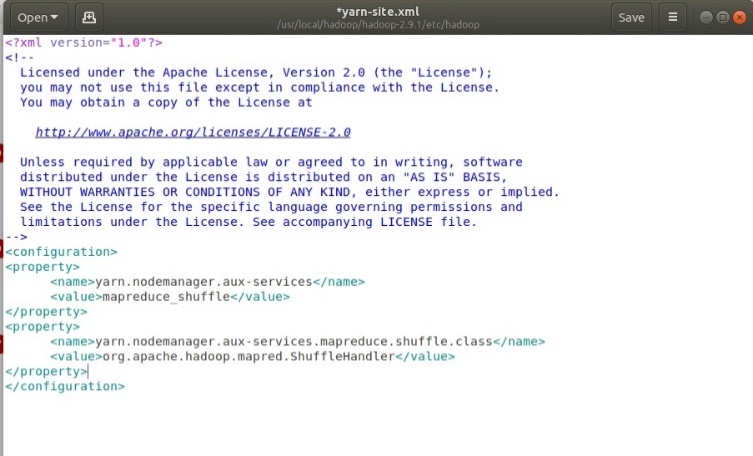
<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

</configuration>



5. Настройка mapred-site.xml.

Файл изначально не существует, есть шаблон, который нужно скопировать командой:

cp /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/mapred-site.xml.template /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/mapred-site.xml

Открыть и отредактировать созданный файл

sudo nano /usr/local/hadoop/hadoop-2.9.2/etc/hadoop/mapred-site.xml

Добавить строки:

<configuration>

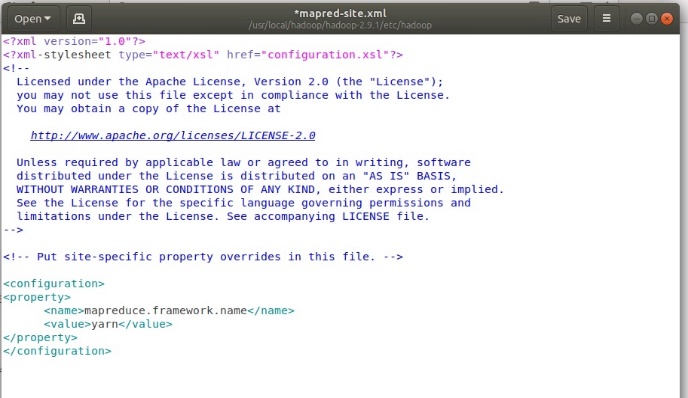
<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>



На этом настройка закончена, необходимо перезагрузить систему:

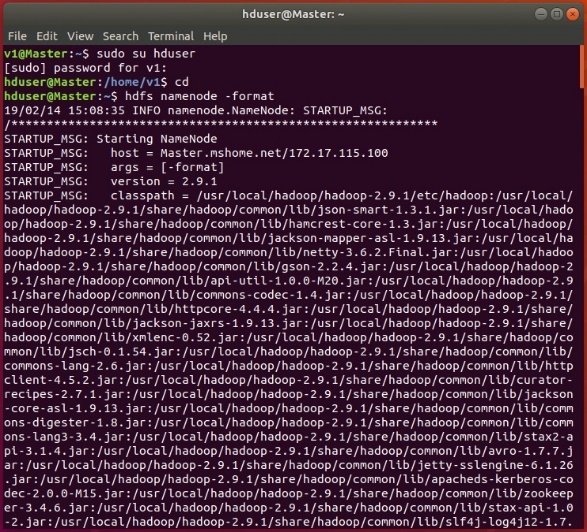
sudo reboot 0

## Форматирование HDFS и запуск Hadoop

1. Форматирование (от пользователя hduser)

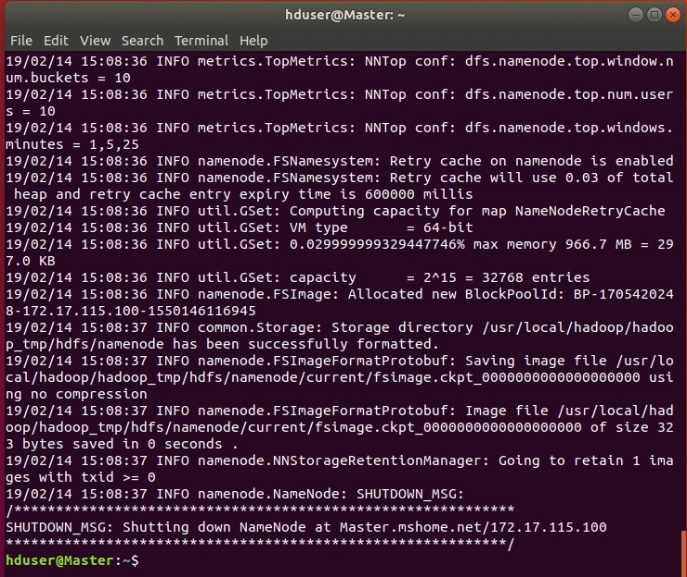
hdfs namenode -format

В ходе форматирования в терминале будут отображены логи:



.

