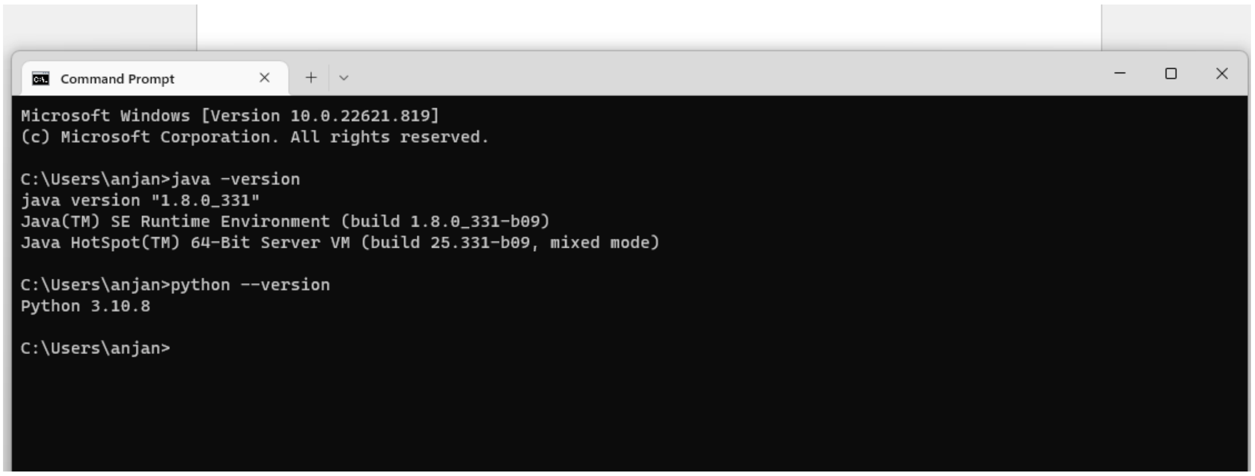
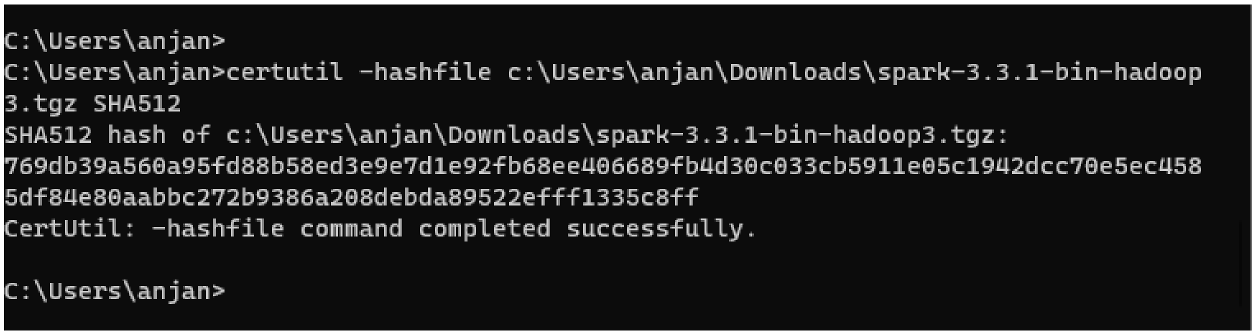
LAB4: Data Processing:

Before installing spark, we need to install java and python. I have installed below are versions which I have installed.



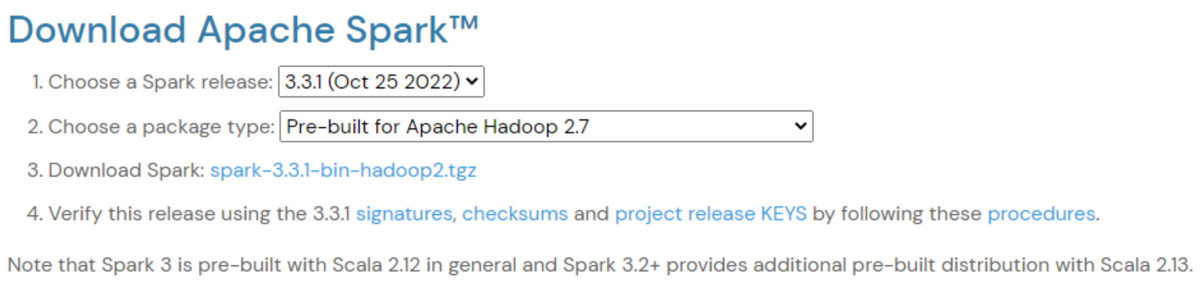
```
Microsoft Windows [Version 10.0.22621.819]
(c) Microsoft Corporation. All rights reserved.

C:\Users\anjan>java -version
java version "1.8.0_331"
Java(TM) SE Runtime Environment (build 1.8.0_331-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.331-b09, mixed mode)

C:\Users\anjan>python --version
Python 3.10.8

C:\Users\anjan>
```

After that we need to check the hashfile of spark



```
C:\Users\anjan>
C:\Users\anjan>certutil -hashfile c:\Users\anjan\Downloads\spark-3.3.1-bin-hadoop
3.tgz SHA512
SHA512 hash of c:\Users\anjan\Downloads\spark-3.3.1-bin-hadoop3.tgz:
769db39a560a95fd88b58ed3e9e7d1e92fb68ee406689fb4d30c033cb5911e05c1942dcc70e5ec458
5df84e80aabbc272b9386a208debda89522efff1335c8ff
CertUtil: -hashfile command completed successfully.

C:\Users\anjan>
```

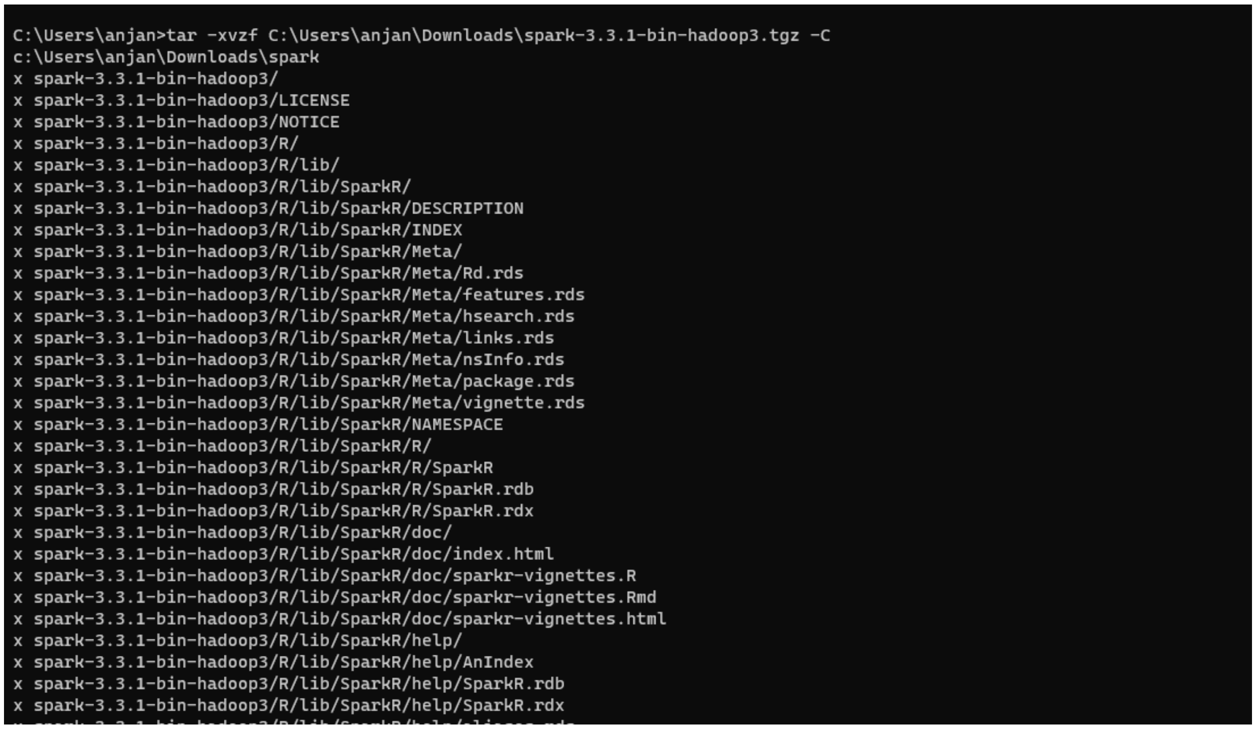
I have downloaded the spark-3.3.1-bin-hadoop3 from the below link and unzipped in c folder

