

Chapter 1

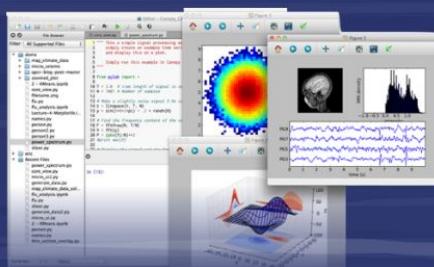


← → C https://www.enthought.com

ENTHOUGHT SCIENTIFIC COMPUTING SOLUTIONS

PRODUCTS TRAINING CONSULTING COMPANY CONTACT

DOWNLOADS: Canopy | PyXLL | View cart (\$0) | Create Account or Log In



ENTHOUGHT
CANOPY

- One-Click Python Deployment
- Analysis Environment
- Development Platform
- Integrated Training on Demand

[Get Canopy >](#)

Python Training on Demand Enthought Canopy Python for Excel Software Consulting

Enthought's mission is to significantly improve the way scientific computing is accomplished by providing powerful tools for quantitative data analysis and visualization.

Secure | https://store.enthought.com/downloads/

DOWNLOADS: [Canopy](#) | [PyXLL](#) | [View cart \(\\$0\)](#) | [Create Account or Log In](#)

ENTHOUGHT
SCIENTIFIC COMPUTING SOLUTIONS

PRODUCTS TRAINING CONSULTING COMPANY CONTACT

Download Canopy Express

By downloading Canopy you acknowledge your acceptance of all the terms and conditions of the applicable license.

v2.1.1	v1.7.4	Documentation	Platform	Python	Released	Size	MD5	
			Linux [64-bit]	2.7	download	2017-05-24	697.7 MB	624589a8c2f1647153c2c179000496e2
			Linux [64-bit]	3.5	download	2017-05-24	574.7 MB	770e89b488d001233687c515e28fb946
			macOS [64-bit]	2.7	download	2017-05-24	572.0 MB	07125467274a0ce7f2b9a984d0306694
			macOS [64-bit]	3.5	download	2017-05-24	464.0 MB	29f27e8de1d3f5fe5584821393f79642
			Windows [64-bit]	2.7	download	2017-05-24	513.7 MB	32de1a526d28da1399a2743050df6105
			Windows [32-bit]	2.7	download	2017-05-24	420.8 MB	c7d75419a3e3c05bbca68669af103802
			Windows [64-bit]	3.5	download	2017-05-24	431.2 MB	39c3f8a1882b149d2a39f0f0632e4e0e
			Windows [32-bit]	3.5	download	2017-05-24	350.1 MB	c2edaa90294adc724f3ea8d3cccf9c4

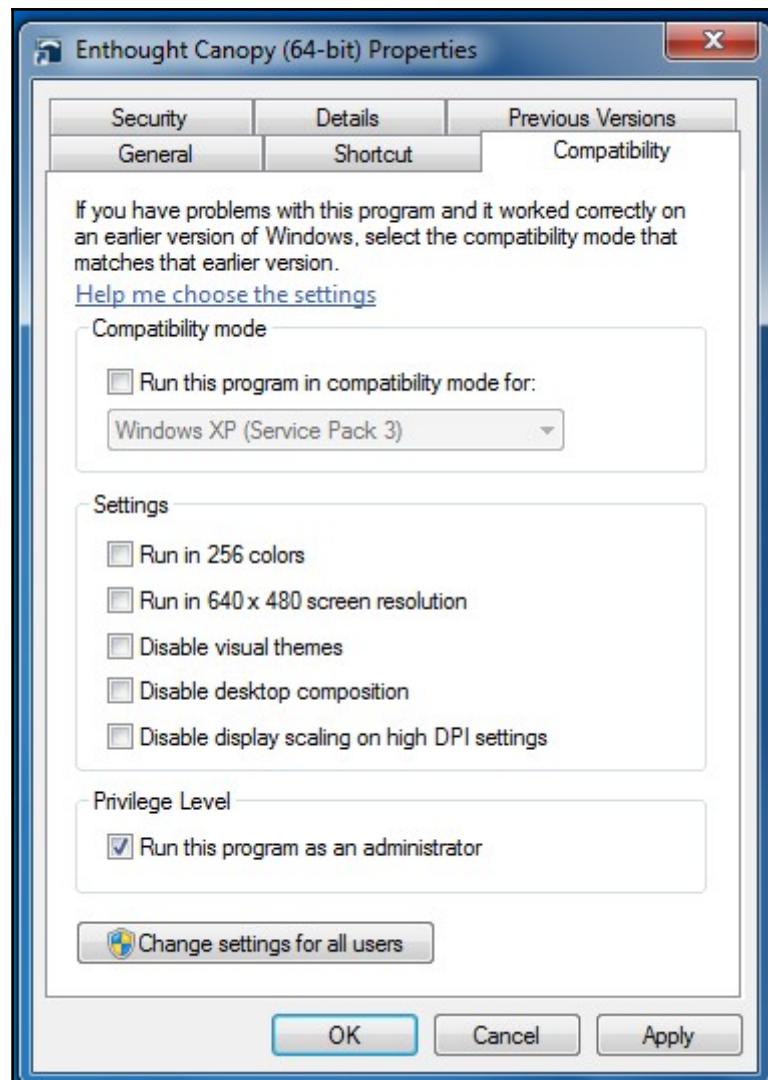
About Canopy Express

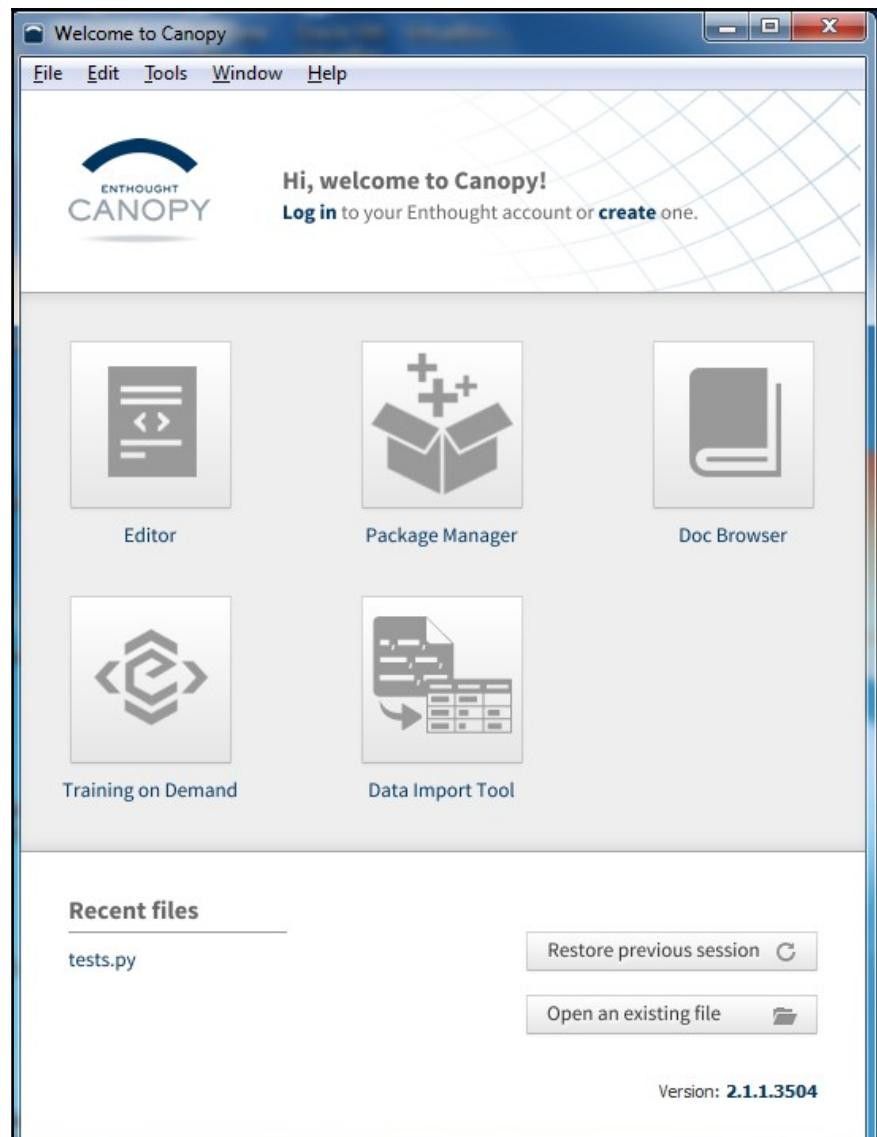
Canopy Express is free for all users and includes access to over 450 pre-built, tested, and dependency-aware Python packages, plus an integrated analysis environment.

Additional features, such as a Data Import Tool, graphical debugger and variable browser, online training courses, and technical support, are also available with a Canopy subscription plan.

Python Training from the Pros

With Canopy you'll have a robust environment and tools for working in Python. Now learn how to maximize your results with training from Enthought's experts.





www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html

ORACLE

Menu

Sign In | Country | Call

Oracle Technology Network > Java > Java SE > Downloads

Java SE Downloads

Java Platform (JDK) 8u131

NetBeans with JDK 8

Java Platform, Standard Edition

Important planned change for MD5-signed JARs

Starting with the April Critical Patch Update releases, planned for April 18 2017, all JRE versions will treat JARs signed with MD5 as unsigned. Learn more and view testing information.

For more information on cryptographic algorithm support, please check the JRE and JDK Crypto Roadmap.

Installation Instructions

Release Notes

Oracle License

Java SE Products

Third Party Licenses

Certified System Configurations

Readme Files

- JDK ReadMe
- JRE ReadMe

JDK DOWNLOAD

Server JRE DOWNLOAD

JRE DOWNLOAD

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

Which Java package do I need?

- Software Developers: JDK (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.

Java Platform, Standard Edition

Java SE 8u131

Java SE 8u131 includes important security fixes and bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.

[Learn more ➔](#)

Important planned change for MD5-signed JARs

Starting with the April Critical Patch Update releases, planned for April 18 2017, all JRE versions will treat JARs signed with MD5 as unsigned. Learn more and view testing instructions.

For more information on cryptographic algorithm support, please check the JRE and JDK Crypto Roadmap.

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK ReadMe
 - JRE ReadMe

JDK

[DOWNLOAD ↴](#)

Server JRE

[DOWNLOAD ↴](#)

JRE

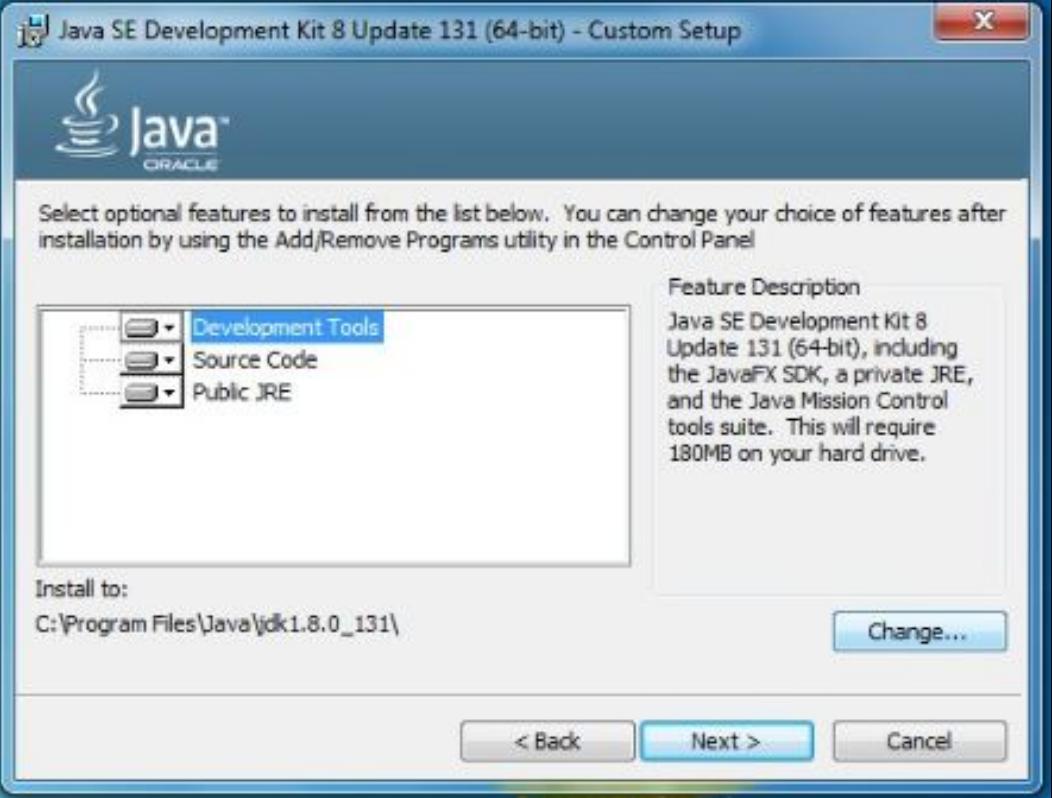
[DOWNLOAD ↴](#)

Java SE Development Kit 8u131

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.66 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.95 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u131-solaris-x64.tar.Z
Solaris x64	96.96 MB	jdk-8u131-solaris-x64.tar.gz
Windows x86	191.22 MB	jdk-8u131-windows-i586.exe
Windows x64	198.03 MB	jdk-8u131-windows-x64.exe



Java SE Development Kit 8 Update 131 (64-bit) - Change Folder



Browse to the new destination folder

Look in:

jdk1.8.0_131



Folder name:

c:\jdk

OK

Cancel

Java Setup - Destination Folder



Destination Folder

Click "Change" to install Java to a different folder.

Install to:

C:\Program Files\Java\jre1.8.0_131

Change...



spark.apache.org

 *Lightning-fast cluster computing*

Download Libraries Documentation Examples Community Developers Apache Software Foundation

Apache Spark™ is a fast and general engine for large-scale data processing.

Speed

Run programs up to 100x faster than Hadoop MapReduce in memory, or 10x faster on disk.

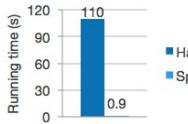
Apache Spark has an advanced DAG execution engine that supports acyclic data flow and in-memory computing.

Ease of Use

Write applications quickly in Java, Scala, Python, R.

Spark offers over 80 high-level operators that make it easy to build parallel apps. And you can use it *interactively* from the Scala, Python and R shells.

