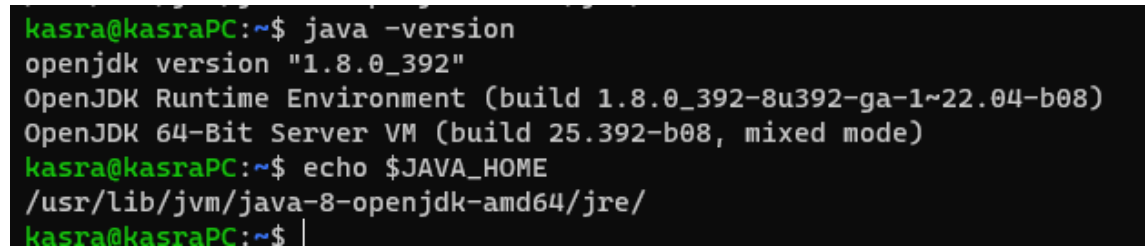


Download and install java

A terminal window with a black background and green text. The prompt is 'kasra@kasraPC:~\$'. The first command is 'java -version', which outputs 'openjdk version "1.8.0_392"', 'OpenJDK Runtime Environment (build 1.8.0_392-8u392-ga-1~22.04-b08)', and 'OpenJDK 64-Bit Server VM (build 25.392-b08, mixed mode)'. The second command is 'echo \$JAVA_HOME', which outputs '/usr/lib/jvm/java-8-openjdk-amd64/jre/'. The prompt is now 'kasra@kasraPC:~\$' with a cursor.

```
kasra@kasraPC:~$ java -version
openjdk version "1.8.0_392"
OpenJDK Runtime Environment (build 1.8.0_392-8u392-ga-1~22.04-b08)
OpenJDK 64-Bit Server VM (build 25.392-b08, mixed mode)
kasra@kasraPC:~$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64/jre/
kasra@kasraPC:~$
```

Figure 1: alt text

also to set the JAVA_HOME we used the following command:

```
export JAVA_HOME=$(readlink -f /usr/bin/java | sed "s:bin/java::")
```

Download and install Scala

also, we could use this command to set the SCALA_HOME:

```
export SCALA_HOME=$(readlink -f /usr/bin/scala | sed "s:bin/scala::")
```

since `which scala` and `which scalac` works fine (they have links in `/usr/bin`), we don't need to add the SCALA_HOME to the PATH but anyway, we can set it using the following command:

```
export PATH=$PATH:$SCALA_HOME/bin
```

Download and install Hadoop

in the above commands, we downloaded the hadoop 2.7 from the official website and we can extract it to `~/hadoop` directory using the following command:

```
tar -xf hadoop-2.7.7.tar.gz
rm -rf ~/hadoop
mkdir ~/hadoop -p
mv hadoop-2.7.7/* ~/hadoop
rmdir hadoop-2.7.7
```

to set the HADOOP_HOME and adding it to the PATH we used the following commands:

```
export HADOOP_HOME=~/hadoop
export PATH=~/hadoop/bin:$PATH
```

finally, to verify the installation:

```

kasra@kasraPC:~$ namei $(which scala)
f: /usr/bin/scala
d /
d usr
d bin
l scala -> /etc/alternatives/scala
d /
d etc
d alternatives
l scala -> /usr/share/scala-2.11/bin/scala
d /
d usr
d share
d scala-2.11
d bin
- scala

kasra@kasraPC:~$ export SCALA_HOME="/usr/share/scala-2.11"
kasra@kasraPC:~$ echo $SCALA_HOME
/usr/share/scala-2.11
kasra@kasraPC:~$ scala -version
Scala code runner version 2.11.12 -- Copyright 2002-2017, LAMP/EPFL
kasra@kasraPC:~$ |

```

Figure 2: alt text

```

kasra@kasraPC:~/Downloads$ wget https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
--2024-02-26 21:56:09-- https://archive.apache.org/dist/hadoop/common/hadoop-2.7.7/hadoop-2.7.7.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.180.204.189, 2a01:4f9:1a:a884::2
Connecting to archive.apache.org (archive.apache.org)[65.180.204.189]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 218728521 (209M) [application/x-gzip]
Saving to: 'hadoop-2.7.7.tar.gz'

hadoop-2.7.7.tar.gz          100%[=====] 208.59M  7.54MB/s   in 32s

2024-02-26 21:56:42 (6.46 MB/s) - 'hadoop-2.7.7.tar.gz' saved [218728521/218728521]