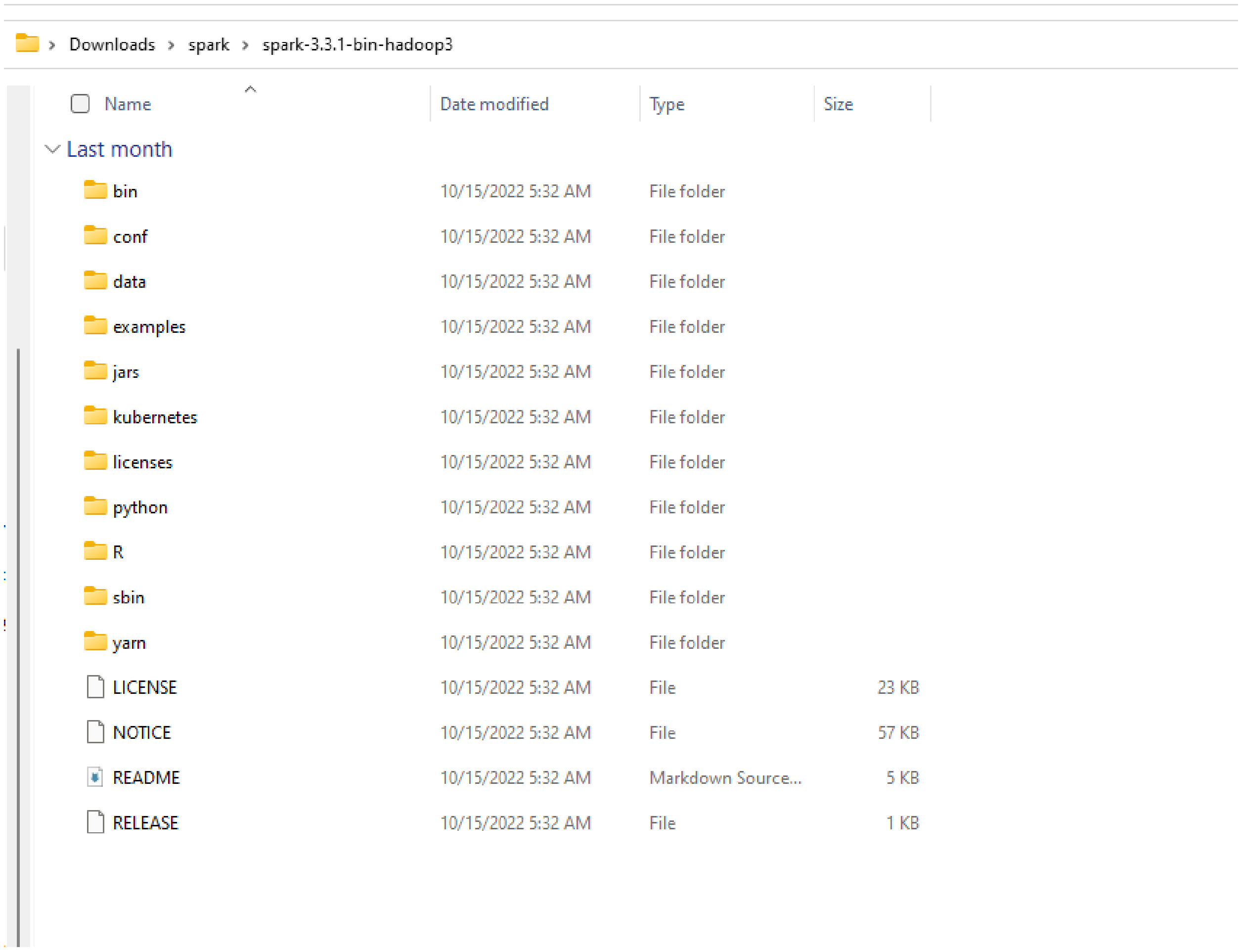


Download Apache Spark™

1. Choose a Spark release: 3.3.1 (Oct 25 2022)
2. Choose a package type: Pre-built for Apache Hadoop 2.7
3. Download Spark: spark-3.3.1-bin-hadoop2.tgz
4. Verify this release using the 3.3.1 signatures, checksums and project release KEYS by following these procedures.

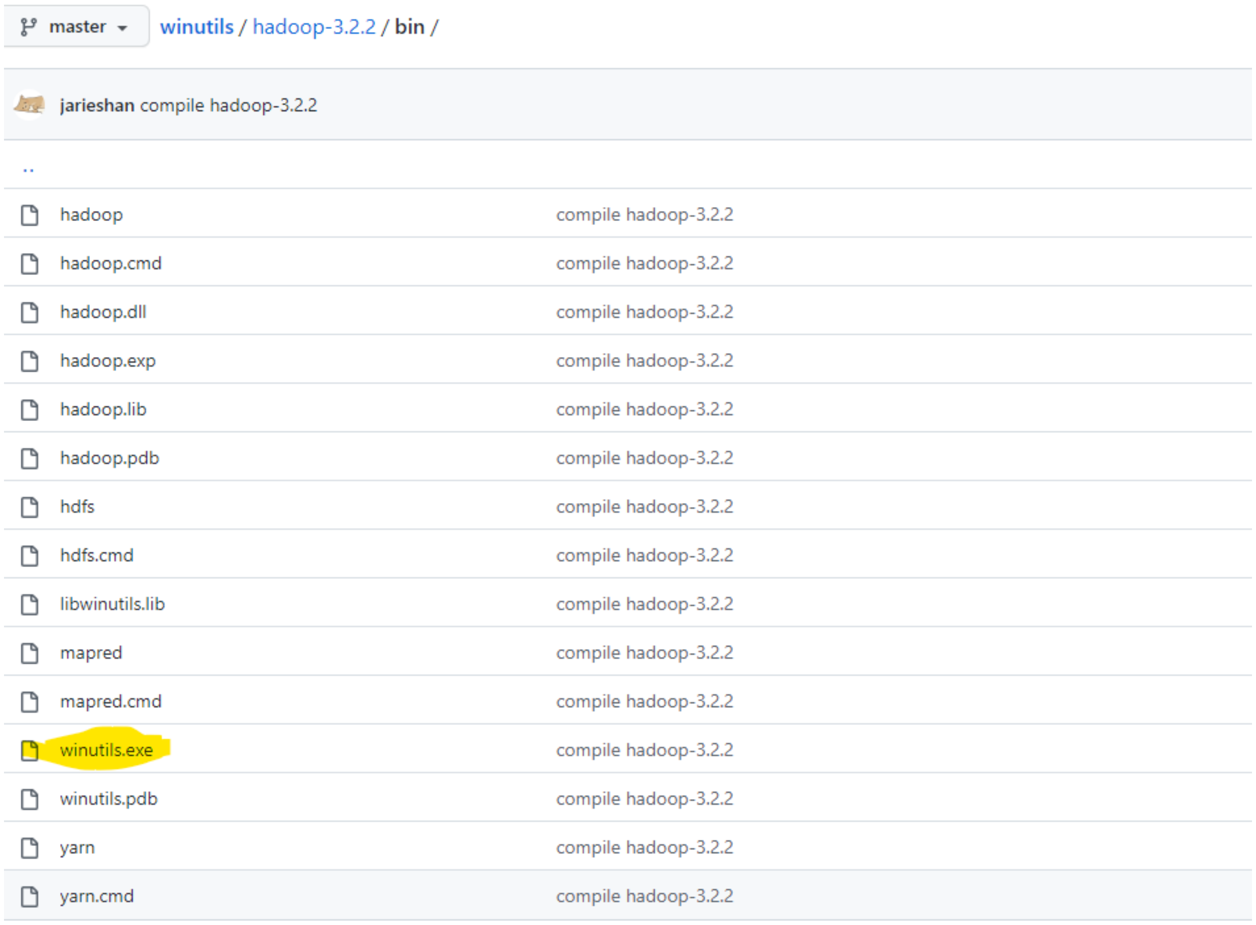
Note that Spark 3 is pre-built with Scala 2.12 in general and Spark 3.2+ provides additional pre-built distribution with Scala 2.13.


```
C:\Users\anjan>tar -xvzf C:\Users\anjan\Downloads\spark-3.3.1-bin-hadoop3.tgz -C
c:\Users\anjan\Downloads\spark
x spark-3.3.1-bin-hadoop3/
x spark-3.3.1-bin-hadoop3/LICENSE
x spark-3.3.1-bin-hadoop3/NOTICE
x spark-3.3.1-bin-hadoop3/R/
x spark-3.3.1-bin-hadoop3/R/lib/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/DESCRIPTION
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/INDEX
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/Rd.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/features.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/hsearch.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/links.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/nsInfo.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/package.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/Meta/vignette.rds
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/NAMESPACE
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/R/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/R/SparkR
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/R/SparkR.rdb
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/R/SparkR.rdx
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/doc/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/doc/index.html
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/doc/sparkr-vignettes.R
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/doc/sparkr-vignettes.Rmd
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/doc/sparkr-vignettes.html
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/help/
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/help/AnIndex
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/help/SparkR.rdb
x spark-3.3.1-bin-hadoop3/R/lib/SparkR/help/SparkR.rdx
```


Downloads > spark > spark-3.3.1-bin-hadoop3

Name	Date modified	Type	Size
Last month			
bin	10/15/2022 5:32 AM	File folder	
conf	10/15/2022 5:32 AM	File folder	
data	10/15/2022 5:32 AM	File folder	
examples	10/15/2022 5:32 AM	File folder	
jars	10/15/2022 5:32 AM	File folder	
kubernetes	10/15/2022 5:32 AM	File folder	
licenses	10/15/2022 5:32 AM	File folder	
python	10/15/2022 5:32 AM	File folder	
R	10/15/2022 5:32 AM	File folder	
sbin	10/15/2022 5:32 AM	File folder	
yarn	10/15/2022 5:32 AM	File folder	
LICENSE	10/15/2022 5:32 AM	File	23 KB
NOTICE	10/15/2022 5:32 AM	File	57 KB
README	10/15/2022 5:32 AM	Markdown Source...	5 KB
RELEASE	10/15/2022 5:32 AM	File	1 KB

From the Hadoop-3.2.2 link I have copied the winutils.exe and placed them in separate folder called Hadoop which is in c.