Running time (s)



System	Running time (s)
Hadoop	110
Spark	0.9

Logistic regression in Hadoop and Spark

Latest News

- Spark 2.1.1 released (May 02, 2017)
- Spark Summit (June 5-7th, 2017, San Francisco) agenda posted (Mar 31, 2017)
- Spark Summit East (Feb 7-9th, 2017, Boston) agenda posted (Jan 04, 2017)
- Spark 2.1.0 released (Dec 28, 2016)

Archive

[Download Spark](#)

Built-in Libraries:

- SQL and DataFrames
- Spark Streaming
- MLlib (machine learning)
- GraphX (graph)

Third-Party Projects

```
text_file = spark.textFile("hdfs://...")  
text_file.flatMap(lambda line: line.split())  
    .map(lambda word: (word, 1))  
    .reduceByKey(lambda a, b: a+b)  
  
Word count in Spark's Python API
```

Download Apache Spark™

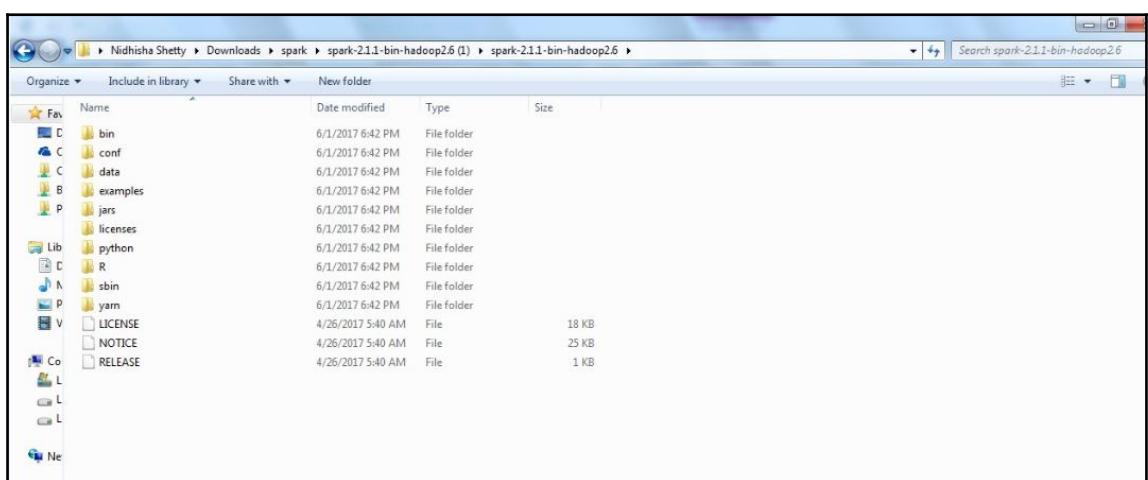
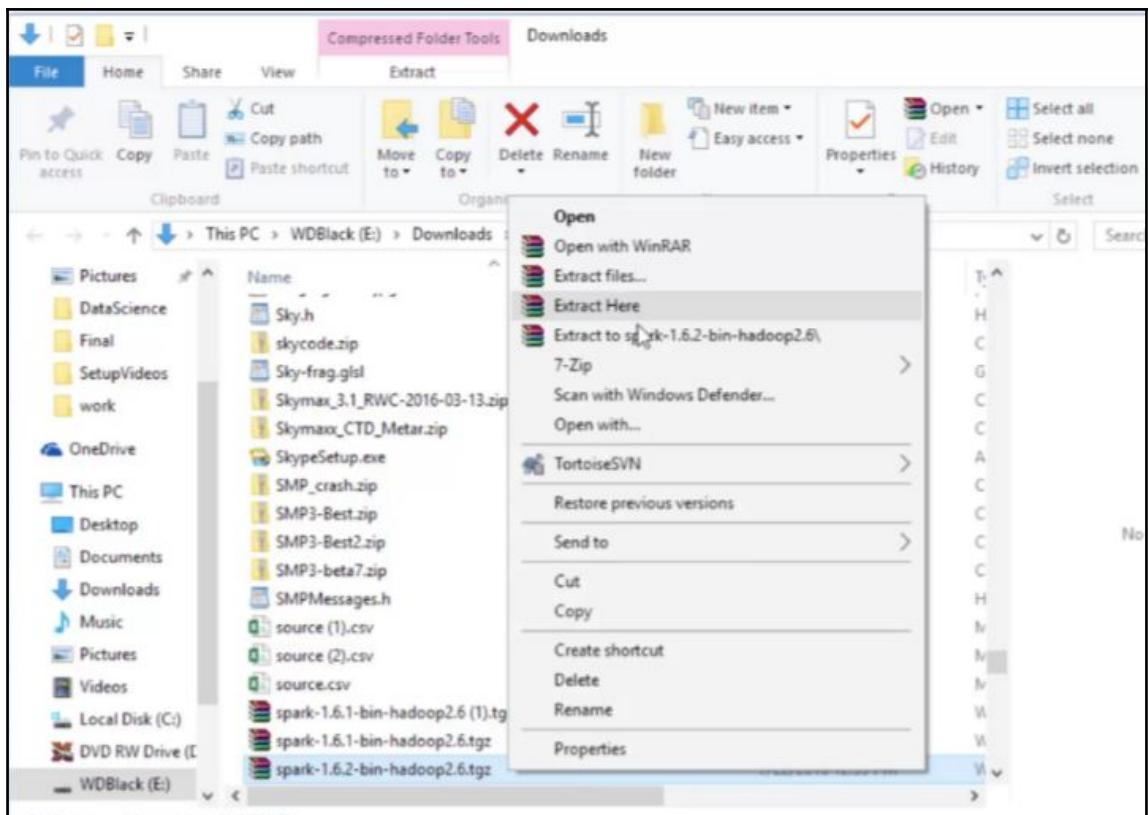
- Choose a Spark release: [2.1.1 \(May 02 2017\)](#)
- Choose a package type: [Pre-built for Apache Hadoop 2.7 and later](#)
- Choose a download type: [Direct Download](#)
- Download Spark: [spark-2.1.1-bin-hadoop2.7.tgz](#)
- Verify this release using the [2.1.1 signatures and checksums](#) and [project release KEYS](#).

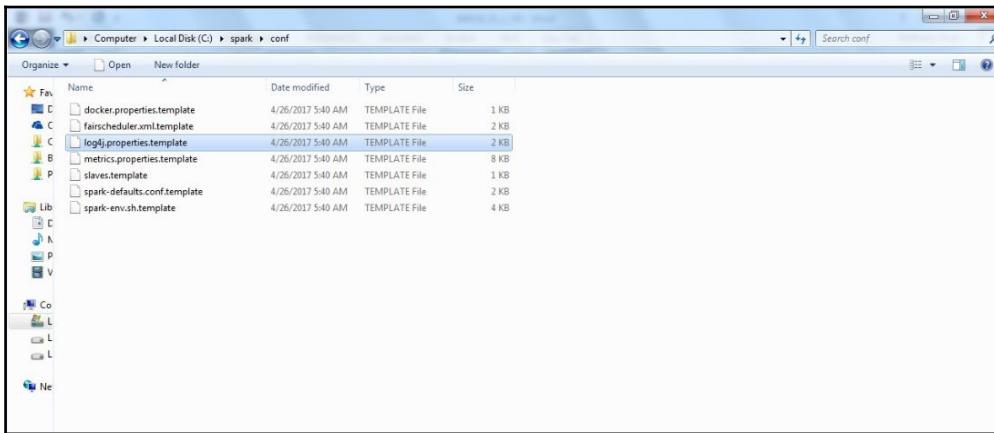
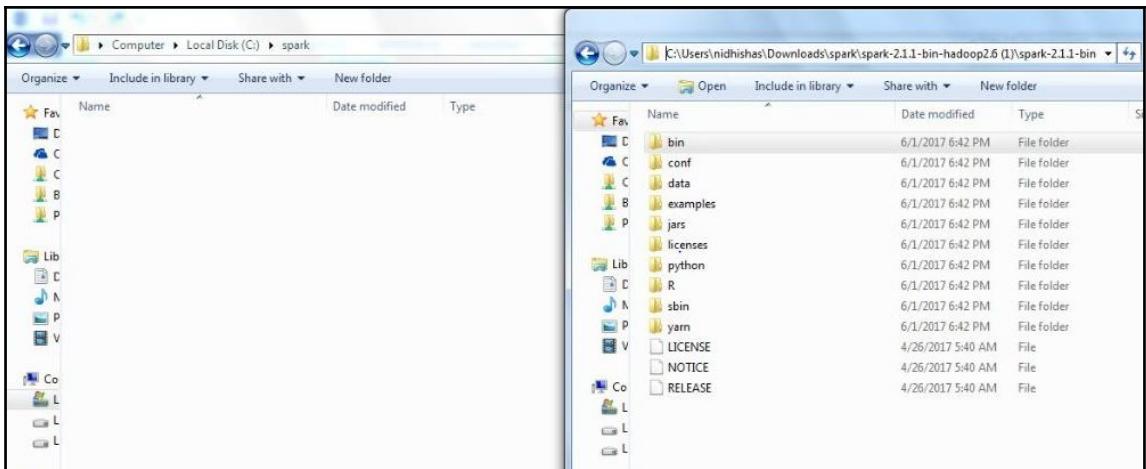
Note: Starting version 2.0, Spark is built with Scala 2.11 by default. Scala 2.10 users should download the Spark source package and build with Scala 2.10 support.

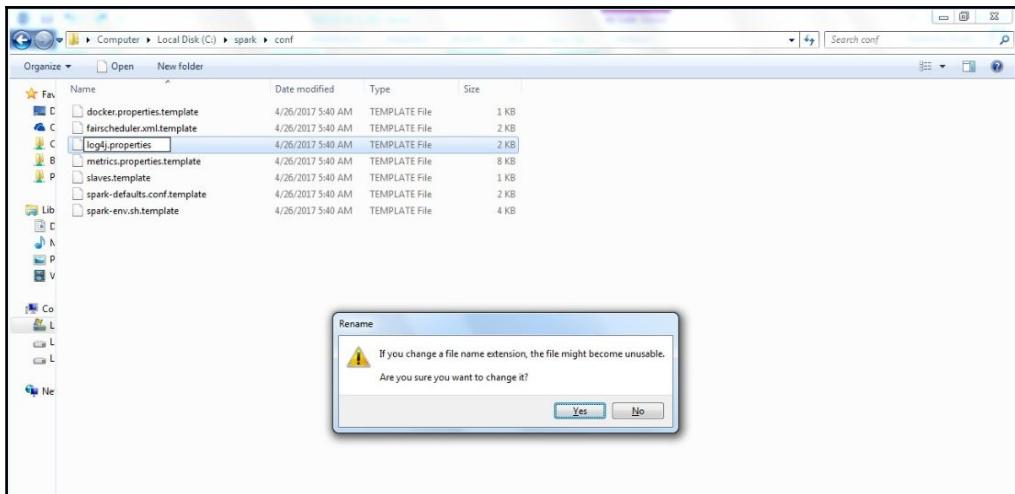
www.rarlab.com/download.htm

RARLAB WinRAR and RAR archiver downloads

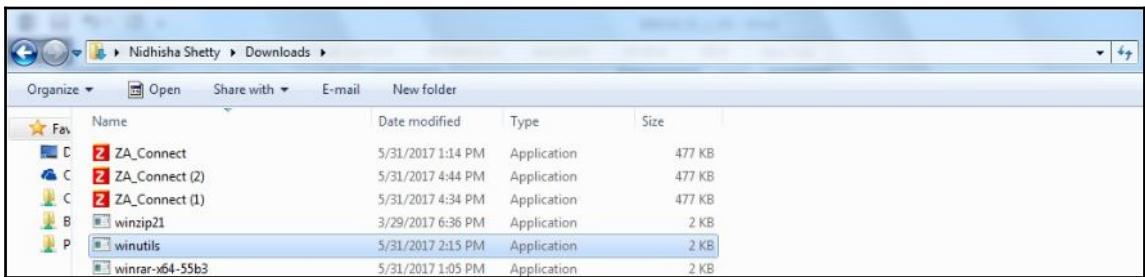
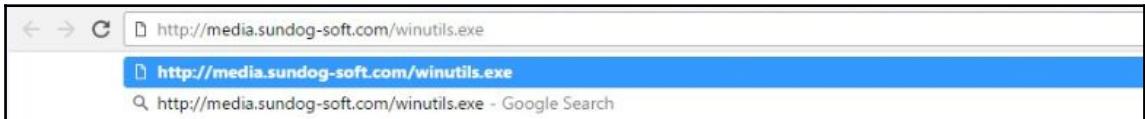
Home	Latest English WinRAR and RAR beta versions			
RAR	Software name	User interface	License	Size
	WinRAR x86 (32 bit) 5.50 beta 3	Graphical and command line	Trial	1947 KB
	WinRAR x64 (64 bit) 5.50 beta 3	Graphical and command line	Trial	2162 KB
	RAR 5.50 beta 3 for Linux x64	Command line only	Trial	531 KB
	RAR 5.50 beta 3 for Linux x86	Command line only	Trial	521 KB
	RAR 5.50 beta 3 for FreeBSD	Command line only	Trial	920 KB
	RAR 5.50 beta 3 for Mac OS X	Command line only	Trial	499 KB
Downloads	Latest localized WinRAR beta versions			
Dealers	Language	Version	Size	
	Arabic (32 bit)	5.50 beta 3	1993 KB	
	Arabic (64 bit)	5.50 beta 3	2209 KB	
	Armenian (32 bit)	5.50 beta 3	1989 KB	
	Armenian (64 bit)	5.50 beta 3	2204 KB	
	Chinese Traditional (32 bit)	5.50 beta 3	2192 KB	
	Chinese Traditional (64 bit)	5.50 beta 3	2413 KB	
	English (32 bit)	5.50 beta 3	1947 KB	
	English (64 bit)	5.50 beta 3	2162 KB	
	Finnish (32 bit)	5.50 beta 3	1989 KB	
	Finnish (64 bit)	5.50 beta 3	2206 KB	
	French (32 bit)	5.50 beta 3	2044 KB	
	French (64 bit)	5.50 beta 3	2261 KB	
	German (32 bit)	5.50 beta 3	2067 KB	
	German (64 bit)	5.50 beta 3	2293 KB	
	Hungarian (32 bit)	5.50 beta 3	1987 KB	
	Hungarian (64 bit)	5.50 beta 3	2205 KB	
	Lithuanian (32 bit)	5.50 beta 3	2014 KB	
	Lithuanian (64 bit)	5.50 beta 3	2232 KB	
	Mongolian (32 bit)	5.50 beta 2	1995 KB	
	Mongolian (64 bit)	5.50 beta 2	2213 KB	
	Portuguese (32 bit)	5.50 beta 3	1988 KB	
	Portuguese (64 bit)	5.50 beta 3	2206 KB	
	Portuguese Brazilian (32 bit)	5.50 beta 3	3444 KB	
	Portuguese Brazilian (64 bit)	5.50 beta 3	3659 KB	
	Romanian (32 bit)	5.50 beta 2	2022 KB	
	Romanian (64 bit)	5.50 beta 2	2240 KB	
	Russian (32 bit)	5.50 beta 3	2064 KB	
	Russian (64 bit)	5.50 beta 3	2329 KB	
	Serbian Cyrillic (32 bit)	5.50 beta 3	2027 KB	
	Serbian Cyrillic (64 bit)	5.50 beta 3	2243 KB	
	Swedish (32 bit)	5.50 beta 3	1988 KB	
	Swedish (64 bit)	5.50 beta 3	2204 KB	
	Ukrainian (32 bit)	5.50 beta 3	1990 KB	
	Ukrainian (64 bit)	5.50 beta 3	2209 KB	