```

Figure 3: alt text

```

kásra@kásraPC:~$ hadoop version
Hadoop 2.7.7
Subversion Unknown -r c1aad84bd27cd79c3d1a7dd58202a8c3ee1ed3ac
Compiled by stevel on 2018-07-18T22:47Z
Compiled with protoc 2.5.0
From source with checksum 792e15d20b12c74bd6f19a1fb886490
This command was run using /home/kásra/.hadoop/share/hadoop/common/hadoop-common-2.7.7.jar
kásra@kásraPC:~$ |

```

Download and install Spark

download the spark 3.2.1:

```

kaszeg@kaszegPC:~/Downloads$ wget https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-hadoop2.7.tgz
--2024-02-26 22:36:21-- https://archive.apache.org/dist/spark/spark-3.2.1/spark-3.2.1-bin-hadoop2.7.tgz
Resolving archive.apache.org (archive.apache.org)... 65.188.204.189, 2a01:499:1:a800::2
Connecting to archive.apache.org (archive.apache.org)|65.188.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 272637746 (260M) [application/x-gzip]
Saving to: 'spark-3.2.1-bin-hadoop2.7.tgz.'

spark-3.2.1-bin-hadoop2.7.tgz      100%[=====] 260.01M  8.62MB/s   in 40s
2024-02-26 22:37:01 (6.53 MB/s) - 'spark-3.2.1-bin-hadoop2.7.tgz.' saved [272637746/272637746]

kaszeg@kaszegPC:~/Downloads$

```

extract the spark to ~/.spark:

```

kasta@kastaPC:~/Downloads$ tar xf spark-3.2.1-bin-hadoop2.7.tgz
kasta@kastaPC:~/Downloads$ LS
ls: command not found
kasta@kastaPC:~/Downloads$ ls
apache-hive-3.13.3-bin.tar.gz  hadoop-2.7.7.tar.gz  hadoop-3.3.4.tar.gz  hbase-2.5.6-bin.tar.gz  quarto-1.3.458-linux-amd64.deb  spark-3.2.1-bin-hadoop2.7  spark-3.2.1-bin-hadoop2.7.tgz
kasta@kastaPC:~/Downloads$ ls spark-3.2.1-bin-hadoop2.7
LICENSE NOTICE  R  README.md  RELEASE  bin  conf  data  examples  jars  kubernetes  licenses  python  sbin  yarn
kasta@kastaPC:~/Downloads$ cd ..
kasta@kastaPC:~$ mkdir spark
kasta@kastaPC:~$ mv Downloads/spark-3.2.1-bin-hadoop2.7/* .spark/
kasta@kastaPC:~$ ls .spark/
LICENSE NOTICE  R  README.md  RELEASE  bin  conf  data  examples  jars  kubernetes  licenses  python  sbin  yarn

```

set the `SPARK_HOME` and add it to the `PATH`:

```
export SPARK_HOME=~/.spark
```

```
export PATH=~/.spark/bin:$PATH
```

finally, to verify the installation:

```

kasra@kasraPC:~$ spark-shell
24/02/27 02:19:26 WARN Utils: Your hostname, kasraPC resolves to a loopback address: 127.0.1.1; using 172.18.241.184 instead (on interface eth0)
24/02/27 02:19:26 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/02/27 02:19:31 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://172.18.241.184:4040
Spark context available as 'sc' (master = local[*], app id = local-1708996772099).
Spark session available as 'spark'.
Welcome to

  /---/  _--  /---/  /---/
 _\V_ _\V_ _\V_ _\V_ _\V_
/---/  /---/  /---/  /---/  version 3.2.1
  /_/_

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 1.8.0_392)
Type in expressions to have them evaluated.
Type :help for more information.

scala> |

```

Run the WordCount example in Apache Hadoop

to make a jar file using the codes, we need to compile the codes using the following command:

```
javac -classpath $(hadoop classpath) *.java
```

to verify the compilation:

```
ksara@ksaraPC:~/projects/uni_lu_big_data/hw0$ javac -classpath $(hadoop classpath) *.java
ksara@ksaraPC:~/projects/uni_lu_big_data/hw0$ ls *.class
HadoopWordCount$Map.class      HadoopWordCount.class      'HadoopWordPairs$Reduce.class'  'HadoopWordStripes$Map.class'  HadoopWordStripes.class
HadoopWordCount$Reduce.class   'HadoopWordPairs$Map.class'  HadoopWordPairs.class          'HadoopWordStripes$Reduce.class'
```

finally, to make a jar file:

```
jar cf wc.jar *.class
```

in this point we can delete the `.class` files to keep the directory clean:

```
rm *.class
```