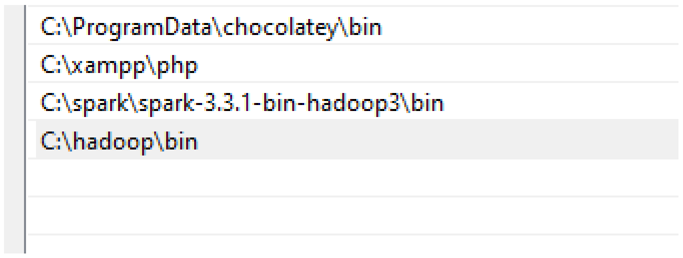
master / winutils / hadoop-3.2.2 / bin /

jarieshan compile hadoop-3.2.2

..

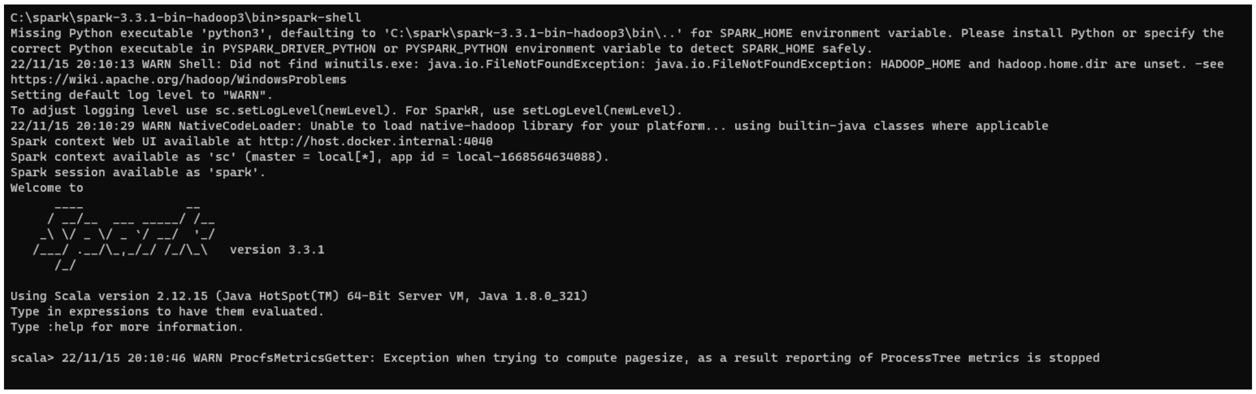
hadoop	compile hadoop-3.2.2
hadoop.cmd	compile hadoop-3.2.2
hadoop.dll	compile hadoop-3.2.2
hadoop.exp	compile hadoop-3.2.2
hadoop.lib	compile hadoop-3.2.2
hadoop.pdb	compile hadoop-3.2.2
hdfs	compile hadoop-3.2.2
hdfs.cmd	compile hadoop-3.2.2
libwinutils.lib	compile hadoop-3.2.2
mapred	compile hadoop-3.2.2
mapred.cmd	compile hadoop-3.2.2
winutils.exe	compile hadoop-3.2.2
winutils.pdb	compile hadoop-3.2.2
yarn	compile hadoop-3.2.2
yarn.cmd	compile hadoop-3.2.2

In environmental variables added the spark and hadoop

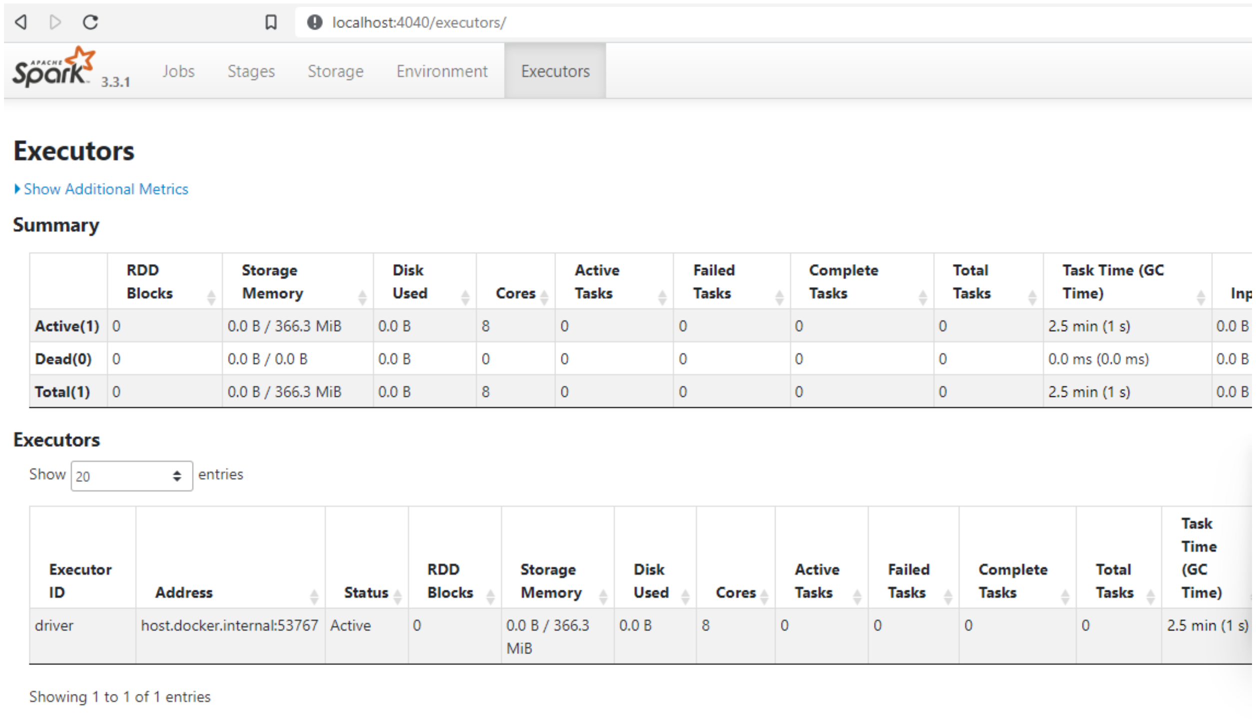


```
C:\ProgramData\chocolatey\bin
C:\xampp\php
C:\spark\spark-3.3.1-bin-hadoop3\bin
C:\hadoop\bin
```

By running spark-shell



```
C:\spark\spark-3.3.1-bin-hadoop3\bin>spark-shell
Missing Python executable 'python3', defaulting to 'C:\spark\spark-3.3.1-bin-hadoop3\bin\..' for SPARK_HOME environment variable. Please install Python or specify the
correct Python executable in PYSPARK_DRIVER_PYTHON or PYSPARK_PYTHON environment variable to detect SPARK_HOME safely.
22/11/15 20:10:13 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.home.dir are unset. -see
https://wiki.apache.org/hadoop/WindowsProblems
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/15 20:10:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://host.docker.internal:4040
Spark context available as 'sc' (master = local[*], app id = local-1668564634088).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_321)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 22/11/15 20:10:46 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessTree metrics is stopped
```



localhost:4040/executors/

Executors

▸ Show Additional Metrics

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Inp
Active(1)	0	0.0 B / 366.3 MiB	0.0 B	8	0	0	0	0	2.5 min (1 s)	0.0 B
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B
Total(1)	0	0.0 B / 366.3 MiB	0.0 B	8	0	0	0	0	2.5 min (1 s)	0.0 B