Если все настроено правильно, в логах не должно быть записей, содержащих **ERROR**, **WARN** допускаются. При удачном форматировании конец логов должен выглядеть приблизительно так:



2. Запуск HADOOP

Запуск HADOOP заключается в запуске менеджеров HDFS и YARN. Для этого необходимо запустить два скрипта.

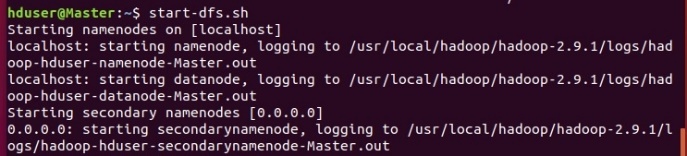
Для HDFS: **start-dfs.sh**

Примечание. Если появляется сообщение, ответьте yes (как показано ниже):

The authenticity of host 'localhost (::1)' can't be established.

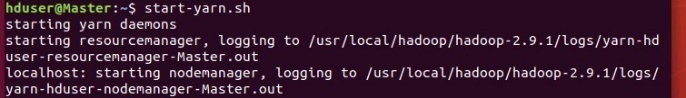
ECDSA key fingerprint is SHA256:Ls7j3nWb9CJzfPX4e9PsQ0nwY2t21QH238x/BoqGw88.

Are you sure you want to continue connecting (yes/no)? yes



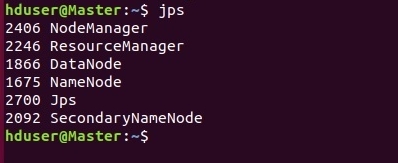
.

Для YARN: **start-yarn.sh**



.

Успешность запуска всех демонов HADOOP можно проверить с помощью команды **jps**.



.

Если все настроено правильно, должно быть запущено 5 процессов HADOOP: NodeManager, NameNode, DataNode, ResourceManager, SecondaryNameNode.

Для облегчения процедуры запуска HADOOP можно создать простой скрипт, для этого перейдите в домашний каталог пользователя **hduser** с помощью команды **cd**.

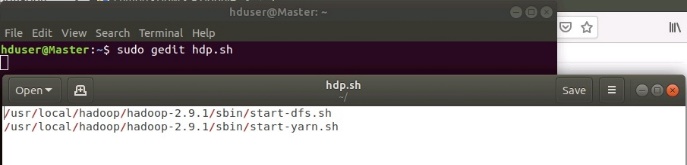
Создайте файл hdp.sh и отредактируйте его с помощью команды:

sudo nano hdp.sh

Добавим в редакторе следующие строки:

/usr/local/hadoop/hadoop-2.9.2/sbin/start-dfs.sh

/usr/local/hadoop/hadoop-2.9.2/sbin/start-yarn.sh



Сохраняем и выходим. Теперь нужно дать права на исполнение для файла hdp.sh. Это делается командой:

sudo chmod 777 hdp.sh

Запускаем наш скрипт:

./hdp.sh



Итак Вы настроили однонодовый кластер HADOOP.

3. Остановить демоны Hadoop:

stop-dfs.sh

stop-yarn.sh

# Установка, настройка и проверка работы Spark

## Установка и настройка Spark

1. Распаковать архив (если отказано в доступе, то под user).

cd /home/user/lab78

sudo tar -xzvf spark-2.4.6-bin-hadoop2.7.tgz распаковать архив

2. Переименовать распакованный каталог и перенести его в рабочий каталог /usr/local с помощью команды mv. Следует отметить, что команда mv создаёт несуществующий каталог, в нашем примере это spark-2.4.6:

mv spark-2.4.6-bin-hadoop2.7 spark-2.4.6

sudo su root

sudo mv spark-2.4.6 /usr/local

3. Назначить владельца каталога spark-2.4.6 (назначить владельцем hduser):

sudo chown hduser:hadoop -R /usr/local/spark-2.4.6/

4. Настроить переменные окружения

sudo su hduser

cd

sudo apt install python2 установить python2

sudo nano .bashrc

В конец добавить следующие строки с переменными:

export SPARK\_HOME=/usr/local/spark-2.4.6

export PATH=$PATH:$SPARK\_HOME/bin

export PYSPARK\_PYTHON=/usr/bin/python2

export PYSPARK\_DRIVER\_PYTHON=python2

5. Принять настройки и проверить

source .bashrc

spark-shell --version должна появиться картинка *Spark*

6. Теперь необходимо настроить файл spark-env.sh

Перейти в каталог настроек:

cd $SPARK\_HOME/conf

Создать файл spark-env.sh из шаблона (template):

sudo cp spark-env.sh.template spark-env.sh

Открыть spark-env.sh в редакторе:

sudo nano spark-env.sh

и добавить в конец строки:

export SPARK\_LOCAL\_IP=127.0.0.1

export PYSPARK\_PYTHON=/usr/bin/python2

export PYSPARK\_DRIVER\_PYTHON=python2

## Проверка работы Hadoop+Spark

Всё под пользователем hduser

1. Запустите Hadoop и проверьте запуск:

cd

start-dfs.sh

start-yarn.sh

jps

2. Создайте каталог в HDFS (файловая система Hadoop) и посмотрите список в корневой папке:

hdfs dfs -mkdir /chapter5

hdfs dfs -ls /

3. Запустите оболочку для работы с Python в Spark:

pyspark

4. Последовательно введите следующие операторы для проверки записи/чтения файла из HDFS:

from pyspark.sql import SparkSession

sparkSession = SparkSession.builder.appName("example-pyspark-read-and-write").getOrCreate()

# Create data

data = [('First', 1), ('Second', 2), ('Third', 3), ('Fourth', 4), ('Fifth', 5)]

df = sparkSession.createDataFrame(data)

# Write into HDFS

df.write.csv("hdfs://localhost:9000/chapter5/example.csv")

# Read from HDFS

df\_load = sparkSession.read.csv('hdfs://localhost:9000/chapter5/example.csv')

df\_load.show()

5. Завершите работу с pyspark:

quit()

6. Запустите оболочку для работы с Scala в Spark и выйдите из неё:

spark-shell

:q (из оболочки можно выходить по Ctrl D)

3. Остановите демоны Hadoop:

stop-dfs.sh

stop-yarn.sh

## Подсчёт числа слов в тексте (на примере пьесы Шекспира «Гамлет»)

Все действия выполняются под пользователем hduser (sudo su hduser).

1. Запустите демоны Hadoop, если они ещё не запущены, и проверьте их запуск

start-dfs.sh

start-yarn.sh

jps

2. Перепишите файл с пьесой «Гамлет» из локальной файловой системы в файловую систему Hadoop:

hdfs dfs -put /home/user/lab78/gamlet\_en.txt /chapter5

hdfs dfs -ls /chapter5

Примечание. Если у hduser нет доступа к lab78, то добавить его в группу user:

sudo gpasswd -a hduser user

3. Запустите оболочку python в Spark:

pyspark

4. Запустите скрипт для подсчёта числа слов в файле gamlet\_en.txt.

Примечание. Разберитесь с этой программой. Выделите приведённый ниже текст, скопируйте его в буфер обмена, в переместите его в окно терминала.

from pyspark import SparkContext

from datetime import datetime

f = sc.textFile("hdfs://namenode:9000/chapter5/gamlet\_en.txt")

counts = f.flatMap(lambda line:line.split(" ")) \

.map(lambda word: (word , 1)) \

.reduceByKey(lambda a, b: a + b) \

.sortBy(lambda a: a[1], ascending=False)

counts.saveAsTextFile("/home/hduser/res/")

sm = counts.map(lambda x: x[1]).sum()

print ("Summa %d" % (sm))

quit()

Примечание. При повторном использовании предыдущей программы старую директорию res надо предварительно удалить (rm -r res).

5. Проверьте работу скрипта:

cd res

ls

nano part-00000

6. Остановите демоны Hadoop после окончания л/р (обязательно!):

stop-dfs.sh

stop-yarn.sh

Примечание. Ещё некоторые полезные команды для работы с HDFS:

hdfs dfs -ls -r / показать корневой каталог с подкаталогами (-r)

hdfs dfs -rm -skipTrash /путь к файлу удалить файл без перемещения в корзину, для удаления каталога надо добавить опцию -R; путь указывается, начиная с корня

hdfs dfs -mkdir /путь к каталогу создать каталог

hdfs dfs -cp /dir/file1 /otherDir/file2 копировать файл

hdfs dfs -mv /dir/file1 /otherDir переместить файл

hdfs dfs -put localfile /dir/file копирование локального файла в директорию HDFS

hdfs dfs -get /dir/file localfile копирование файла из HDFS в локальную файловую систему

# Содержание отчета по лабораторной работе

Отчет должен содержать:

- краткое описание шагов установки Hadoop и Spark (раздел 2 и пункт 3.1),

- скрипт и результаты проверки работы Hadoop и Spark (пункт 3.2),

- скрипт и результат подсчёта числа слов в файле, первые 20 значений (пункт 3.3),

- пояснение результатов.

# Источники информации

1. <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html>

2. <https://xn---57-qdd4aqo.xn--p1ai/pages/aadress.php?page=468>

3. <https://habr.com/ru/post/206196/>

4. <https://xn---57-qdd4aqo.xn--p1ai/pages/aadress.php?page=497#razdel2>

5. <https://gist.github.com/Alexflex/c8c53346b2feed4863956ab633fdd493>