Now we can use this command to run the HadoopWordCount:

```
hadoop jar HadoopWordCount.jar HadoopWordCount ~/data/wikipedia/enwiki-articles/AA/ output
```

to verify the output:

```

kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ ls output/
_SUCCESS part-r-000000
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ head output/part-r-000000
6069
! 27
!" 1
!") 1
!, 2
!Velocity 1
!Xóð 1
!nu" 1
!style="background-color:#E9E9E9" 11
" 162
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ tail output/part-r-000000
黑皮公禱書 1
나라가 1
산다, 1
식목일) 1
죽어야 1
first 1
first 1
flavoenzyme 1
4#00000), 1
. 1
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$

```

for the HadoopWordPairs:

```
rm -rf output
```

```
hadoop jar HadoopWordCount.jar HadoopWordPairs ~/data/wikipedia/enwiki-articles/AA/ output
```

to verify the output:

```
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ head output/part-r-000000 -n 20
!)".:The      1
!:In      1
!,:&,      1
!,:so      1
!:!      1
!:"m"\ "n"      1
!:"n"      1
!:0      2
!:1      2
!:2      2
!:3      2
!:4      2
!:BOOT      1
!:Formal      1
!:No      1
!:Notes      1
!:Rotation      1
!:Total      1
!:Translation      1
!:colspan="2"      1
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ |
```

for the HadoopWordStripes:

```
rm -rf output
```

```
hadoop jar HadoopWordCount.jar HadoopWordStripes ~/data/wikipedia/enwiki-articles/AA/ output
```

to verify the output:

```
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ head output/part-r-000000
org.apache.hadoop.io.MapWritable@ba4258d7
! org.apache.hadoop.io.MapWritable@fdb6304d
!" org.apache.hadoop.io.MapWritable@755cd0a
!)". org.apache.hadoop.io.MapWritable@7882cd2
!,: org.apache.hadoop.io.MapWritable@df2087e6
!Velocity org.apache.hadoop.io.MapWritable@1
!Xóó org.apache.hadoop.io.MapWritable@96f91a7b
!nu" org.apache.hadoop.io.MapWritable@74d8d9ae
!style="background-color:#E9E9E9" org.apache.hadoop.io.MapWritable@a2a3af7f
" org.apache.hadoop.io.MapWritable@b45305f2
kasra@kasraPC:~/projects/uni_lu_big_data/hw0$ |
```

Run the WordCount Example in Apache Spark

to run the `SparkWordCount` we need to install sbt using here and then we need a `build.sbt` file as follows:

```
name := "Simple Project"
```

```
version := "1.0"
```

```
scalaVersion := "2.12.18"
```

```
libraryDependencies += "org.apache.spark" %% "spark-sql" % "3.5.0"
```

we need to make a directory structure as follows:

```
$ find src build.*
```

```
src
```

```
src/main
```

```
src/main/scala
```

```
src/main/scala/SparkWordCount.scala
```

```
build.sbt
```

and then we can use the following command to run the `SparkWordCount`:

```
sbt package
```

```
spark-submit --class SparkWordCount target/scala-2.12/simple-project_2.12-1.0.jar ~/data/wikipedia/e
```

to verify the output:

```

kasra@kasraPC:~/projects/uni_lu_big_data/hw0/scala$ ls output/
_SUCCESS      part-00005    part-00011    part-00017    part-00023    part-00029    part-00035    part-6
part-00000     part-00006    part-00012    part-00018    part-00024    part-00030    part-00036    part-6
part-00001     part-00007    part-00013    part-00019    part-00025    part-00031    part-00037    part-6
part-00002     part-00008    part-00014    part-00020    part-00026    part-00032    part-00038    part-6
part-00003     part-00009    part-00015    part-00021    part-00027    part-00033    part-00039    part-6
part-00004     part-00010    part-00016    part-00022    part-00028    part-00034    part-00040    part-6
kasra@kasraPC:~/projects/uni_lu_big_data/hw0/scala$ head output/part-00000 -n 20
(stacks).,1)
(Trek".,2)
(Shapes.,1)
(Kumar's,1)
(bone,271)
(汉堡包 / 漢堡包,1)
(id="4527",1)
(Wakayama,,1)
(CAD-CAM,1)
((16th-century,1)
(Kurau,1)
(screenwriters,,2)
("Dead",,1)
(Coombs;,1)
(Boat,15)
(D66,,1)
(Bern;,1)
(linotype,2)
(Genoese,25)
(Dailey,2)
kasra@kasraPC:~/projects/uni_lu_big_data/hw0/scala$ |

```