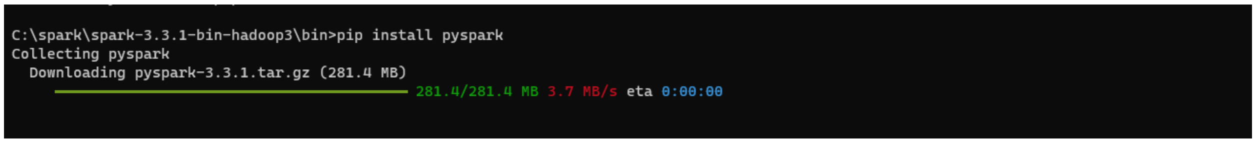
Executors

Show 20 entries

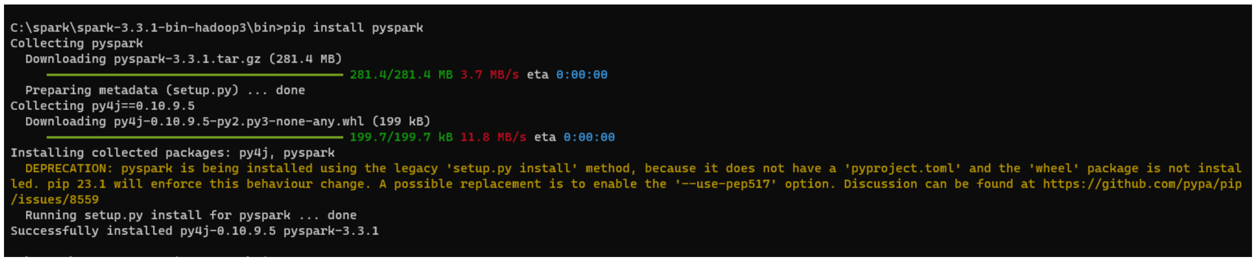
Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)
driver	host.docker.internal:53767	Active	0	0.0 B / 366.3 MiB	0.0 B	8	0	0	0	0	2.5 min (1 s)

Showing 1 to 1 of 1 entries

Now to install pyspark using pip command

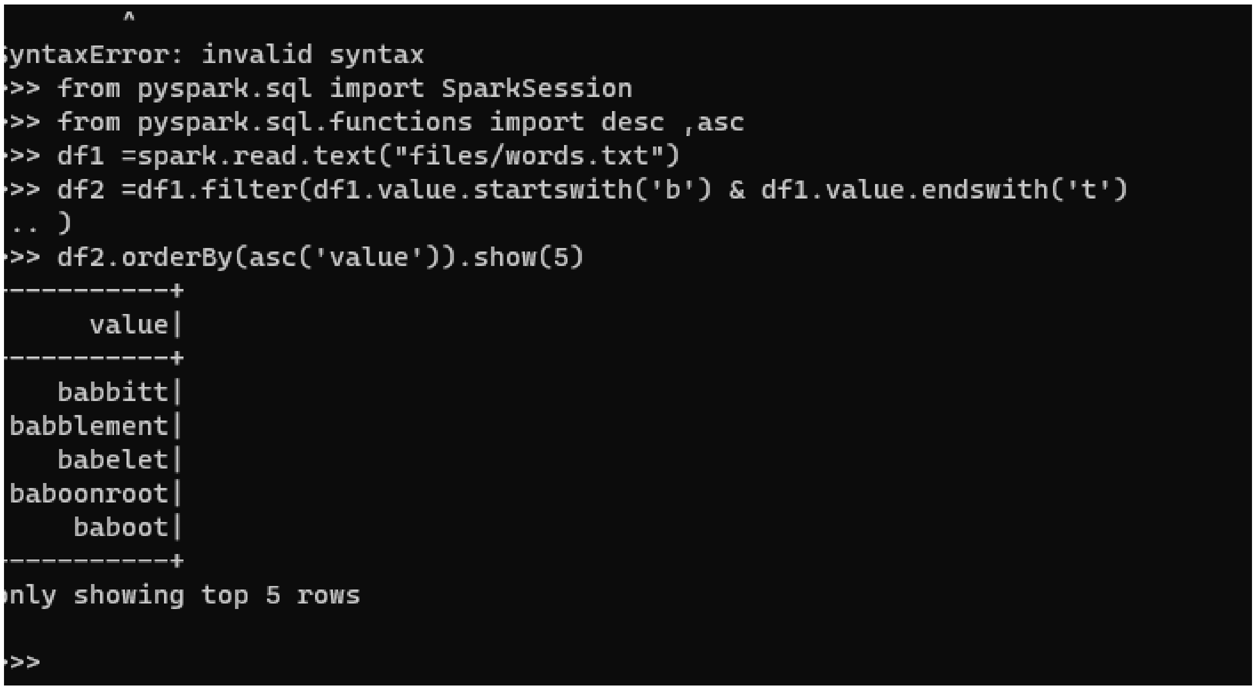


```
C:\spark\spark-3.3.1-bin-hadoop3\bin>pip install pyspark
Collecting pyspark
  Downloading pyspark-3.3.1.tar.gz (281.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.4/281.4 MB 3.7 MB/s eta 0:00:00
```



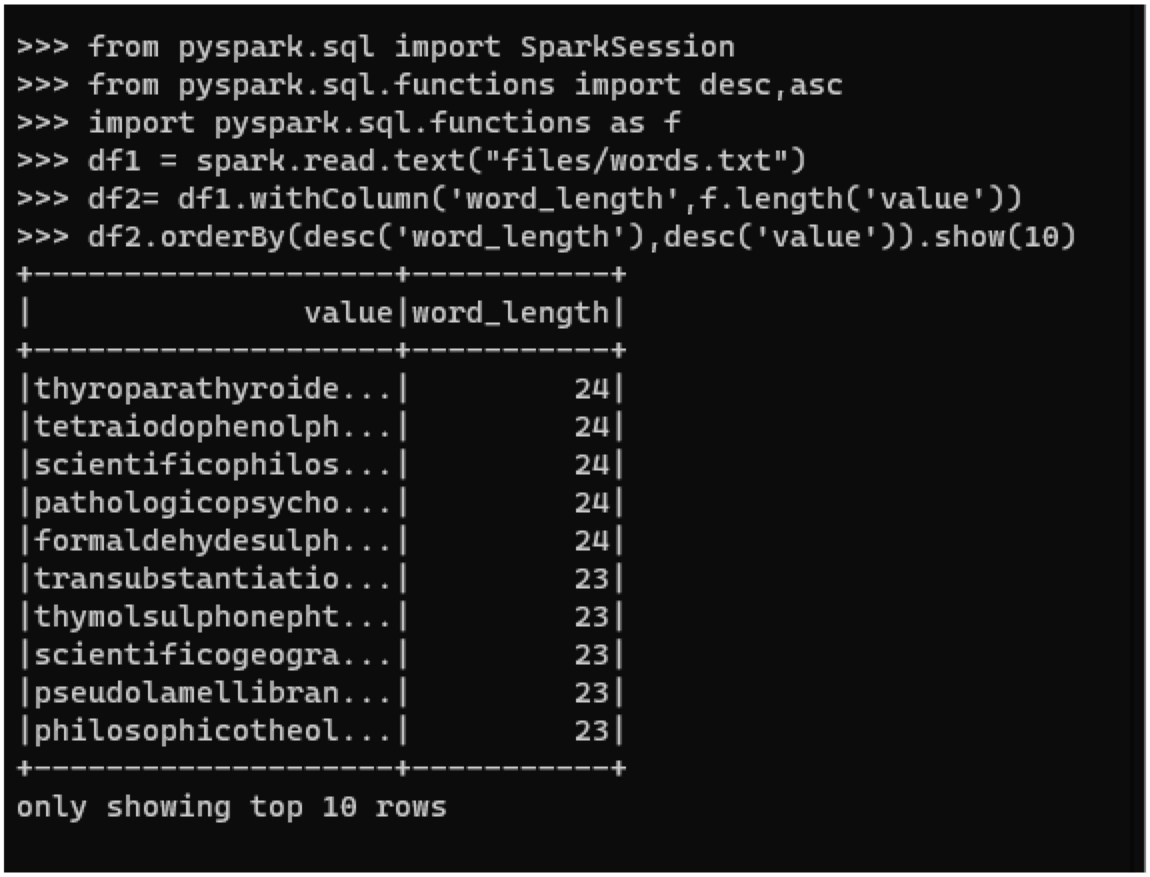
```
C:\spark\spark-3.3.1-bin-hadoop3\bin>pip install pyspark
Collecting pyspark
  Downloading pyspark-3.3.1.tar.gz (281.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 281.4/281.4 MB 3.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting py4j==0.10.9.5
  Downloading py4j-0.10.9.5-py2.py3-none-any.whl (199 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.7/199.7 kB 11.8 MB/s eta 0:00:00
Installing collected packages: py4j, pyspark
  DEPRECATION: pyspark is being installed using the legacy 'setup.py install' method, because it does not have a 'pyproject.toml' and the 'wheel' package is not instal
led. pip 23.1 will enforce this behaviour change. A possible replacement is to enable the '--use-pep517' option. Discussion can be found at https://github.com/pypa/pip
/issues/8559
  Running setup.py install for pyspark ... done
Successfully installed py4j-0.10.9.5 pyspark-3.3.1
```