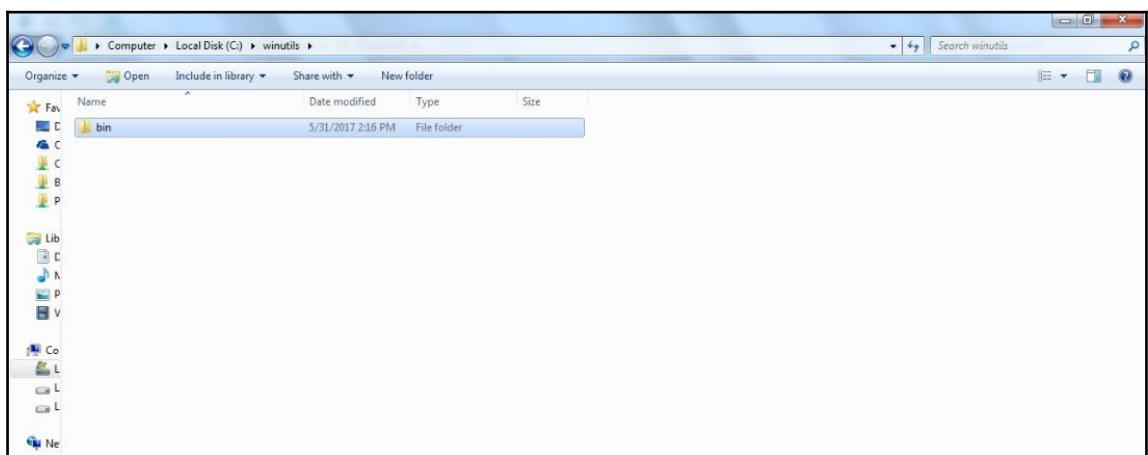
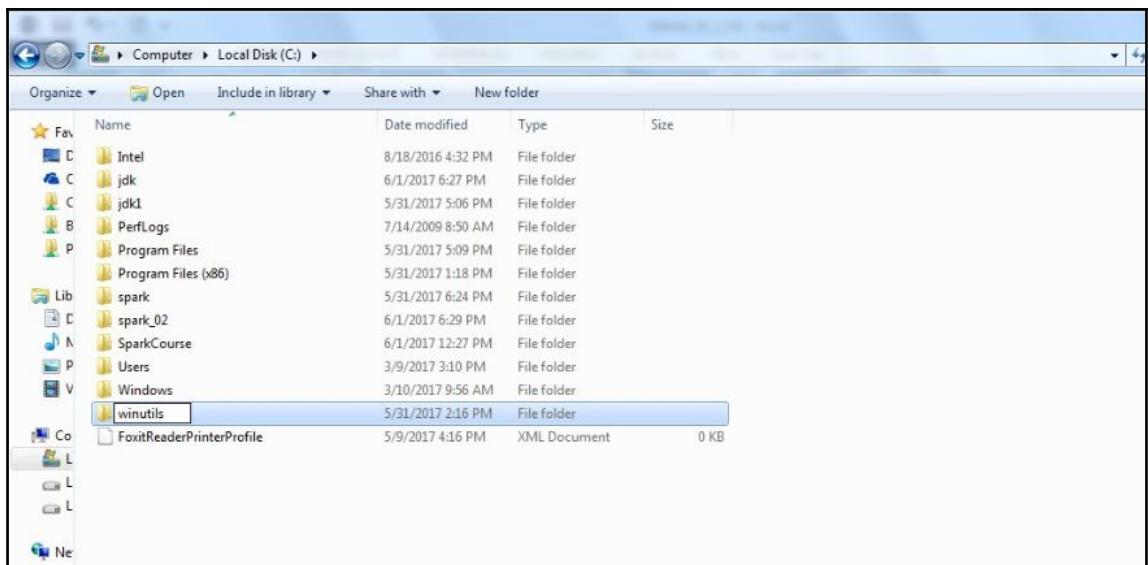


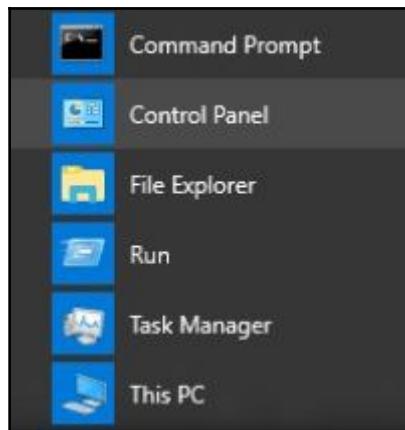
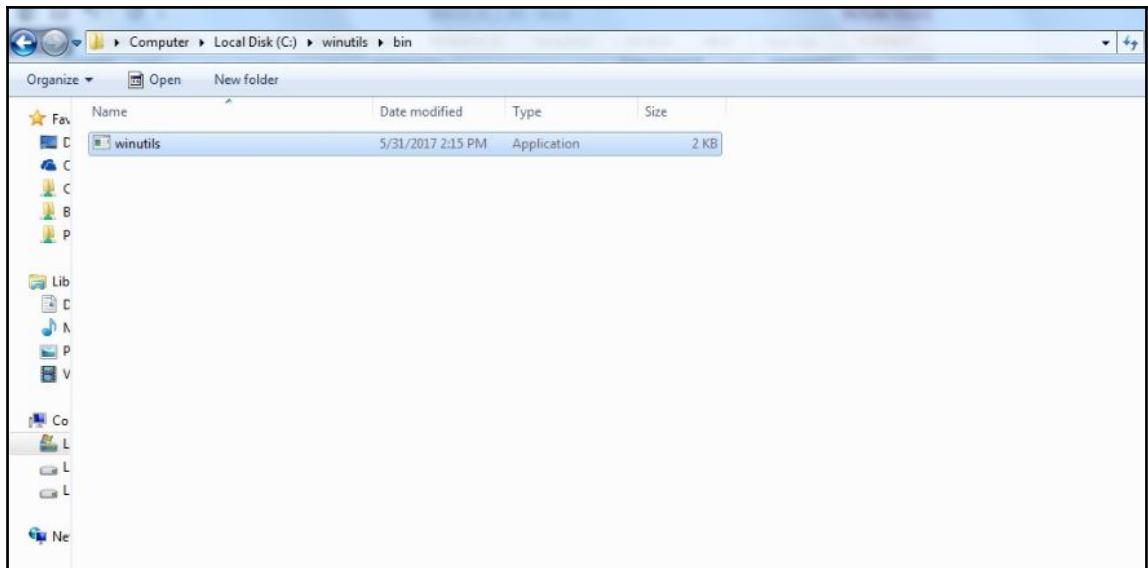


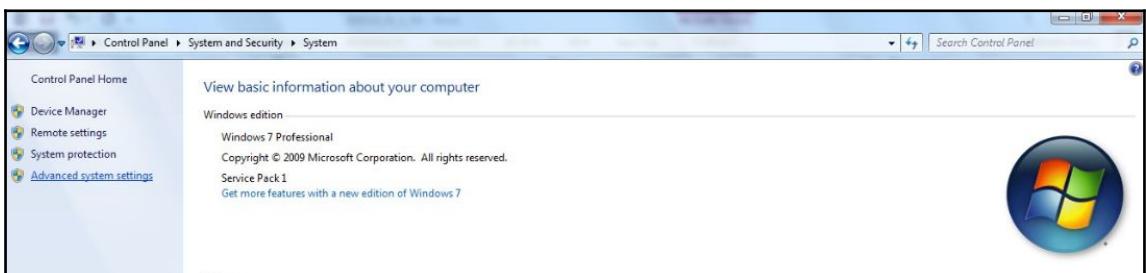
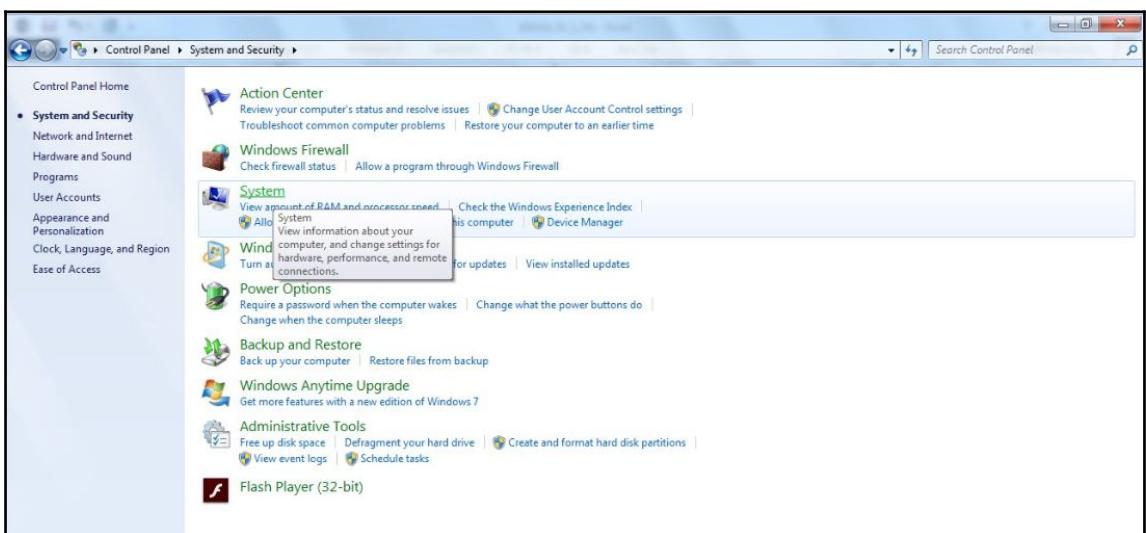


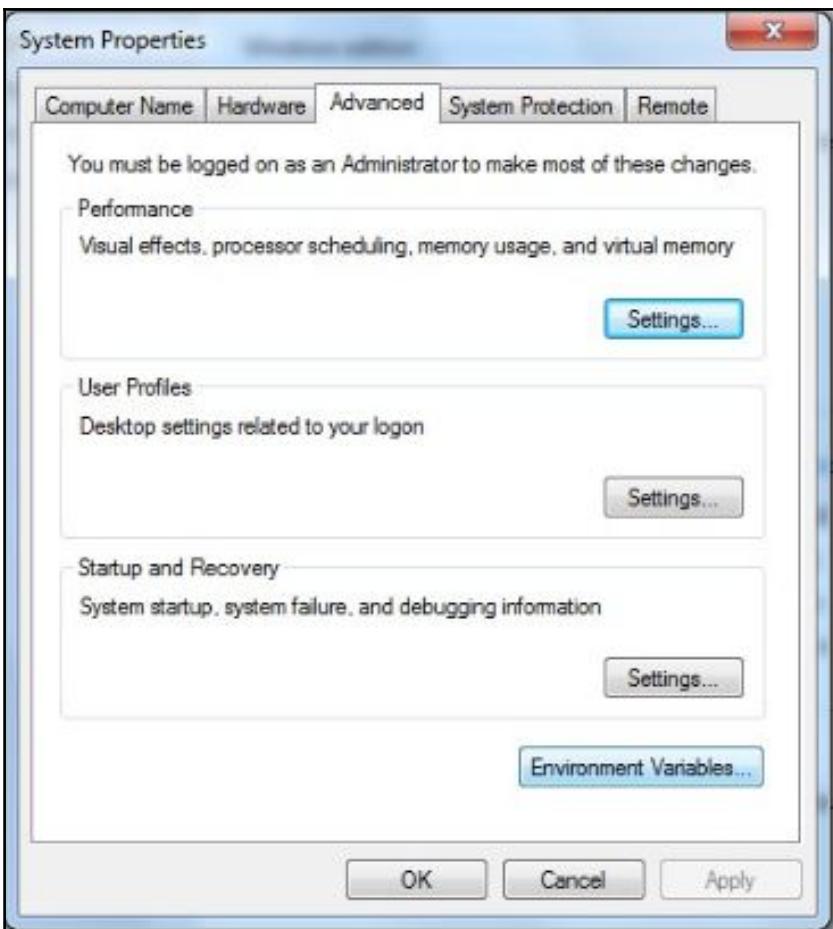
```
18 # Set everything to be logged to the console
19 log4j.rootCategory=INFO, console
20 log4j.appender.console=org.apache.log4j.ConsoleAppender
21 log4j.appender.console.target=System.err
22 log4j.appender.console.layout=org.apache.log4j.PatternLayout
23 log4j.appender.console.layout.ConversionPattern=%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n
24
```

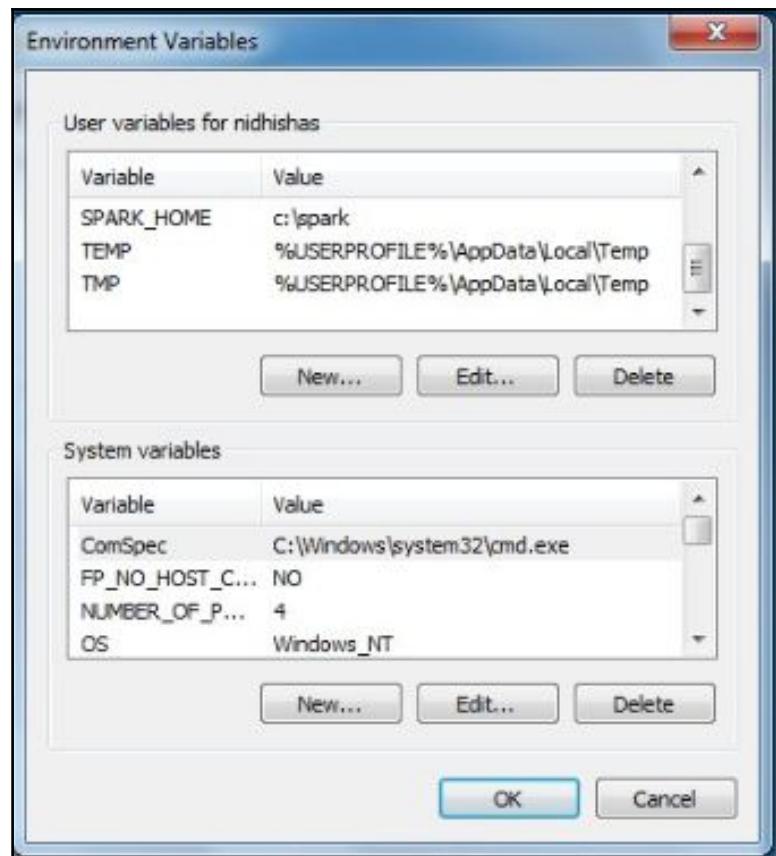


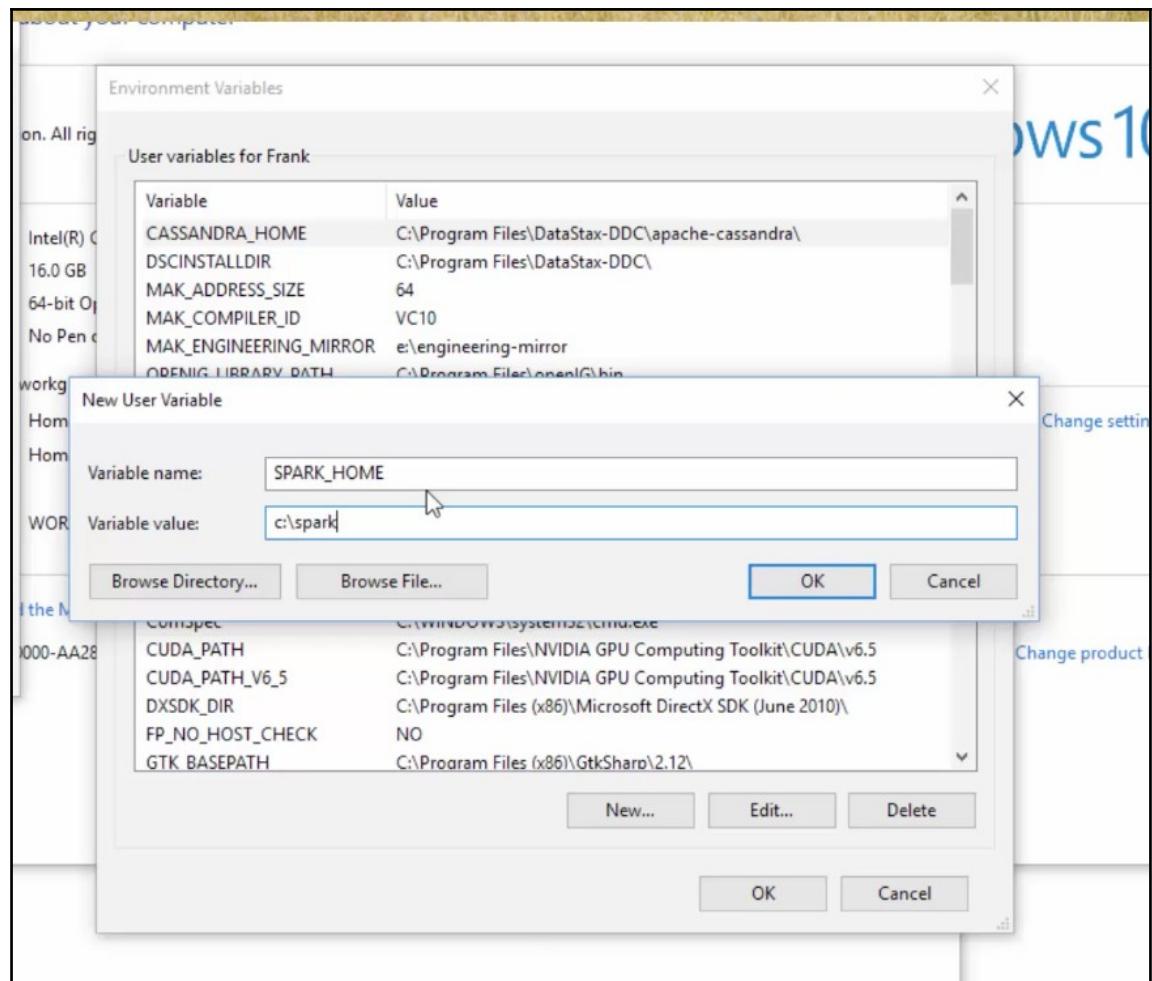


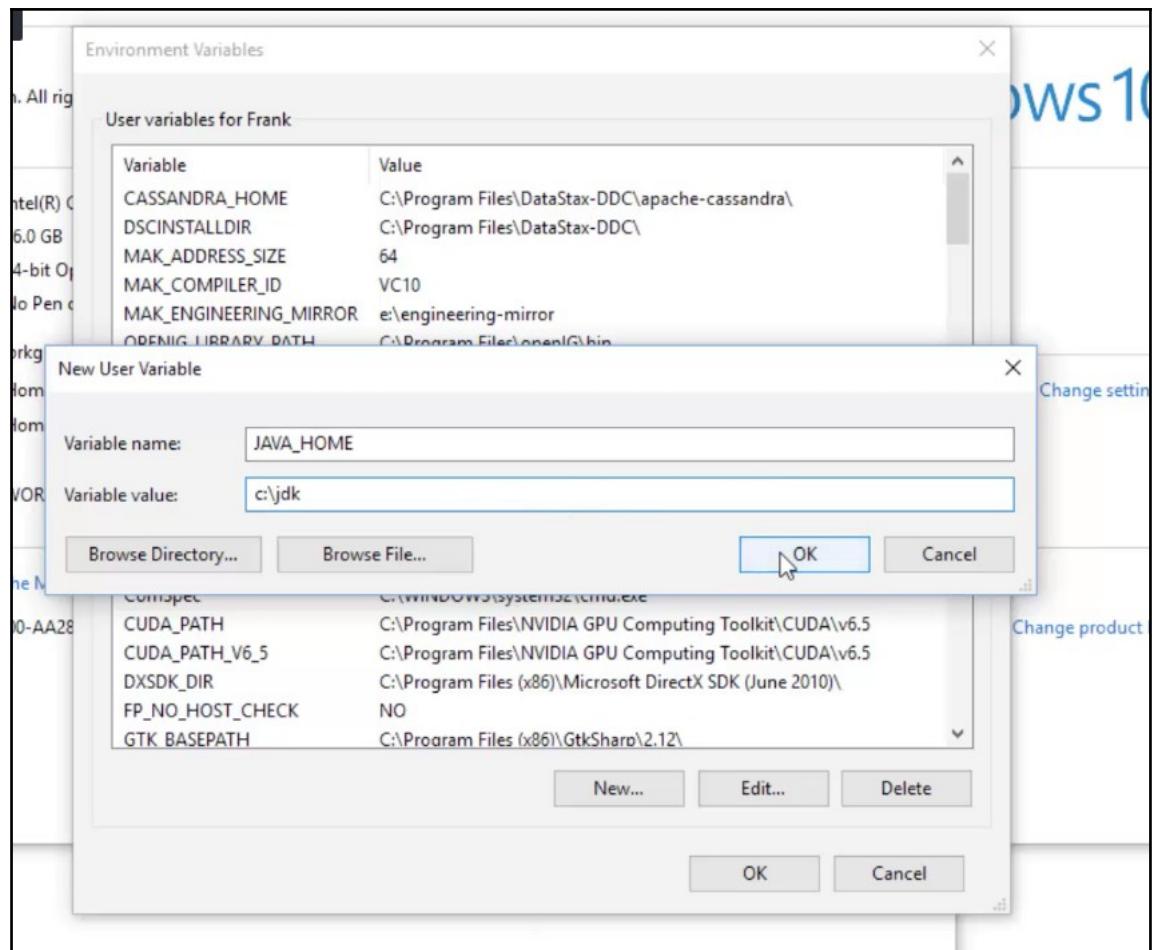












User variables for Frank	
Variable	Value
CASSANDRA_HOME	C:\Program Files\DataStax-DDC\apache-cassandra\
DSCINSTALLDIR	C:\Program Files\DataStax-DDC\
JAVA_HOME	c:\jdk
MAK_ADDRESS_SIZE	64
MAK_COMPILER_ID	VC10
MAK_ENGINEERING_MIRROR	eengineering-mirror

New User Variable

Variable name:

Variable value:

Comspec	C:\Windows\system\cmd.exe
CUDA_PATH	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v6.5
CUDA_PATH_V6_5	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v6.5
DXSDK_DIR	C:\Program Files (x86)\Microsoft DirectX SDK (June 2010)\
FP_NO_HOST_CHECK	NO
GTK BASEPATH	C:\Program Files (x86)\GtkSharp\2.12\

Environment Variables

X

User variables for Frank

Variable	Value
MAK_ENGINEERING_MIRROR	e:\engineering-mirror
OPENIG_LIBRARY_PATH	C:\Program Files\openIG\bin
OSG_BIN	C:\Program Files (x86)\OpenSceneGraph\bin
OSGDIR	C:\Program Files (x86)\OpenSceneGraph
OSGEARTH_BIN	C:\Program Files\OSGEARTH\bin
PATH	c:\users\frank\appdata\local\enthought\canopy\user\scripts;C:\Pr...
SILVERLINING_ENABLE_DEB...	1

New...

Edit...

Delete

Edit environment variable

X

```
c:\users\frank\appdata\local\enthought\canopy\user\scripts  
C:\Program Files (x86)\OpenSceneGraph\bin  
C:\Program Files (x86)\OSGEARTH\bin  
C:\cygwin64\bin  
C:\Program Files (x86)\CMake 2.8\bin  
C:\Program Files (x86)\Graphviz2.38\bin  
C:\Users\Frank\AppData\Local\Enthought\Canopy\User  
C:\Users\Frank\AppData\Local\Enthought\Canopy\User\Scripts  
C:\Users\Frank\AppData\Roaming\npm  
E:\boost\boost_1_59_0\lib64-msvc-12.0  
E:\OSG\OpenSceneGraph-3.4.0\3rdParty\x64\bin  
C:\Program Files\Docker Toolbox  
%ZOOKEEPER_HOME%\bin  
%JAVA_HOME%\bin  
%SPARK_HOME%\bin
```

New

Edit

Browse...

Delete

Move Up

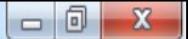
Move Down

Edit text...

OK

Cancel

Canopy Command Prompt



```
(User) C:\Users\nidhishas>cd c:\spark
```

```
c:\ Canopy Command Prompt
<User> C:\Users\nidhishas>cd c:\spark
<User> c:\spark>dir
 Volume in drive C has no label.
 Volume Serial Number is B4??-8D??
Directory of c:\spark

05/31/2017  06:24 PM    <DIR>          .
05/31/2017  06:24 PM    <DIR>          ..
05/31/2017  01:25 PM    <DIR>          bin
05/31/2017  06:26 PM            23 CHANGES.txt
06/01/2017  06:59 PM    <DIR>          conf
05/31/2017  01:25 PM    <DIR>          data
05/31/2017  01:25 PM    <DIR>          examples
05/31/2017  01:25 PM    <DIR>          jars
04/26/2017  05:40 AM        17,811 LICENSE
05/31/2017  01:25 PM    <DIR>          licenses
04/26/2017  05:40 AM        24,645 NOTICE
05/31/2017  01:25 PM    <DIR>          python
05/31/2017  01:25 PM    <DIR>          R
04/26/2017  05:40 AM        3,817 README.md
04/26/2017  05:40 AM         128 RELEASE
05/31/2017  01:25 PM    <DIR>          sbin
05/31/2017  01:25 PM    <DIR>          yarn
05/31/2017  01:25 PM    <DIR>          5 File(s)      46,424 bytes
12 Dir(s)   181,354,946,560 bytes free

<User> c:\spark>
```

(User) c:\spark> pyspark

```
<User> c:\spark>pyspark
Enthought Deployment Manager -- https://www.enthought.com
Python 2.7.13 |Enthought, Inc. <x86_64>| <default, Mar 2 2017, 16:05:12> [MSC v.
1.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(
newLevel).
17/06/07 15:01:46 WARN NativeCodeLoader: Unable to load native-hadoop library fo
r your platform... using builtin-java classes where applicable
17/06/07 15:01:50 WARN ObjectStore: Failed to get database global_temp, returnin
g NoSuchObjectException
Welcome to

    _____
   /     \
  /       \
 /  _   _ \
 \  \ / / /
  \  \ / / /
   \  \ / / /
    \  \ / / /
     \  \ / / /
      \  \ / / /
       \  \ / / /
        \  \ / / /
         \  \ / / /
          \  \ / / /
           \  \ / / /
            \  \ / / /
             \  \ / / /
              \  \ / / /
               \  \ / / /
                \  \ / / /
                 \  \ / / /
                  \  \ / / /
                   \  \ / / /
                     \  \ / / /
                      \  \ / / /
                        \  \ / / /
                         \  \ / / /
                          \  \ / / /
                           \  \ / / /
                            \  \ / / /
                             \  \ / / /
                              \  \ / / /
                               \  \ / / /
                                \  \ / / /
                                 \  \ / / /
                                   \  \ / / /
                                     \  \ / / /
                                       \  \ / / /
                                         \  \ / / /
                                           \  \ / / /
                                             \  \ / / /
                                               \  \ / / /
                                                 \  \ / / /
                                                   \  \ / / /
                                                     \  \ / / /
                                                       \  \ / / /
                                                         \  \ / / /
                                                           \  \ / / /
                                                             \  \ / / /
                                                               \  \ / / /
                                                                 \  \ / / /
                                                                   \  \ / / /
                                                                     \  \ / / /
                                                                       \  \ / / /
                                                                         \  \ / / /
                                                                           \  \ / / /
                                                                             \  \ / / /
                                                                               \  \ / / /
                                                                                 \  \ / / /
                                                                                   \  \ / / /
                                                                                     \  \ / / /
                       version 2.1.1

Using Python version 2.7.13 (default, Mar 2 2017 16:05:12)
SparkSession available as 'spark'.
```

```
>>> rdd = sc.textFile("README.md")
```

```
>>> rdd = sc.textFile("README.md")
>>> rdd.count()
[Stage 0:>                                         (0 + 2) / 2]
104
\\
```

The screenshot shows the GroupLens website at https://grouplens.org. The header features the "grouplens" logo and navigation links for "about", "datasets", "publications", and "blog". The main section has a teal background with the title "Social Computing Research at the University of Minnesota". Below the title is a subtitle: "GroupLens advances the theory and practice of social computing by building and understanding systems used by real people".

The screenshot shows the MovieLens "top picks" page at https://movielens.org/explore/top-picks. The page displays a grid of movie recommendations. At the top, it says "MovieLens recommends these movies" and "top picks". It shows 30,000 movies. A "sort by:" dropdown is set to "recommended". The movies listed include "The Godfather", "The Usual Suspect", "Spirited Away", "Casablanca", "Rear Window", "Blade Runner", "The Lives of Others", "Dr. Strangelove", "Black Mirror (2011)", "My Neighbor Totoro", "North by Northwest", "Pulp Fiction", "Memento", "Amélie", "City of God", "Schindler's List", "Vertigo", "Cosmos", "Inception", "Ewok", "Battlestar Galactica", "WALL-E", "Band of Brothers", and "Casablanca". Each movie entry includes its title, year, rating, and a small thumbnail image.