1.1)To get the top 5 which starts with b and ends with t. these are top 5 rows



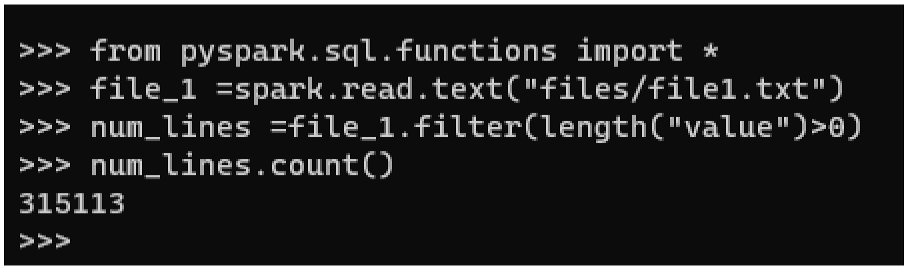
```
       ^
yntaxError: invalid syntax
>> from pyspark.sql import SparkSession
>> from pyspark.sql.functions import desc ,asc
>> df1 =spark.read.text("files/words.txt")
>> df2 =df1.filter(df1.value.startswith('b') & df1.value.endswith('t')
.. )
>> df2.orderBy(asc('value')).show(5)
----------+
     value|
----------+
   babbitt|
babblement|
   babelet|
baboonroot|
    baboot|
----------+
nly showing top 5 rows

>>
```

1.2)The code to get the top 10 word .I have used the below code



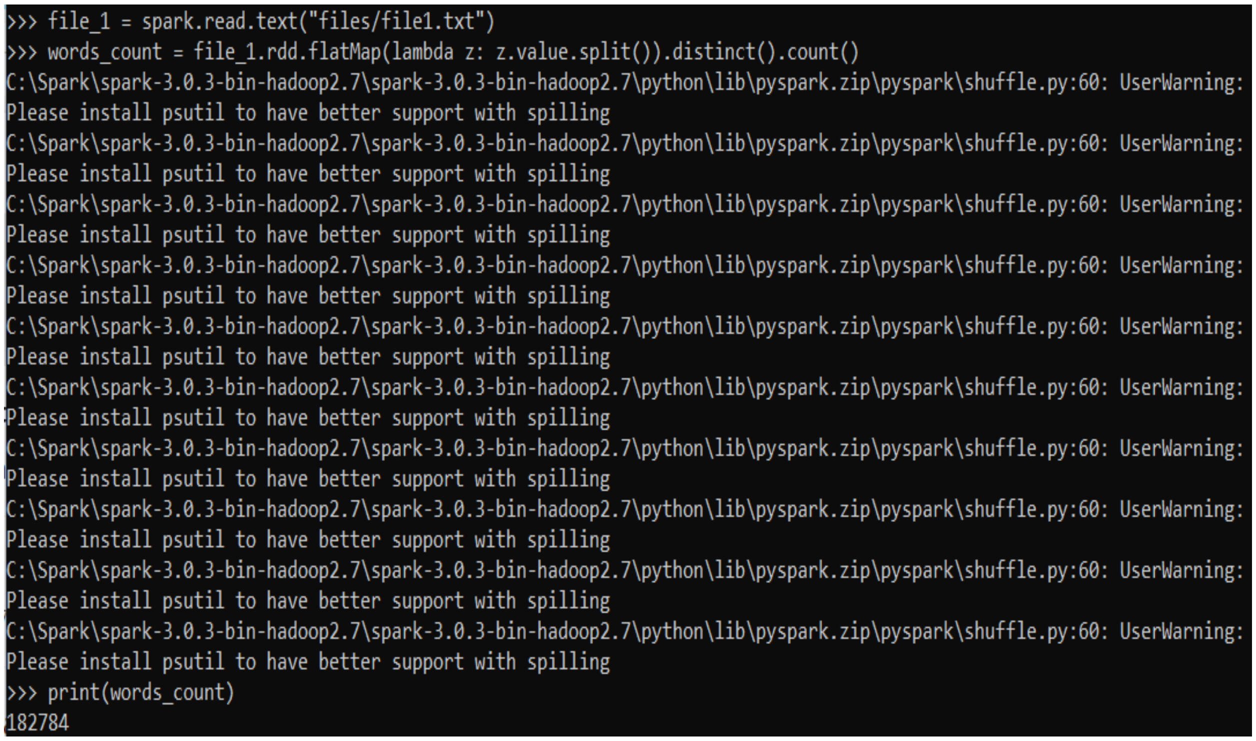
```
>>> from pyspark.sql import SparkSession
>>> from pyspark.sql.functions import desc,asc
>>> import pyspark.sql.functions as f
>>> df1 = spark.read.text("files/words.txt")
>>> df2= df1.withColumn('word_length',f.length('value'))
>>> df2.orderBy(desc('word_length'),desc('value')).show(10)
+--------------------+-----------+
|               value|word_length|
+--------------------+-----------+
|thyroparathyroide...|         24|
|tetraiodophenolph...|         24|
|scientificophilos...|         24|
|pathologicopsycho...|         24|
|formaldehydesulph...|         24|
|transubstantiatio...|         23|
|thymolsulphonepht...|         23|
|scientificogeogra...|         23|
|pseudolamellibran...|         23|
|philosophicotheol...|         23|
+--------------------+-----------+
only showing top 10 rows
```

1.3)To get the number of line from file 1.txt this is the below code:



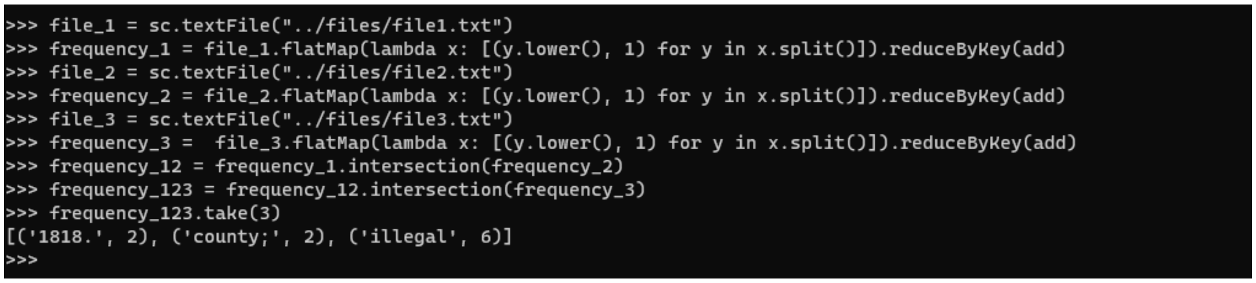
```
>>> from pyspark.sql.functions import *
>>> file_1 =spark.read.text("files/file1.txt")
>>> num_lines =file_1.filter(length("value")>0)
>>> num_lines.count()
315113
>>>
```

To get the word count below is the code



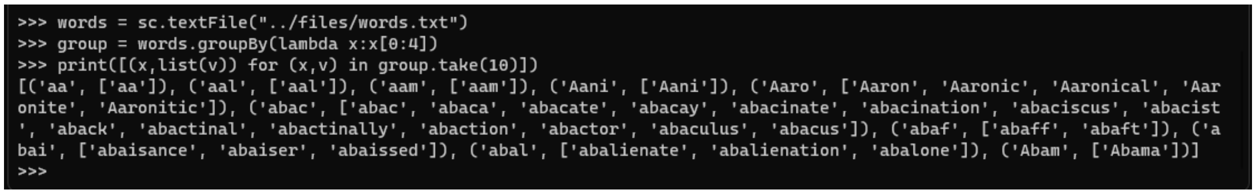
```
>>> file_1 = spark.read.text("files/file1.txt")
>>> words_count = file_1.rdd.flatMap(lambda z: z.value.split()).distinct().count()
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
C:\Spark\spark-3.0.3-bin-hadoop2.7\spark-3.0.3-bin-hadoop2.7\python\lib\pyspark.zip\pyspark\shuffle.py:60: UserWarning:
Please install psutil to have better support with spilling
>>> print(words_count)
182784
```

1.4)To get common word in all the 3 files below is the code using flatMap



```
>>> file_1 = sc.textFile("../files/file1.txt")
>>> frequency_1 = file_1.flatMap(lambda x: [(y.lower(), 1) for y in x.split()]).reduceByKey(add)
>>> file_2 = sc.textFile("../files/file2.txt")
>>> frequency_2 = file_2.flatMap(lambda x: [(y.lower(), 1) for y in x.split()]).reduceByKey(add)
>>> file_3 = sc.textFile("../files/file3.txt")
>>> frequency_3 =  file_3.flatMap(lambda x: [(y.lower(), 1) for y in x.split()]).reduceByKey(add)
>>> frequency_12 = frequency_1.intersection(frequency_2)
>>> frequency_123 = frequency_12.intersection(frequency_3)
>>> frequency_123.take(3)
[('1818.', 2), ('county;', 2), ('illegal', 6)]
>>>
```