[Secure | https://grouplens.org/datasets/movielens/](https://grouplens.org/datasets/movielens/)

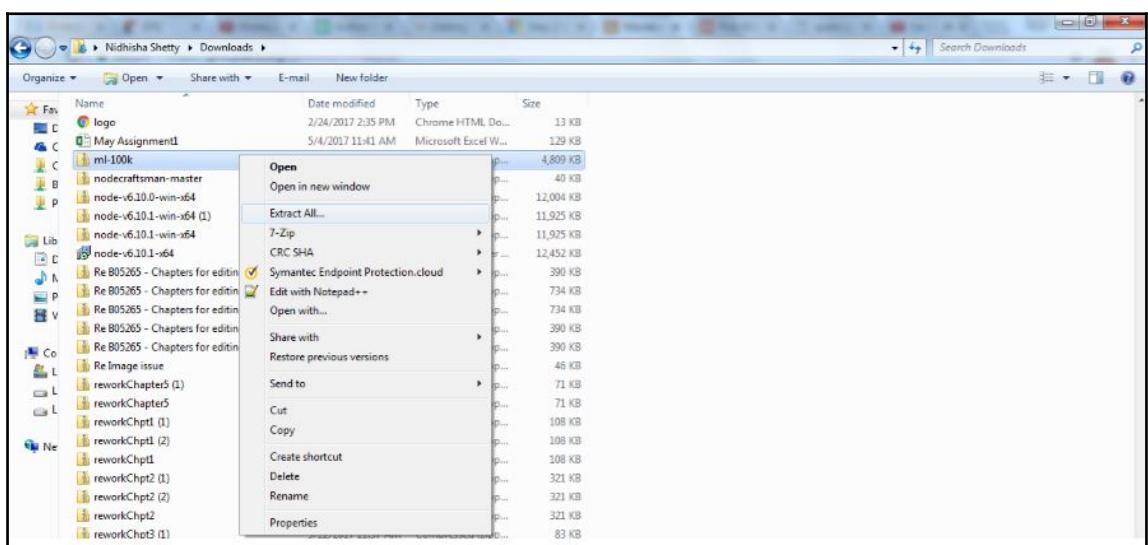
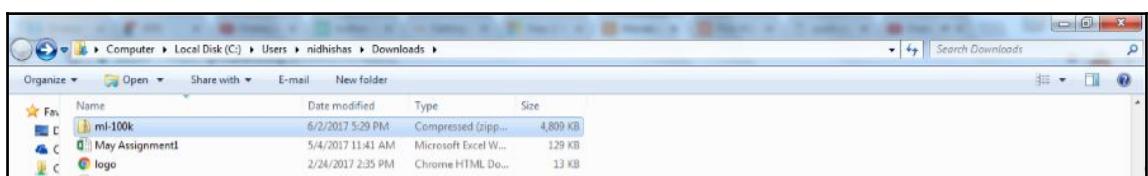
older datasets

MovieLens 100K Dataset

Stable benchmark dataset: 100,000 ratings from 1000 users on 1700 movies. Released 4/1998.

- [README.txt](#)
- [ml-100k.zip](#) (size: 5 MB, [checksum](#))
- [Index of unzipped files](#)

Permalink: <http://grouplens.org/datasets/movielens/100k/>



File Explorer window showing the contents of the 'ml-100k' folder:

Organize	Name	Date modified	Type	Size
Fav	alibut.pl	6/2/2017 5:38 PM	PL File	1 KB
C	mkut.sh	6/2/2017 5:38 PM	SH File	1 KB
C	README	6/2/2017 5:38 PM	File	7 KB
B	u.data	6/2/2017 5:38 PM	DATA File	1,933 KB
P	u.genre	6/2/2017 5:38 PM	GENRE File	1 KB
P	u.info	6/2/2017 5:38 PM	INFO File	1 KB
Lib	u.item	6/2/2017 5:38 PM	ITEM File	231 KB
C	u.occupation	6/2/2017 5:38 PM	OCCUPATION File	1 KB
N	u.user	6/2/2017 5:38 PM	USER File	23 KB
P	u1.base	6/2/2017 5:38 PM	BASE File	1,550 KB
V	u1.test	6/2/2017 5:38 PM	TEST File	384 KB
	u2.base	6/2/2017 5:38 PM	BASE File	1,547 KB
Co	u2.test	6/2/2017 5:38 PM	TEST File	386 KB
L	u3.base	6/2/2017 5:38 PM	BASE File	1,546 KB
L	u3.test	6/2/2017 5:38 PM	TEST File	388 KB
L	u4.base	6/2/2017 5:38 PM	BASE File	1,545 KB
	u4.test	6/2/2017 5:38 PM	TEST File	388 KB
N	u5.base	6/2/2017 5:38 PM	BASE File	1,545 KB
	u5.test	6/2/2017 5:38 PM	TEST File	389 KB
	ua.base	6/2/2017 5:38 PM	BASE File	1,751 KB
	ua.test	6/2/2017 5:38 PM	TEST File	183 KB
	ub.base	6/2/2017 5:38 PM	BASE File	1,751 KB
	ub.test	6/2/2017 5:38 PM	TEST File	183 KB

File Explorer window showing the contents of the 'SparkCourse' folder:

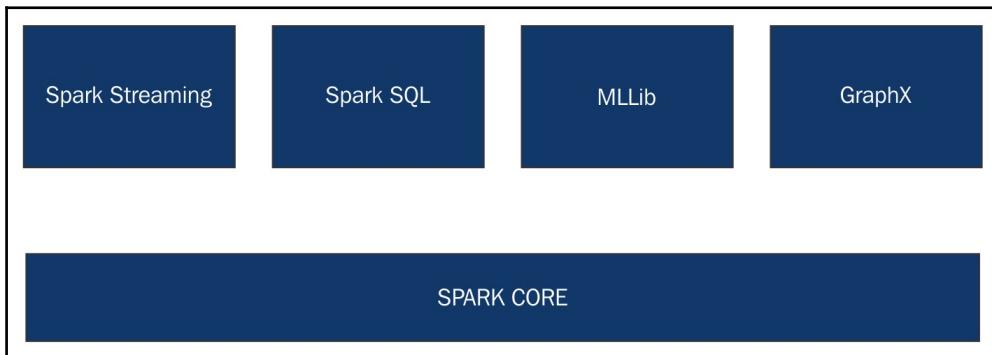
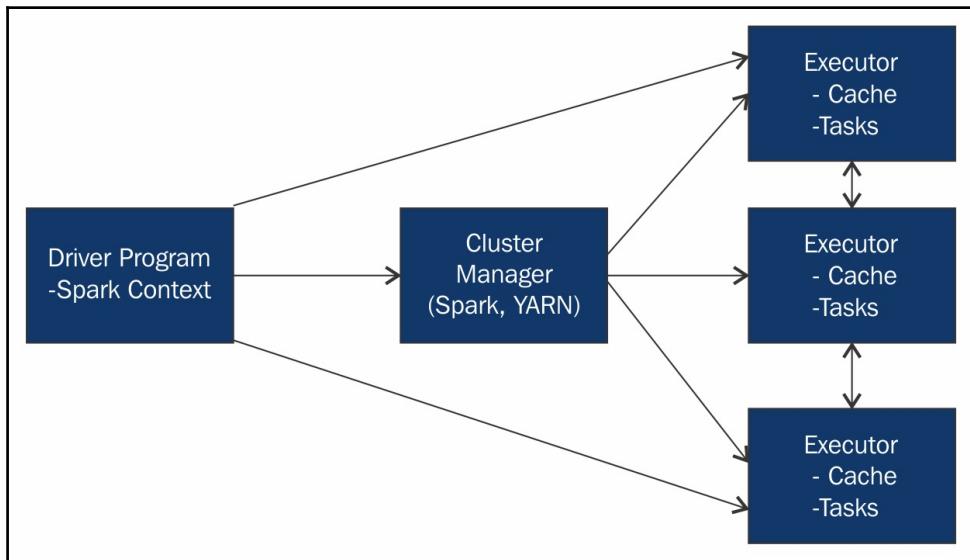
Organize	Name	Date modified	Type	Size
Fav	ml-100k	6/2/2017 5:40 PM	File folder	
C	ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB
C				
P				
C				
Lib				

```
Administrator: Canopy Command Prompt
<User> C:\Users\nidhishas>cd c:\SparkCourse
<User> c:\SparkCourse>dir
Volume in drive C has no label.
Volume Serial Number is B4??-8D??
Directory of c:\SparkCourse
06/07/2017  03:04 PM    <DIR> .
06/07/2017  03:04 PM    <DIR> ..
06/02/2017  05:40 PM    <DIR> ml-100k
06/07/2017  03:04 PM           452 ratings-counter.py
               1 File(s)      452 bytes
               3 Dir(s)  171,987,267,584 bytes free
<User> c:\SparkCourse>
```

```
<User> c:\SparkCourse>spark-submit ratings-counter.py
```

```
<User> c:\SparkCourse>spark-submit ratings-counter.py
1 6110
2 11370
3 27145
4 34174
5 21201
```

Chapter 2



Python code to square number in a data set:

```
nums=sc.parallelize ([1, 2, 3, 4])
squared=nums.map (lambda x: x * x). collect()
```

Scala code to square numbers in a data set:

```
val nums=sc.parallelize(List(1, 2, 3, 4))
val squared=input.map(x=>x * x). collect()
```

196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

196	242	3	881250949
-----	-----	---	-----------

186	302	3	891717742
-----	-----	---	-----------

22	377	1	878887116
----	-----	---	-----------

196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
244	51	2	880606923
166	346	1	886397596

```
196 242 3 881250949  
186 302 3 891717742  
22 377 1 878887116  
244 51 2 880606923  
166 346 1 886397596
```

```
ratings = lines.map(lambda x: x.split()[2])
```

```
3  
3  
1  
2  
1
```

```
196 242 3 881250949  
186 302 3 891717742  
22 377 1 878887116  
244 51 2 880606923  
166 346 1 886397596
```

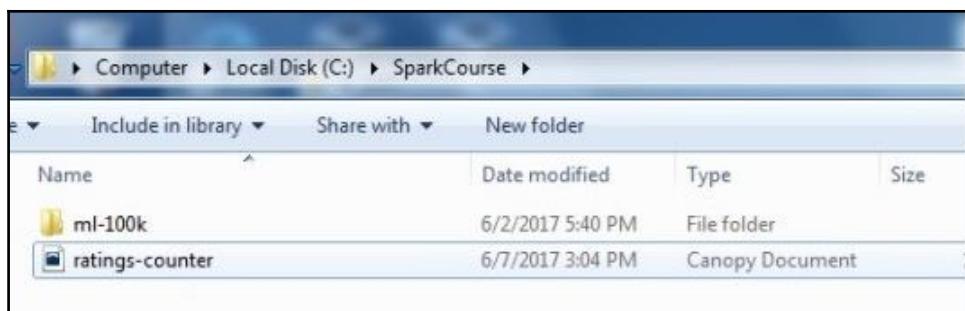
```
3  
3  
1  
2  
1
```

```
result = ratings.countByValue()
```

3		
3		
1	—————>	(3, 2)
2		(1, 2)
1		(2, 1)

(3, 2)
(1, 2)
(2, 1)

12
21
32



```
<User> c:\SparkCourse>spark-submit ratings-counter.py
1 6110
2 11370
3 27145
4 34174
5 21201
```

▷ Input Data: ID, name, age, number of friends

0, Will,33,385
1, Jean-Luc,33,2
2, Hugh,55,221
3, Deanna,40,465
4, QUARK,68,21

0, Will,33,385
1, Jean-Luc,33,2
2, Hugh,55,221
3, Deanna,40,465
4, Quark,68,21

33, 385
33, 2
55, 221
40, 465
...

(33, 385)=> (33, (385, 1))

(33, 385) => (33, (385, 1))

(33, 385)=> (33, (385, 1))
(33, 2) => (33, (2, 1))
(55, 221)=> (55, (221, 1))

(33, 385) => (33, (385, 1))
(33, 2)=> (33, (2, 1))
(55, 221)=> (55, (221, 1))

(33, (387, 2))

(33, (387, 2))=> (33, 193.5)

(33, (387, 2))=> (33, 193.5)

(33, (387, 2))=> (33, 193.5)

Name	Date modified	Type	Size
ml-100k	6/29/2017 12:37 PM	File folder	
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB

```
friends-by-age.py
1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("FriendsByAge")
4 sc = SparkContext(conf = conf)
5
6 def parseLine(line):
7     fields = line.split(',')
8     age = int(fields[2])
9     numFriends = int(fields[3])
10    return (age, numFriends)
11
12 lines = sc.textFile("file:///SparkCourse/fakefriends.csv")
13 rdd = lines.map(parseLine)
14 totalsByAge = rdd.mapValues(lambda x: (x, 1)).reduceByKey(lambda x, y: (x[0] + y[0], x[1] + y[1]))
15 averagesByAge = totalsByAge.mapValues(lambda x: x[0] / x[1])
16 results = averagesByAge.collect()
17 for result in results:
18     print(result)
19
```

```
<47, 233>
<48, 281>
<49, 184>
<50, 254>
<51, 302>
<52, 340>
<53, 222>
<54, 278>
<55, 295>
<56, 306>
<57, 258>
<58, 116>
<59, 220>
<60, 202>
<61, 256>
<62, 220>
<63, 384>
<64, 281>
<65, 298>
<66, 276>
<67, 214>
<68, 269>
<69, 235>

<User> c:\SparkCourse>
```

```
ITE00100554,18000101,TMAX,-75,,,E,
ITE00100554,18000101,TMIN,-148,,,E,
GM000010962,18000101,PRCP0,,,E,
EZE00100082,18000101,TMAX,-86,,,E,
EZE00100082,18000101,TMIN,-135,,,E,
```

```
ITE00100554,18000101,TMAX,-75,,,E,
ITE00100554,18000101,TMIN,-148,,,E,
GM000010962,18000101,PRCP0,,,E,
EZE00100082,18000101,TMAX,-86,,,E,
EZE00100082,18000101,TMIN,-135,,,E,
```

```
ITE00100554,18000101,TMIN,-148,,,E,
```

Name	Date modified	Type	Size
ml-100k	6/2/2017 5:40 PM	File folder	
1800	6/8/2017 12:23 PM	Microsoft Excel C...	62 KB
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
min-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB

```

1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("MinTemperatures")
4 sc = SparkContext(conf = conf)
5
6 def parseLine(line):
7     fields = line.split(',')
8     stationID = fields[0]
9     entryType = fields[2]
10    temperature = float(fields[3]) * 0.1 * (9.0 / 5.0) + 32.0
11    return (stationID, entryType, temperature)
12
13 lines = sc.textFile("file:///SparkCourse/1800.csv")
14 parsedLines = lines.map(parseLine)
15 minTemps = parsedLines.filter(lambda x: "TMIN" in x[1])
16 stationTemps = minTemps.map(lambda x: (x[0], x[2]))
17 minTemps = stationTemps.reduceByKey(lambda x, y: min(x,y))
18 results = minTemps.collect();
19
20 for result in results:
21     print(result[0] + "\t{:.2f}F".format(result[1]))
22

```

```
(User> c:\SparkCourse>spark-submit min-temperatures.py
```

```
(User> c:\SparkCourse>spark-submit min-temperatures.py
ITE00100554      5.36F
EZE00100082      7.70F
```

Name	Date modified	Type	Size
ml-100k	6/2/2017 5:40 PM	File folder	
1800	6/8/2017 12:23 PM	Microsoft Excel C...	62 KB
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
max-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
min-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB

```

1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("MaxTemperatures")
4 sc = SparkContext(conf = conf)
5
6 def parseLine(line):
7     fields = line.split(',')
8     stationID = fields[0]
9     entryType = fields[2]
10    temperature = float(fields[3]) * 0.1 * (9.0 / 5.0) + 32.0
11    return (stationID, entryType, temperature)
12
13 lines = sc.textFile("file:///SparkCourse/1800.csv")
14 parsedLines = lines.map(parseLine)
15 maxTemps = parsedLines.filter(lambda x: "TMAX" in x[1])
16 stationTemps = maxTemps.map(lambda x: (x[0], x[2]))
17 maxTemps = stationTemps.reduceByKey(lambda x, y: max(x,y))
18 results = maxTemps.collect();
19
20 for result in results:
21     print(result[0] + "\t{:.2f}F".format(result[1]))
22

```

```
(User) c:\SparkCourse>spark-submit max-temperatures.py
ITE00100554      90.14F
EZE00100082      90.14F
```

The quick red
fox jumped
over the lazy
brown dogs

```
lines = sc.textFile("redfox.txt")
rareCaps = lines.map(lambda x: x.upper())
```

THE QUICK RED
FOX JUMPED
OVER THE LAZY
BROWN DOGS

The quick red
fox jumped
over the lazy
brown dogs

```
lines = sc.textfile("redfox.txt")
words=lines.flatMap(lambda x: x.split())
```

THE
quick
red
fox
jumped
over
the
lazy
brown
dogs

SELF - EMPLOYMENT

**Building an Internet
Business of One**

**Frank
Kane**

**Achieving Financial and Personal Freedom
through a Lifestyle Business**



The screenshot shows a Windows File Explorer window with the following directory structure:

Name	Date modified	Type	Size
ml-100k	6/2/2017 5:40 PM	File folder	
1800	6/8/2017 12:23 PM	Microsoft Excel C...	62 KB
book	6/8/2017 12:23 PM	Text Document	259 KB
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
max-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
min-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB
word-count	6/8/2017 12:23 PM	Canopy Document	1 KB

```
1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("WordCount")
4 sc = SparkContext(conf = conf)
5
6 input = sc.textFile("file:///sparkcourse/book.txt")
7 words = input.flatMap(lambda x: x.split())
8 wordCounts = words.countByValue()
9
10 for word, count in wordCounts.items():
11     cleanWord = word.encode('ascii', 'ignore')
12     if (cleanWord):
13         print(cleanWord.decode() + " " + str(count))
14
```

```
DESIGNING 1
clients. 2
clients. 2
made 12
whether 21
this, 10
distract 1
this. 2
below 2
USING 1
this; 1
this: 2
intimidating 1
inadequate 1
meaningless 2
highlighted 1
kind. 1
avoided 1
improving 1
Sessions 1
other 71
incredibly 2
Banner 2
clicks 2
junk 1
kinds 3
webpage, 1
PLAN 3
S-corp 1
incurred 1
extort 1
click, 1
Company) 1
LEECHES 1
click" 1
Site 1
intentionally 2
entirely. 1

<User> c:\SparkCourse>
```

The screenshot shows a Windows File Explorer window with the following details:

- Path:** Computer > Local Disk (C:) > SparkCourse
- File List:**

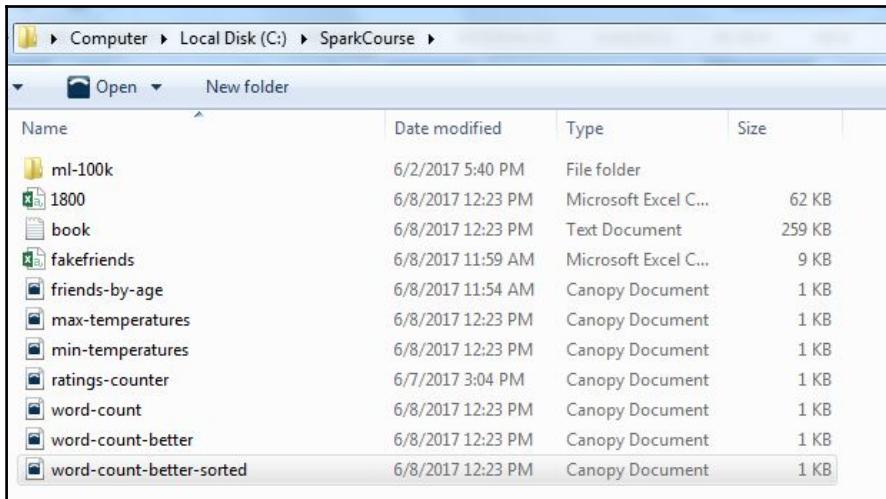
Name	Date modified	Type	Size
ml-100k	6/2/2017 5:40 PM	File folder	
1800	6/8/2017 12:23 PM	Microsoft Excel C...	62 KB
book	6/8/2017 12:23 PM	Text Document	259 KB
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
max-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
min-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
ries	6/7/2017 3:04 PM	Canopy Document	1 KB
ratings-counter	6/8/2017 12:23 PM	Canopy Document	1 KB
word-count	6/8/2017 12:23 PM	Canopy Document	1 KB
word-count-better	6/8/2017 12:23 PM	Canopy Document	1 KB

```

1 import re
2 from pyspark import SparkConf, SparkContext
3
4 def normalizeWords(text):
5     return re.compile(r'\W+', re.UNICODE).split(text.lower())
6
7 conf = SparkConf().setMaster("local").setAppName("WordCount")
8 sc = SparkContext(conf = conf)
9
10 input = sc.textFile("file:///sparkcourse/book.txt")
11 words = input.flatMap(normalizeWords)
12 wordCounts = words.countByValue()
13
14 for word, count in wordCounts.items():
15     cleanWord = word.encode('ascii', 'ignore')
16     if (cleanWord):
17         print(cleanWord.decode() + " " + str(count))
18

```

```
forgivable 1
details 5
normal 1
welcomes 1
mass 5
out 161
conversational 1
clicks 3
disposing 1
troll 1
junk 1
star 1
shown 4
variation 2
stay 7
chance 12
workaholic 1
spreadsheet 2
gap 2
friends 10
incurred 1
exposure 2
shock 1
ended 10
lasted 3
```



A screenshot of a Windows File Explorer window. The path in the address bar is "Computer > Local Disk (C:) > SparkCourse". The window shows a list of files and folders:

Name	Date modified	Type	Size
ml-100k	6/2/2017 5:40 PM	File folder	
1800	6/8/2017 12:23 PM	Microsoft Excel C...	62 KB
book	6/8/2017 12:23 PM	Text Document	259 KB
fakefriends	6/8/2017 11:59 AM	Microsoft Excel C...	9 KB
friends-by-age	6/8/2017 11:54 AM	Canopy Document	1 KB
max-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
min-temperatures	6/8/2017 12:23 PM	Canopy Document	1 KB
ratings-counter	6/7/2017 3:04 PM	Canopy Document	1 KB
word-count	6/8/2017 12:23 PM	Canopy Document	1 KB
word-count-better	6/8/2017 12:23 PM	Canopy Document	1 KB
word-count-better-sorted	6/8/2017 12:23 PM	Canopy Document	1 KB

```

1 import re
2 from pyspark import SparkConf, SparkContext
3
4 def normalizeWords(text):
5     return re.compile(r'\W+', re.UNICODE).split(text.lower())
6
7 conf = SparkConf().setMaster("local").setAppName("WordCount")
8 sc = SparkContext(conf = conf)
9
10 input = sc.textFile("file:///sparkcourse/book.txt")
11 words = input.flatMap(normalizeWords)
12
13 wordCounts = words.map(lambda x: (x, 1)).reduceByKey(lambda x, y: x + y)
14 wordCountsSorted = wordCounts.map(lambda x: (x[1], x[0])).sortByKey()
15 results = wordCountsSorted.collect()
16
17 for result in results:
18     count = str(result[0])
19     word = result[1].encode('ascii', 'ignore')
20     if (word):
21         print(word.decode() + ": \t\t" + count)
22

```

```

with:          315
have:          321
as:            343
be:            369
can:           376
business:      383
i:              387
s:              391
if:             411
are:            424
on:             428
for:            537
is:              560
in:              616
it:              649
that:            747
and:             934
of:              970
a:              1191
the:             1292
your:            1420
to:              1828
you:             1878

```

<User> c:\SparkCourse>

44,8602,37.19
35,5368,65.89
44,3391,40.64
47,6694,14.98
35,680,13.08

44,8602,37.19 35,5368,65.89 44,3391,40.64 47,6694,14.98 35,680,13.08	 ? 	44,77.83 35,78.97 47,14.98
--	---	----------------------------------

```

1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("SpendByCustomer")
4 sc = SparkContext(conf = conf)
5
6 def extractCustomerPricePairs(line):
7     fields = line.split(',')
8     return (int(fields[0]), float(fields[2]))
9
10 input = sc.textFile("file:///sparkcourse/customer-orders.csv")
11 mappedInput = input.map(extractCustomerPricePairs)
12 totalByCustomer = mappedInput.reduceByKey(lambda x, y: x + y)
13
14 results = totalByCustomer.collect();
15 for result in results:
16     print(result)
17 |

```

```

(77, 4327.729999999999)
(78, 4524.509999999999)
(79, 3790.570000000001)
(80, 4727.860000000001)
(81, 5112.709999999999)
(82, 4812.489999999998)
(83, 4635.799999999997)
(84, 4652.939999999999)
(85, 5503.43)
(86, 4908.81)
(87, 5206.4)
(88, 4830.549999999999)
(89, 4851.479999999999)
(90, 5290.409999999998)
(91, 4642.259999999999)
(92, 5379.280000000002)
(93, 5265.750000000001)
(94, 4475.569999999999)
(95, 4876.840000000002)
(96, 3924.230000000001)
(97, 5977.189999999995)
(98, 4297.260000000001)
(99, 4172.289999999998)


```

(User) c:\SparkCourse>

```
(84, 4652.939999999999)
```

```
1 |from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("SpendByCustomerSorted")
4 sc = SparkContext(conf = conf)
5
6 def extractCustomerPricePairs(line):
7     fields = line.split(',')
8     return (int(fields[0]), float(fields[2]))
9
10 input = sc.textFile("file:///sparkcourse/customer-orders.csv")
11 mappedInput = input.map(extractCustomerPricePairs)
12 totalByCustomer = mappedInput.reduceByKey(lambda x, y: x + y)
13
14 #Changed for Python 3 compatibility:
15 #flipped = totalByCustomer.map(lambda (x,y):(y,x))
16 flipped = totalByCustomer.map(lambda x: (x[1], x[0]))
17
18 totalByCustomerSorted = flipped.sortByKey()
19
20 results = totalByCustomerSorted.collect();
21 for result in results:
22     print(result)
23
```

```
(5368.24999999999, 70)
(5368.83, 43)
(5379.28000000002, 92)
(5397.879999999998, 6)
(5413.51000000001, 15)
(5415.15000000001, 63)
(5437.730000000005, 58)
(5496.05000000004, 32)
(5497.479999999998, 61)
(5503.43, 85)
(5517.24000000001, 8)
(5524.949999999998, 0)
(5637.62, 41)
(5642.89, 59)
(5696.84000000003, 42)
(5963.109999999999, 46)
(5977.189999999995, 97)
(5994.59, 2)
(5995.66000000003, 71)
(6065.389999999999, 54)
(6193.109999999999, 39)
(6206.199999999999, 73)
(6375.449999999997, 68)
```

```
(User) c:\SparkCourse>
```

Chapter 3

```
196 242 3 881250949  
186 302 3 891717742  
22 377 1 878887116  
244 51 2 880606923  
166 346 1 886397596  
298 474 4 884182806
```

```
(User> c:\SparkCourse>spark-submit popular-movies.py
```

```
(336, 79)  
(344, 405)  
(350, 204)  
(350, 313)  
(365, 222)  
(367, 172)  
(378, 117)  
(384, 237)  
(390, 98)  
(392, 72)  
(394, 56)  
(413, 127)  
(420, 174)  
(429, 121)  
(431, 300)  
(452, 1)  
(478, 288)  
(481, 286)  
(485, 294)  
(507, 181)  
(588, 100)  
(589, 258)  
(590, 50)
```

```
(User> c:\SparkCourse>
```