1.5)Group words for first 4



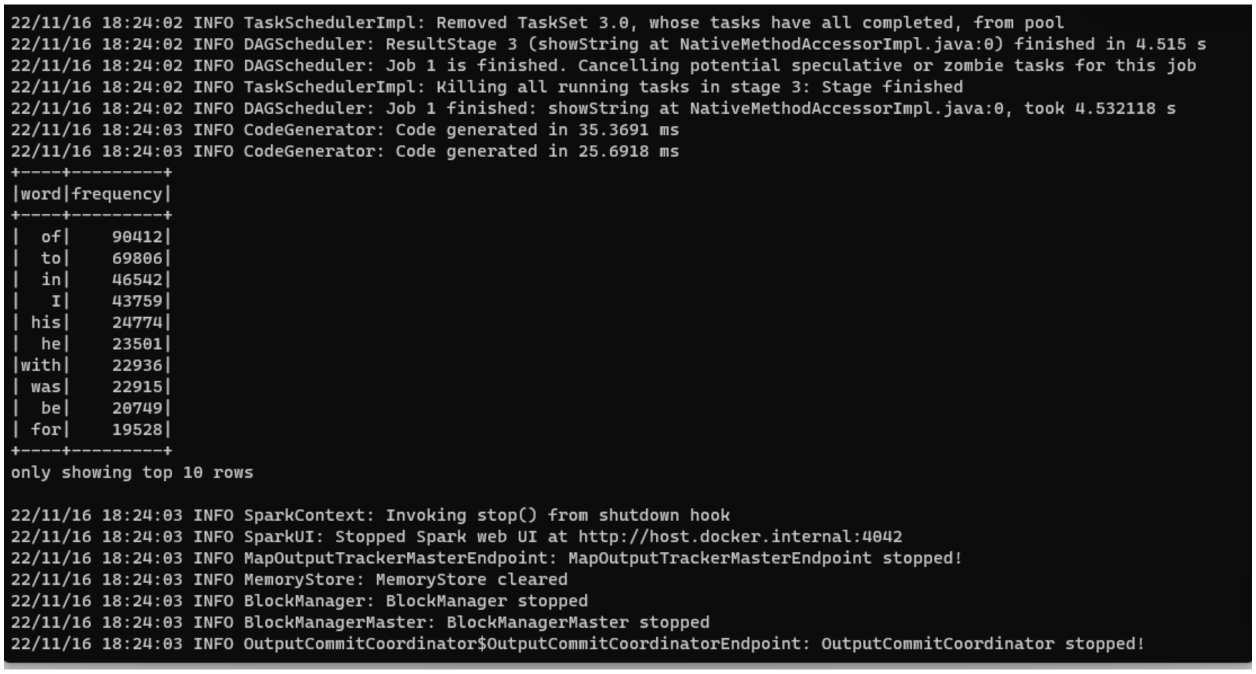
```
>>> words = sc.textFile("../files/words.txt")
>>> group = words.groupBy(lambda x:x[0:4])
>>> print([(x,list(v)) for (x,v) in group.take(10)])
[('aa', ['aa']), ('aal', ['aal']), ('aam', ['aam']), ('Aani', ['Aani']), ('Aaro', ['Aaron', 'Aaronic', 'Aaronical', 'Aar
onite', 'Aaronitic']), ('abac', ['abac', 'abaca', 'abacate', 'abacay', 'abacinate', 'abacination', 'abaciscus', 'abacist
', 'aback', 'abactinal', 'abactinally', 'abaction', 'abactor', 'abaculus', 'abacus']), ('abaf', ['abaff', 'abaft']), ('a
bai', ['abaisance', 'abaiser', 'abaissed']), ('abal', ['abalienate', 'abalienation', 'abalone']), ('Abam', ['Abama'])]
>>>
```

2)Working with standalone cluster:

In the standalone cluster I have created program to count the given words frequency and executed using pyspark_submit command

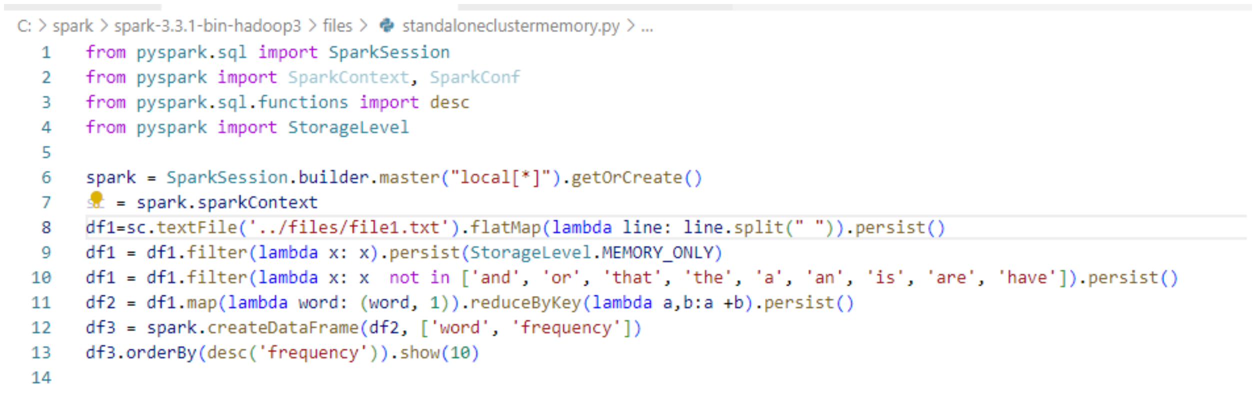
When I have used the disk_only it executed with in time of 4.532118 sec

```
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
from pyspark.sql.functions import desc
from pyspark import StorageLevel

spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
df1 = sc.textFile('../files/file1.txt').flatMap(lambda line: line.split("
")).persist()
df1 = df1.filter(lambda x: x).persist(StorageLevel.DISK_ONLY)
df1 = df1.filter(lambda x: x  not in ['and', 'or', 'that', 'the', 'a',
'an', 'is', 'are', 'have']).persist()
df2 = df1.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a
+b).persist()
df3 = spark.createDataFrame(df2, ['word', 'frequency'])
df3.orderBy(desc('frequency')).show(10)
```

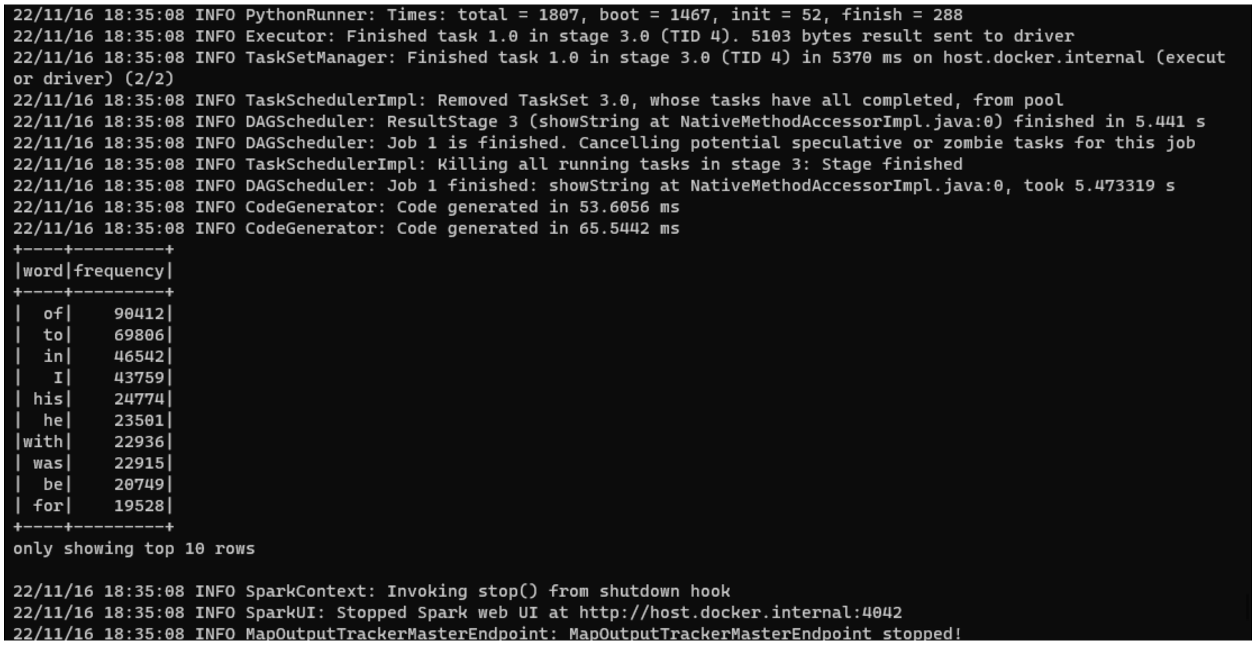


```
22/11/16 18:24:02 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
22/11/16 18:24:02 INFO DAGScheduler: ResultStage 3 (showString at NativeMethodAccessorImpl.java:0) finished in 4.515 s
22/11/16 18:24:02 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/16 18:24:02 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 18:24:02 INFO DAGScheduler: Job 1 finished: showString at NativeMethodAccessorImpl.java:0, took 4.532118 s
22/11/16 18:24:03 INFO CodeGenerator: Code generated in 35.3691 ms
22/11/16 18:24:03 INFO CodeGenerator: Code generated in 25.6918 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows

22/11/16 18:24:03 INFO SparkContext: Invoking stop() from shutdown hook
22/11/16 18:24:03 INFO SparkUI: Stopped Spark web UI at http://host.docker.internal:4042
22/11/16 18:24:03 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
22/11/16 18:24:03 INFO MemoryStore: MemoryStore cleared
22/11/16 18:24:03 INFO BlockManager: BlockManager stopped
22/11/16 18:24:03 INFO BlockManagerMaster: BlockManagerMaster stopped
22/11/16 18:24:03 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
```

But while using the MEMORY_ONLY it is executing with run time of 5.473319s.