Name	Date modified	Type	Size
allbut.pl	6/2/2017 5:40 PM	PL File	1 KB
mku.sh	6/2/2017 5:40 PM	SH File	1 KB
README	6/2/2017 5:40 PM	File	7 KB
u	6/21/2017 1:46 PM	DATA File	2,031 KB
u.genre	6/2/2017 5:40 PM	GENRE File	1 KB
u.info	6/2/2017 5:40 PM	INFO File	1 KB
u	6/2/2017 5:40 PM	ITEM File	231 KB
u.occu		Open	
u.user		7-Zip	
u1.bas		CRC SHA	
u1.test		Edit with Notepad++	
u2.bas		Open with...	
u2.test		Symantec Endpoint Protection.cloud	
u3.bas		Restore previous versions	
u3.test		Send to	
u4.bas		Cut	
u4.test		Copy	
u5.bas		Create shortcut	
u5.test		Delete	
ua.bas		Rename	
ua.test		Properties	
ub.bas			
ub.test			

```

exact?Disclosure%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|1|0|
0
44|Dolores Claiborne (1994)|01-Jan-1994
||http://us.imdb.com/M/title-exact?Dolores%20Claiborne%20
(1994)|0|0|0|0|0|0|1|0|0|0|0|0|0|1|0|0
45|Eat Drink Man Woman (1994)|01-Jan-1994
||http://us.imdb.com/M/title-exact?Yinshi%20Nan%20Nu%20(1994)|0
0|0|0|0|1|0|1|0|0|0|0|0|0|0|0|0|0|0|0|0
46|Exotica (1994)|01-Jan-1994||http://us.imdb.com/M/title-
exact?Exotica%20(1994)|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0|0
47|Ed Wood (1994)|01-Jan-1994||http://us.imdb.com/M/title-
exact?Ed%20Wood%20(1994)|0|0|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0|0
48|Hoop Dreams (1994)|01-Jan-1994||http://us.imdb.com/M/title-
exact?Hoop%20Dreams%20(1994)|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0
49|I.Q. (1994)|01-Jan-1994||http://us.imdb.com/M/title-exact?
I.Q.%20(1994)|0|0|0|0|0|1|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0
50|Star Wars (1977)|01-Jan-1977||http://us.imdb.com/M/title-
exact?Star%20Wars%20(1977)|0|1|1|0|0|0|0|0|0|0|0|0|0|1|1|0|1
|0
51|Legends of the Fall (1994)|01-Jan-1994
||http://us.imdb.com/M/title-exact?Legends%20of%20the%20Fall%20(1994)|0|0|0|0|0|0|0|1|0|0|0|0|0|0|1|0|0|1|1
52|Madness of King George, The (1994)|01-Jan-1994
||http://us.imdb.com/M/title-exact?Madness%20of%20King%20George,%20The%20(1994)|0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0
53|Natural Born Killers (1994)|01-Jan-1994
||http://us.imdb.com/M/title-exact?Natural%20Born%20Killers%20(1994)|0|1|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0
54|Outbreak (1995)|01-Jan-1995||http://us.imdb.com/M/title-
exact?Outbreak%20(1995)|0|1|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0|0|0|0
55|Professional, The (1994)|01-Jan-1994

```

```

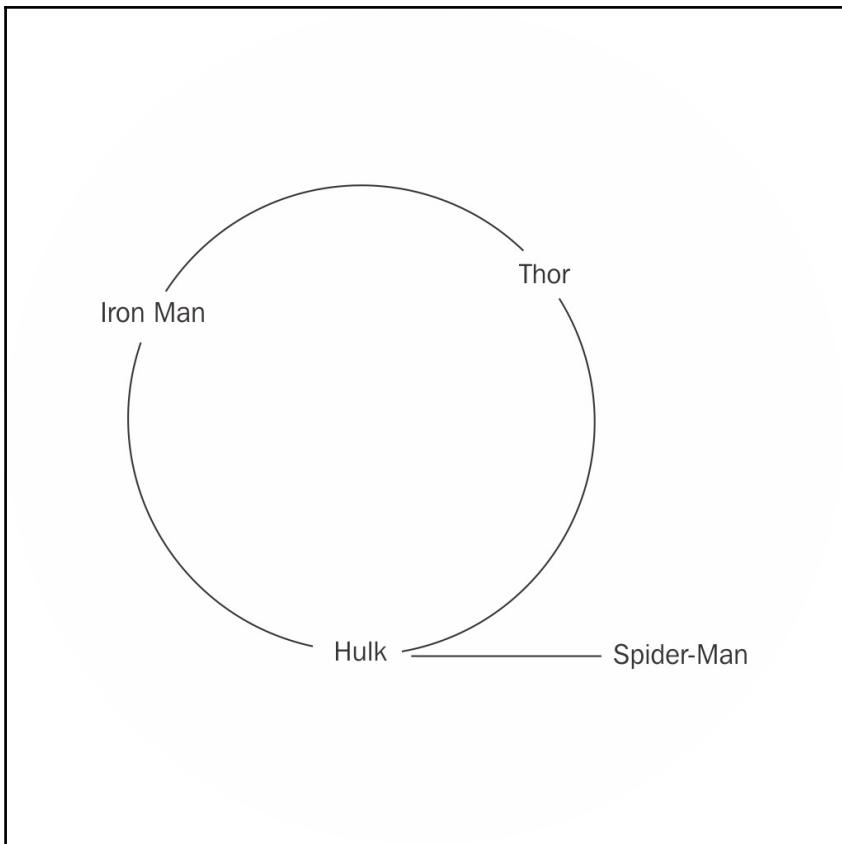
1 from pyspark import SparkConf, SparkContext
2
3 def loadMovieNames():
4     movieNames = {}
5     with open("ml-100k/u.ITEM") as f:
6         for line in f:
7             fields = line.split('|')
8             movieNames[int(fields[0])] = fields[1]
9     return movieNames
10
11 conf = SparkConf().setMaster("local").setAppName("PopularMovies")
12 sc = SparkContext(conf = conf)
13
14 nameDict = sc.broadcast(loadMovieNames())
15
16 lines = sc.textFile("file:///SparkCourse/ml-100k/u.data")
17 movies = lines.map(lambda x: (int(x.split()[1]), 1))
18 movieCounts = movies.reduceByKey(lambda x, y: x + y)
19
20 flipped = movieCounts.map( lambda (x, y) : (y, x))
21 sortedMovies = flipped.sortByKey()
22
23 sortedMoviesWithNames = sortedMovies.map(lambda (count, movie) : (nameDict.value[movie], count))
24
25 results = sortedMoviesWithNames.collect()
26
27 for result in results:
28     print(result)
29

```

```
[User] c:\SparkCourse>spark-submit popular-movies-nicer.py
```

```
('Fugitive, The (1993)', 336)
('Mission: Impossible (1996)', 344)
('Back to the Future (1985)', 350)
('Titanic (1997)', 358)
('Star Trek: First Contact (1996)', 365)
('Empire Strikes Back, The (1980)', 367)
('Rock, The (1996)', 378)
('Jerry Maguire (1996)', 384)
('Silence of the Lambs, The (1991)', 398)
('Twelve Monkeys (1995)', 392)
('Pulp Fiction (1994)', 394)
('Godfather, The (1972)', 413)
('Raiders of the Lost Ark (1981)', 428)
('Independence Day (ID4) (1996)', 429)
('Air Force One (1997)', 431)
('Toy Story (1995)', 452)
('Scream (1996)', 478)
('English Patient, The (1996)', 481)
('Liar Liar (1997)', 485)
('Return of the Jedi (1983)', 507)
('Fargo (1996)', 508)
('Contact (1997)', 509)
('Star Wars (1977)', 583)
```

```
[User] c:\SparkCourse>
```



4395 2237 1767 472 4997 5931 6235 1478 1369 806 3994 6232
3519 4704 2460 763 1602 5306 5358 6121 6160 2459 3173 4963 6166
3518 5409

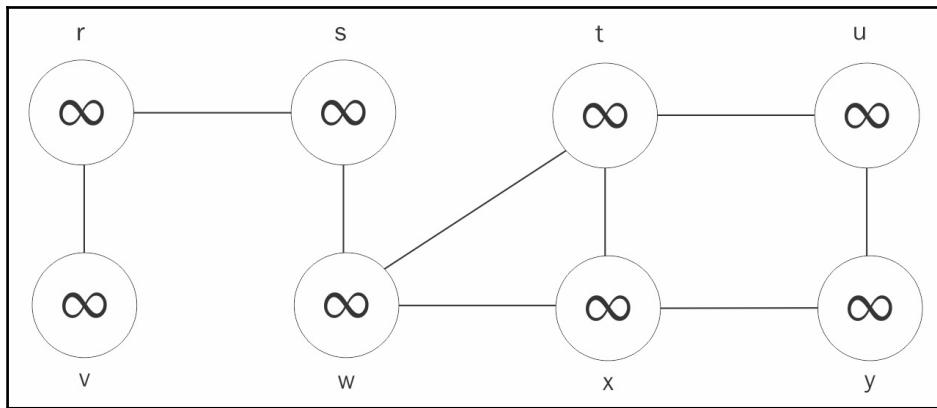
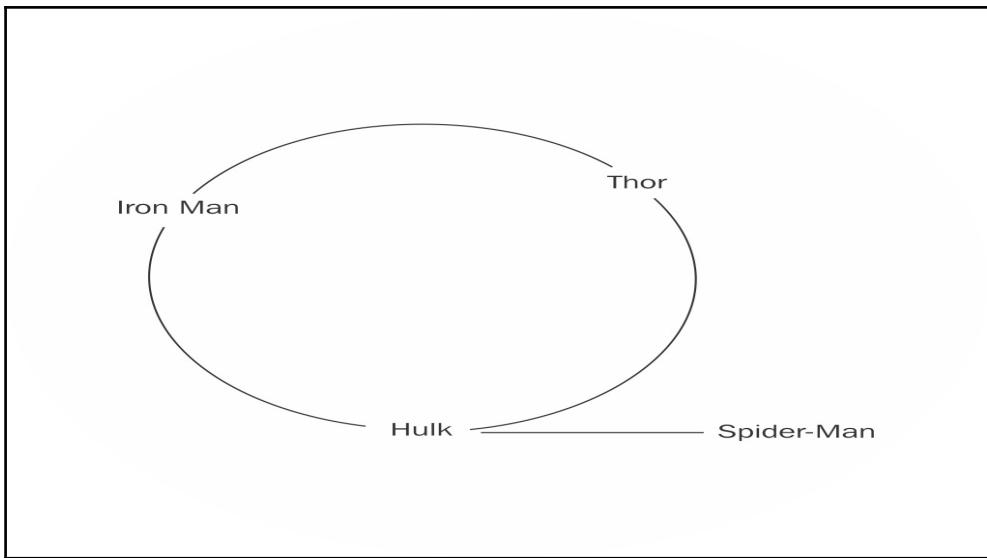
5300 "SPENCER, TRACY"
5301 "SPERZEL, ANTON"
5302 "SPETSBURO, GEN. YURI"
5303 "SPHINX"
5304 "SPHINX II"
5305 "SPHINX III"
5306 "SPIDER-MAN/PETER PAR"
5307 "SPIDER-MAN III/MARTH"
5308 "SPIDER CLONE/BEN"
5309 "SPIDER-WOMAN/JESSICA"

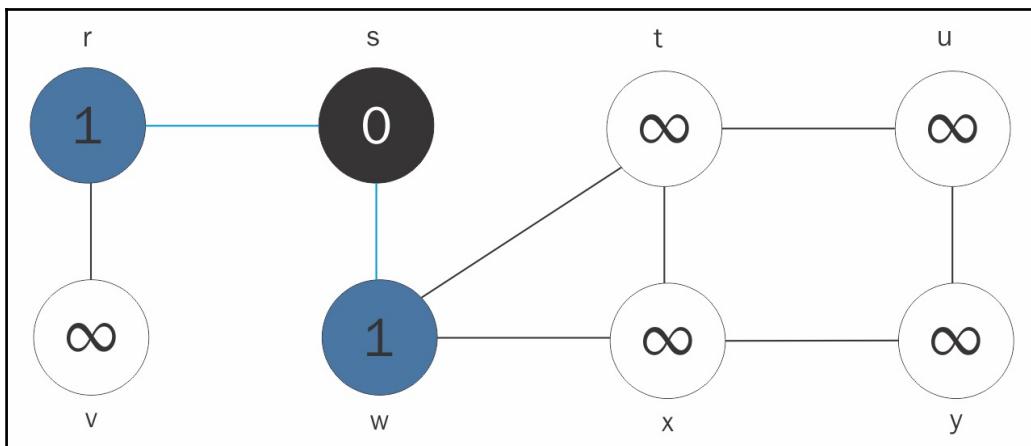
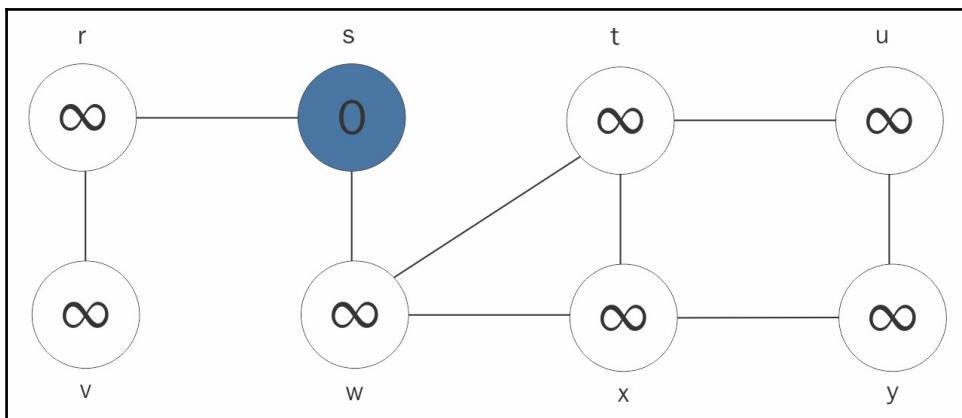
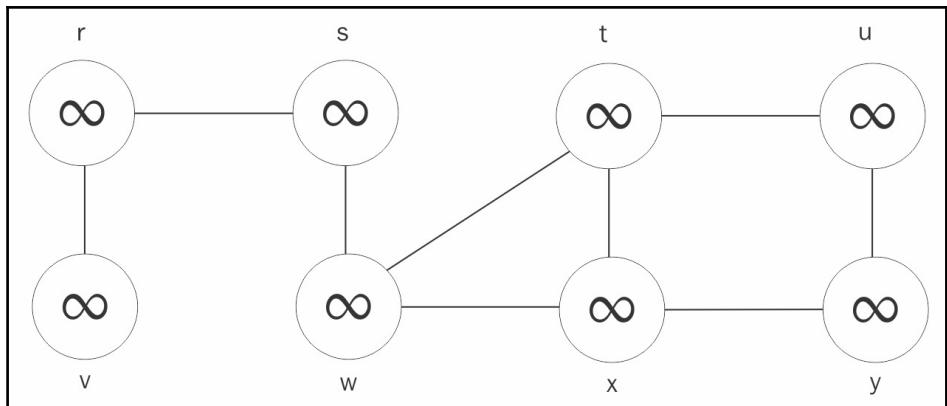
```
most-popular-superhero.py
1 from pyspark import SparkConf, SparkContext
2
3 conf = SparkConf().setMaster("local").setAppName("PopularHero")
4 sc = SparkContext(conf = conf)
5
6 def countCoOccurrences(line):
7     elements = line.split()
8     return (int(elements[0]), len(elements) - 1)
9
10 def parseNames(line):
11     fields = line.split(',')
12     return (int(fields[0]), fields[1].encode("utf8"))
13
14 names = sc.textFile("file:///SparkCourse/marvel-names.txt")
15 namesRdd = names.map(parseNames)
16
17 lines = sc.textFile("file:///SparkCourse/marvel-graph.txt")
18
19 pairings = lines.map(countCoOccurrences)
20 totalFriendsByCharacter = pairings.reduceByKey(lambda x, y : x + y)
21 flipped = totalFriendsByCharacter.map(lambda (x,y) : (y,x))
22
23 mostPopular = flipped.max()
24
25 mostPopularName = namesRdd.lookup(mostPopular[1])[0]
26
27 print(mostPopularName + " is the most popular superhero, with " +
28       str(mostPopular[0]) + " co-appearances.")
29
```