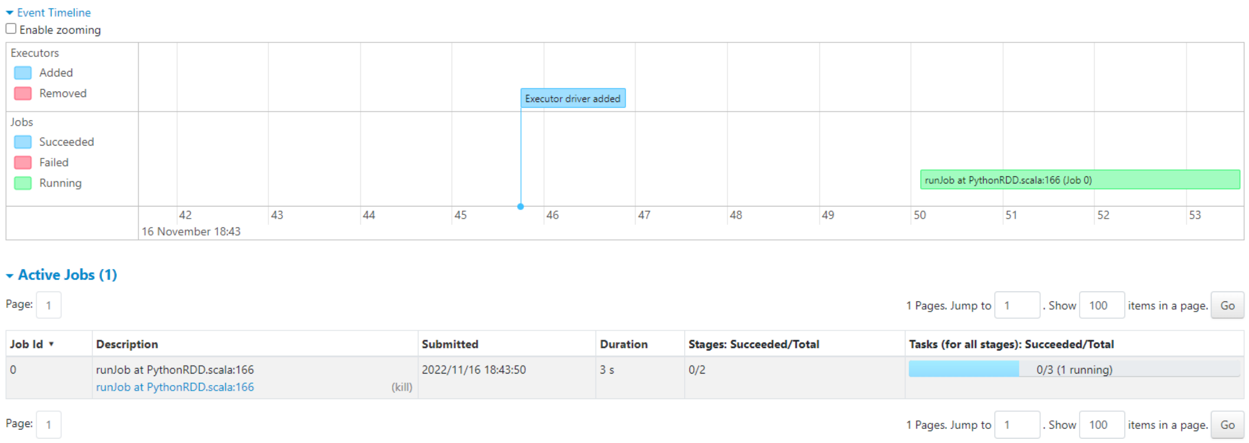


```
C: > spark > spark-3.3.1-bin-hadoop3 > files > standaloneclustermemory.py > ...
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
from pyspark.sql.functions import desc
from pyspark import StorageLevel

spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
df1=sc.textFile('../files/file1.txt').flatMap(lambda line: line.split(" ")).persist()
df1 = df1.filter(lambda x: x).persist(StorageLevel.MEMORY_ONLY)
df1 = df1.filter(lambda x: x  not in ['and', 'or', 'that', 'the', 'a', 'an', 'is', 'are', 'have']).persist()
df2 = df1.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b).persist()
df3 = spark.createDataFrame(df2, ['word', 'frequency'])
df3.orderBy(desc('frequency')).show(10)
```



```
22/11/16 18:35:08 INFO PythonRunner: Times: total = 1807, boot = 1467, init = 52, finish = 288
22/11/16 18:35:08 INFO Executor: Finished task 1.0 in stage 3.0 (TID 4). 5103 bytes result sent to driver
22/11/16 18:35:08 INFO TaskSetManager: Finished task 1.0 in stage 3.0 (TID 4) in 5370 ms on host.docker.internal (execut
or driver) (2/2)
22/11/16 18:35:08 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
22/11/16 18:35:08 INFO DAGScheduler: ResultStage 3 (showString at NativeMethodAccessorImpl.java:0) finished in 5.441 s
22/11/16 18:35:08 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/16 18:35:08 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 18:35:08 INFO DAGScheduler: Job 1 finished: showString at NativeMethodAccessorImpl.java:0, took 5.473319 s
22/11/16 18:35:08 INFO CodeGenerator: Code generated in 53.6056 ms
22/11/16 18:35:08 INFO CodeGenerator: Code generated in 65.5442 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows

22/11/16 18:35:08 INFO SparkContext: Invoking stop() from shutdown hook
22/11/16 18:35:08 INFO SparkUI: Stopped Spark web UI at http://host.docker.internal:4042
22/11/16 18:35:08 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

As we can see the cpu time for the disk_only is less than the Memory_only. Memory_only uses high space compare to disk_only .



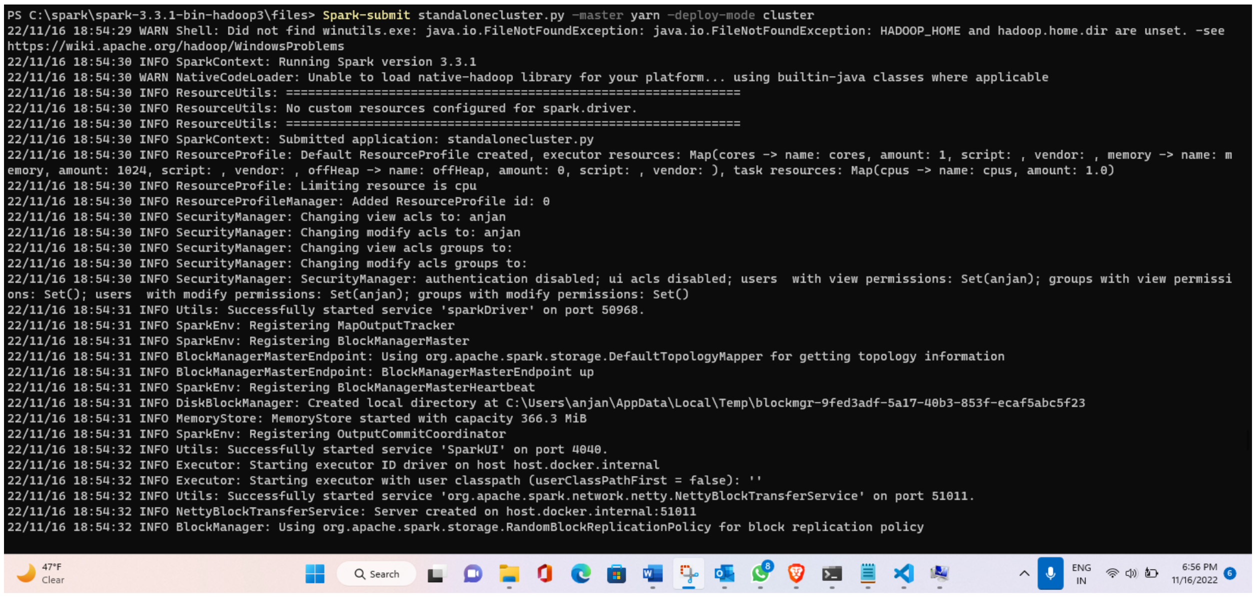
Active Jobs (1)

Page: 1   1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166 (kill)	2022/11/16 18:43:50	3 s	0/2	0/3 (1 running)

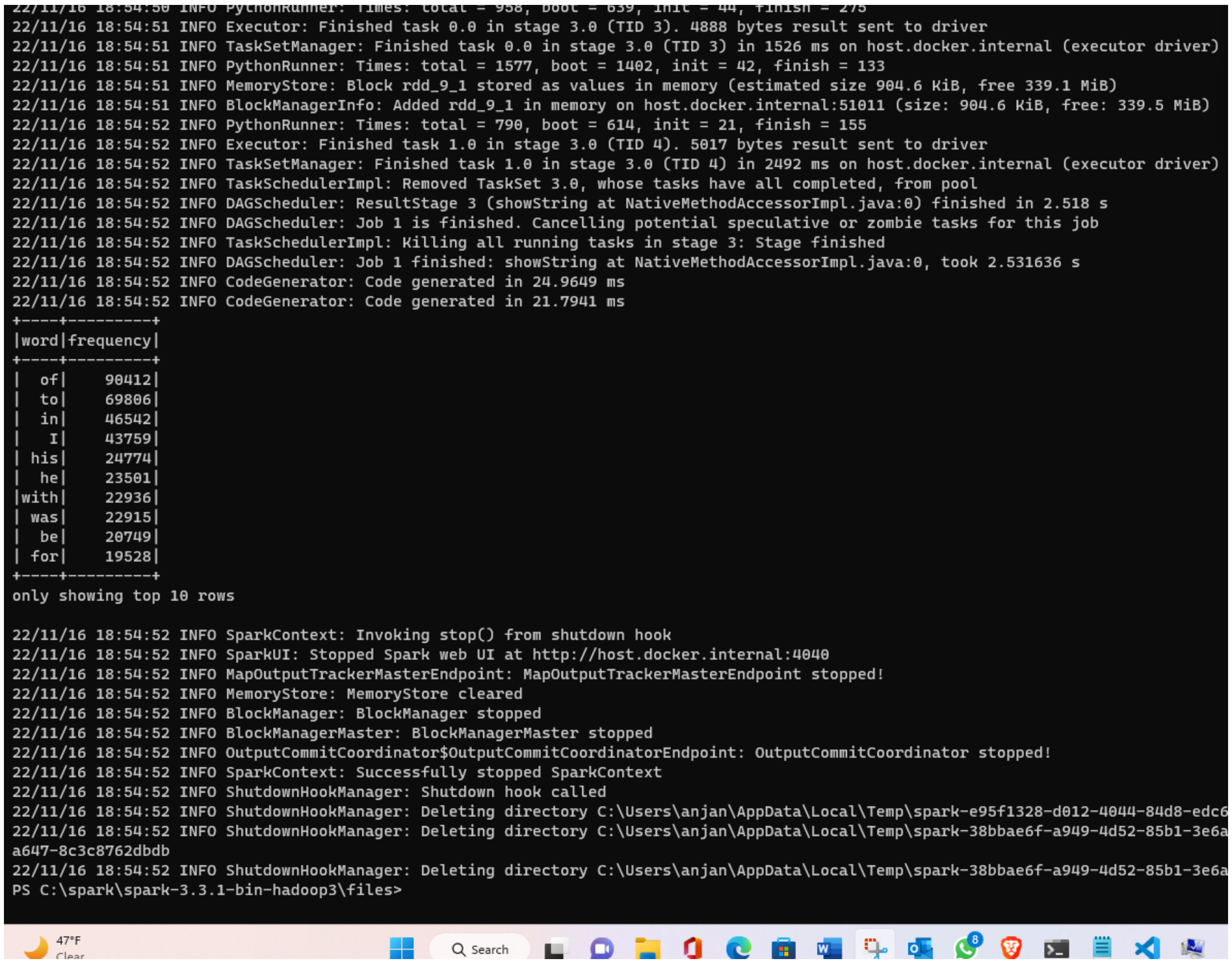
Page: 1   1 Pages. Jump to 1 . Show 100 items in a page. Go

3)Working with spark standalone cluster:



```
PS C:\spark\spark-3.3.1-bin-hadoop3\files> Spark-submit standalonecluster.py -master yarn -deploy-mode cluster
22/11/16 18:54:29 WARN Shell: Did not find winutils.exe: java.io.FileNotFoundException: java.io.FileNotFoundException: HADOOP_HOME and hadoop.home.dir are unset. -see
https://wiki.apache.org/hadoop/WindowsProblems
22/11/16 18:54:30 INFO SparkContext: Running Spark version 3.3.1
22/11/16 18:54:30 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
22/11/16 18:54:30 INFO ResourceUtils: ==============================================================
22/11/16 18:54:30 INFO ResourceUtils: No custom resources configured for spark.driver.
22/11/16 18:54:30 INFO ResourceUtils: ==============================================================
22/11/16 18:54:30 INFO SparkContext: Submitted application: standalonecluster.py
22/11/16 18:54:30 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(cores -> name: cores, amount: 1, script: , vendor: , memory -> name: m
emory, amount: 1024, script: , vendor: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpus, amount: 1.0)
22/11/16 18:54:30 INFO ResourceProfile: Limiting resource is cpu
22/11/16 18:54:30 INFO ResourceProfileManager: Added ResourceProfile id: 0
22/11/16 18:54:30 INFO SecurityManager: Changing view acls to: anjan
22/11/16 18:54:30 INFO SecurityManager: Changing modify acls to: anjan
22/11/16 18:54:30 INFO SecurityManager: Changing view acls groups to:
22/11/16 18:54:30 INFO SecurityManager: Changing modify acls groups to:
22/11/16 18:54:30 INFO SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users  with view permissions: Set(anjan); groups with view permissi
ons: Set(); users  with modify permissions: Set(anjan); groups with modify permissions: Set()
22/11/16 18:54:31 INFO Utils: Successfully started service 'sparkDriver' on port 50968.
22/11/16 18:54:31 INFO SparkEnv: Registering MapOutputTracker
22/11/16 18:54:31 INFO SparkEnv: Registering BlockManagerMaster
22/11/16 18:54:31 INFO BlockManagerMasterEndpoint: Using org.apache.spark.storage.DefaultTopologyMapper for getting topology information
22/11/16 18:54:31 INFO BlockManagerMasterEndpoint: BlockManagerMasterEndpoint up
22/11/16 18:54:31 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
22/11/16 18:54:31 INFO DiskBlockManager: Created local directory at C:\Users\anjan\AppData\Local\Temp\blockmgr-9fed3adf-5a17-40b3-853f-ecaf5abc5f23
22/11/16 18:54:31 INFO MemoryStore: MemoryStore started with capacity 366.3 MiB
22/11/16 18:54:31 INFO SparkEnv: Registering OutputCommitCoordinator
22/11/16 18:54:32 INFO Utils: Successfully started service 'SparkUI' on port 4040.
22/11/16 18:54:32 INFO Executor: Starting executor ID driver on host host.docker.internal
22/11/16 18:54:32 INFO Executor: Starting executor with user classpath (userClassPathFirst = false): ''
22/11/16 18:54:32 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBlockTransferService' on port 51011.
22/11/16 18:54:32 INFO NettyBlockTransferService: Server created on host.docker.internal:51011
22/11/16 18:54:32 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for block replication policy
```

By using the master yarn the execution time is reduced to 2.531636.



```
22/11/16 18:54:50 INFO PythonRunner: Times: total = 958, boot = 639, init = 44, finish = 275
22/11/16 18:54:51 INFO Executor: Finished task 0.0 in stage 3.0 (TID 3). 4888 bytes result sent to driver
22/11/16 18:54:51 INFO TaskSetManager: Finished task 0.0 in stage 3.0 (TID 3) in 1526 ms on host.docker.internal (executor driver)
22/11/16 18:54:51 INFO PythonRunner: Times: total = 1577, boot = 1402, init = 42, finish = 133
22/11/16 18:54:51 INFO MemoryStore: Block rdd_9_1 stored as values in memory (estimated size 904.6 KiB, free 339.1 MiB)
22/11/16 18:54:51 INFO BlockManagerInfo: Added rdd_9_1 in memory on host.docker.internal:51011 (size: 904.6 KiB, free: 339.5 MiB)
22/11/16 18:54:52 INFO PythonRunner: Times: total = 790, boot = 614, init = 21, finish = 155
22/11/16 18:54:52 INFO Executor: Finished task 1.0 in stage 3.0 (TID 4). 5017 bytes result sent to driver
22/11/16 18:54:52 INFO TaskSetManager: Finished task 1.0 in stage 3.0 (TID 4) in 2492 ms on host.docker.internal (executor driver)
22/11/16 18:54:52 INFO TaskSchedulerImpl: Removed TaskSet 3.0, whose tasks have all completed, from pool
22/11/16 18:54:52 INFO DAGScheduler: ResultStage 3 (showString at NativeMethodAccessorImpl.java:0) finished in 2.518 s
22/11/16 18:54:52 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie tasks for this job
22/11/16 18:54:52 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 18:54:52 INFO DAGScheduler: Job 1 finished: showString at NativeMethodAccessorImpl.java:0, took 2.531636 s
22/11/16 18:54:52 INFO CodeGenerator: Code generated in 24.9649 ms
22/11/16 18:54:52 INFO CodeGenerator: Code generated in 21.7941 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows

22/11/16 18:54:52 INFO SparkContext: Invoking stop() from shutdown hook
22/11/16 18:54:52 INFO SparkUI: Stopped Spark web UI at http://host.docker.internal:4040
22/11/16 18:54:52 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
22/11/16 18:54:52 INFO MemoryStore: MemoryStore cleared
22/11/16 18:54:52 INFO BlockManager: BlockManager stopped
22/11/16 18:54:52 INFO BlockManagerMaster: BlockManagerMaster stopped
22/11/16 18:54:52 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
22/11/16 18:54:52 INFO SparkContext: Successfully stopped SparkContext
22/11/16 18:54:52 INFO ShutdownHookManager: Shutdown hook called
22/11/16 18:54:52 INFO ShutdownHookManager: Deleting directory C:\Users\anjan\AppData\Local\Temp\spark-e95f1328-d012-4044-84d8-edc6
22/11/16 18:54:52 INFO ShutdownHookManager: Deleting directory C:\Users\anjan\AppData\Local\Temp\spark-38bbae6f-a949-4d52-85b1-3e6a
a647-8c3c8762dbdb
22/11/16 18:54:52 INFO ShutdownHookManager: Deleting directory C:\Users\anjan\AppData\Local\Temp\spark-38bbae6f-a949-4d52-85b1-3e6a
PS C:\spark\spark-3.3.1-bin-hadoop3\files>
```

4) In dynamic creation of pods in a application interacts with kubernates API and add them in the following way

1.first we create an instance of our custom resource here the application becomes active checks the number of declared replicas and how many of replicas are available

and creates if a greater number of replicas are required.

2.To add a new node to the cluster the application becomes active checks if there any pending replicas in the instance of our custom resource if so schedules one of them to the new node

3.While removing the node from the cluster the application becomes active. It will make sure the replicas on the removed node are not scheduled to another node.

For each newly created pod or other unscheduled pod, the kube-scheduler chooses the best node to run them. However, each container in a pod has different resource requirements, and each pod has different requirements. Therefore, existing nodes should be filtered according to specific scheduling requirements.Node affinity will allow to constrain which nodes contains Pod can be scheduled on based on node labels.