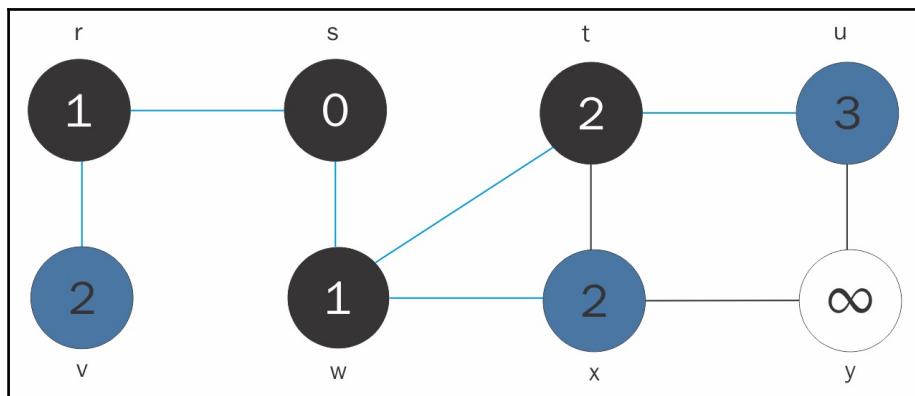
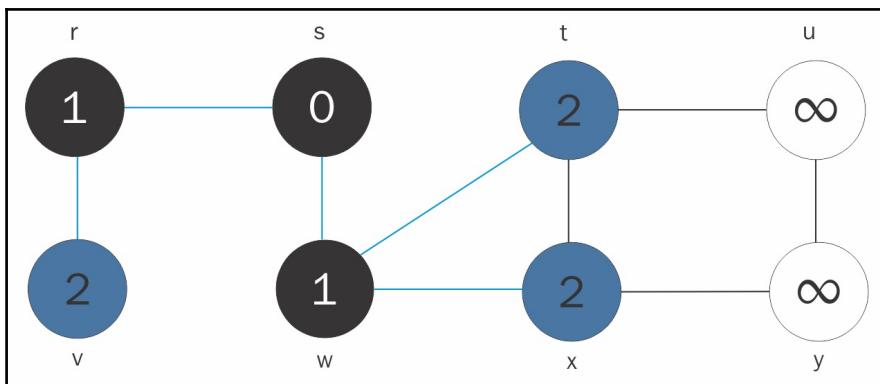
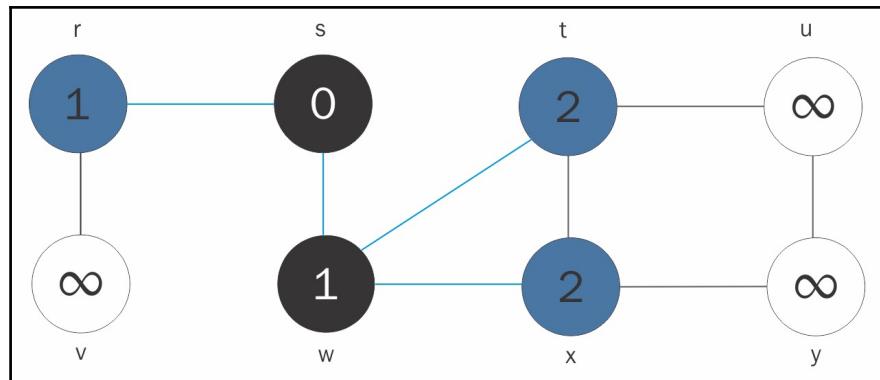
```
<User> c:\SparkCourse>spark-submit most-popular-superhero.py
```

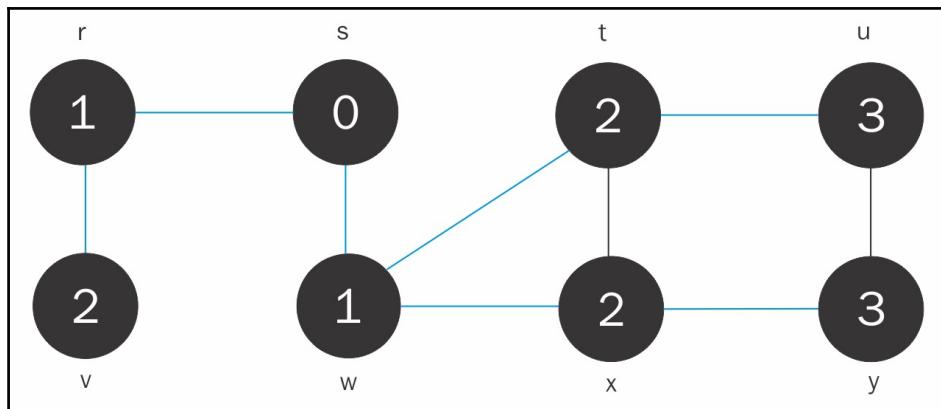
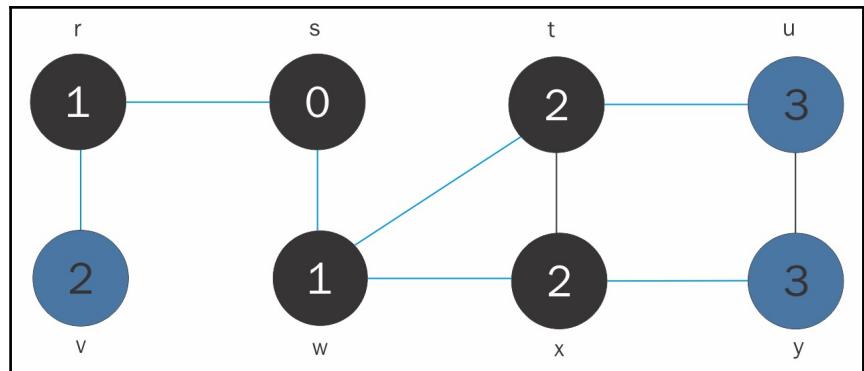
```
CAPTAIN AMERICA is the most popular superhero, with 1933 co-appearances.
```

```
<User> c:\SparkCourse>
```









5983 1165 3836 4361 1282

(5983, (1165, 3836, 4361, 1282), 9999, WHITE)

(5983, (1165, 3836, 4361, 1282), 9999, WHITE)

```

1 #Boilerplate stuff:
2 from pyspark import SparkConf, SparkContext
3
4 conf = SparkConf().setMaster("local").setAppName("DegreesOfSeparation")
5 sc = SparkContext(conf = conf)
6
7 # The characters we wish to find the degree of separation between:
8 startCharacterID = 5306 #SpiderMan
9 targetCharacterID = 14 #ADAM 3,031 (who?)
10
11 # Our accumulator, used to signal when we find the target character during
12 # our BFS traversal.
13 hitCounter = sc.accumulator(0)
14
15 def convertToBFS(line):
16     fields = line.split()
17     heroID = int(fields[0])
18     connections = []
19     for connection in fields[1:]:
20         connections.append(int(connection))
21
22     color = 'WHITE'
23     distance = 9999
24
25     if (heroID == startCharacterID):
26         color = 'GRAY'
27         distance = 0
28
29     return (heroID, (connections, distance, color))
30
31
32 def createStartingRdd():
33     inputFile = sc.textFile("file:///sparkcourse/marvel-graph.txt")
34     return inputFile.map(convertToBFS)
35
36 def bfsMap(node):
37     characterID = node[0]
38     data = node[1]
39     connections = data[0]

```

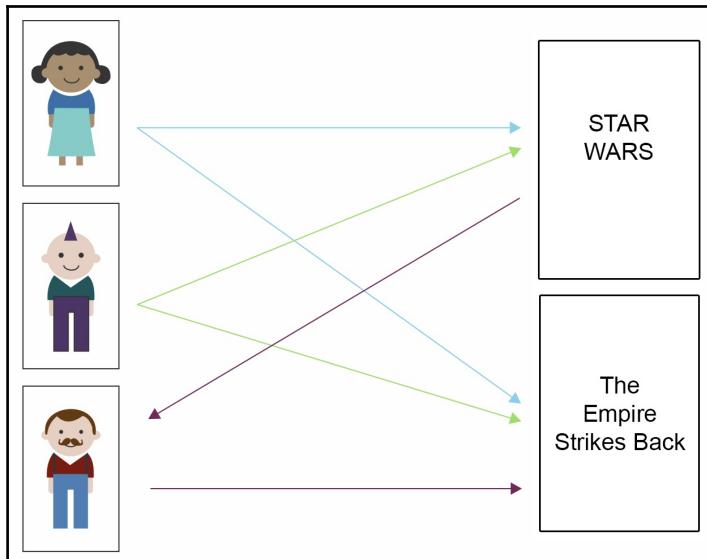
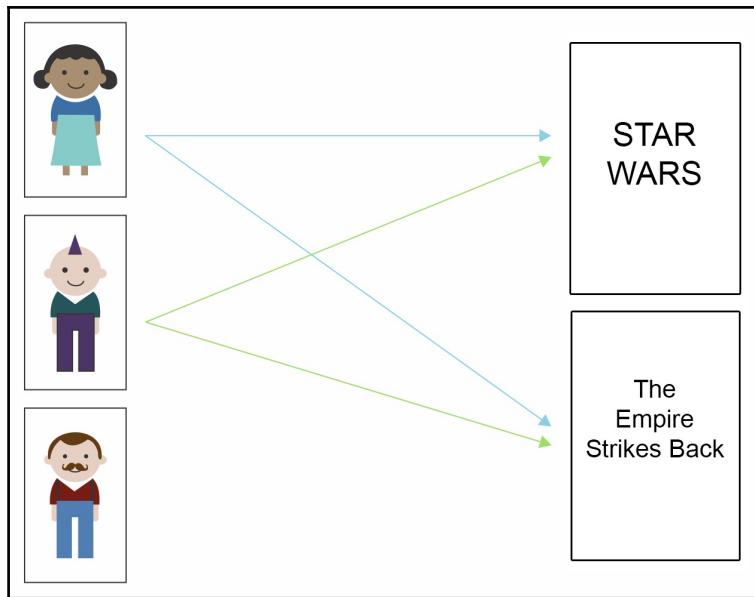
```
(User) c:\SparkCourse>spark-submit degrees-of-separation.py
```

```

Running BFS iteration# 1
Processing 8330 values.
Running BFS iteration# 2
Processing 220615 values.
Hit the target character! From 1 different direction(s).

```

```
(User) c:\SparkCourse>
```



```
(User) c:\SparkCourse>spark-submit movie-similarities.py 50
```

```
Loading movie names...
[Stage 0:>                                <(0 + 0) / 4]
[Stage 0:>                                <(0 + 4) / 4]
[Stage 0:======>                         <(2 + 2) / 4]
[Stage 0:======>                         <(3 + 1) / 4]
[Stage 1:>                                <(0 + 4) / 4]
[Stage 1:======>                         <(1 + 3) / 4]
[Stage 1:======>                         <(2 + 2) / 4]
[Stage 1:======>                         <(3 + 1) / 4]
[Stage 2:>                                <(0 + 4) / 4]
[Stage 2:======>                         <(3 + 1) / 4]

[Stage 5:>                                <(0 + 4) / 4]
[Stage 5:======>                         <(2 + 2) / 4]
[Stage 5:======>                         <(3 + 1) / 4]

[Stage 8:>                                <(0 + 4) / 4]
[Stage 8:======>                         <(1 + 3) / 4]
[Stage 8:======>                         <(3 + 1) / 4]
```

```
Top 10 similar movies for Star Wars (1977)
Empire Strikes Back, The (1980) score: 0.989552207839 strength: 345
Return of the Jedi (1983) score: 0.985723086125 strength: 480
Raiders of the Lost Ark (1981) score: 0.981760098873 strength: 380
20,000 Leagues Under the Sea (1954) score: 0.97893856055 strength: 68
12 Angry Men (1957) score: 0.972657612045 strength: 109
Close Shave, A (1995) score: 0.977594829105 strength: 92
African Queen, The (1951) score: 0.976469222267 strength: 138
Sting, The (1973) score: 0.975151293774 strength: 204
Wrong Trousers, The (1993) score: 0.974868135546 strength: 103
Wallace & Gromit: The Best of Aardman Animation (1996) score: 0.97418161283
strength: 58
```

```
<User> c:\SparkCourse>
```

Chapter 4

The screenshot shows the AWS homepage with a blue header bar. The header includes the AWS logo, navigation links for Menu, Products, Solutions, Pricing, Software, More, English, My Account, and a yellow "Sign In to the Console" button.

Main content area:

- Try AWS with a 10-Minute Tutorial**: "Hello, World!" technical documents to help you get hands-on with AWS. [View Tutorials](#).
- Get Started with AWS for Free**: [Create a Free Account](#).
 - Amazon S3**: 5GB storage, 20k Get requests and 2k Put requests. [View AWS Free Tier Details](#).
- AWS Database Migration Service**: Easily migrate and convert databases. **OVER 28,000 DATABASES MIGRATED**.
- Run Microservices on AWS**: Learn to build and scale containerized microservices with Amazon ECS.
- AWS TechChat**: Stay informed of the latest round up of AWS news and announcements. [Subscribe to AWS TechChat](#).
- Windows on AWS**: Download whitepapers and join live webinars.

Secure | https://aws.amazon.com/ec2/spot/pricing/

Menu  Products Solutions Pricing Software More English My Account Sign In to the Console

PRODUCTS & SERVICES

- Amazon EC2 Spot Instances >
- Product Details >
- Pricing > Spot instances provide you with access to unused Amazon EC2 capacity at steep discounts relative to On-Demand prices. The Spot price fluctuates based on the supply and demand of available unused EC2 capacity.
- Getting Started >
- Spot Bid Advisor >
- FAQs >
- Testimonials >

RELATED LINKS

- Amazon EC2
- Amazon EC2 Purchasing Options
- AWS Documentation - Spot Instances

Manage Your Resources Sign In to the Console

Amazon EC2 Spot Instances Pricing

Spot instances provide you with access to unused Amazon EC2 capacity at steep discounts relative to On-Demand prices. The Spot price fluctuates based on the supply and demand of available unused EC2 capacity.

When you request Spot instances, you specify the maximum Spot price you are willing to pay. Your Spot instance is launched when the Spot price is lower than the price you specified, and will continue to run until you choose to terminate it or the Spot price exceeds the maximum price you specified.

With Spot instances, you will never be charged more than the maximum price you specified. While your instance runs, you are charged the Spot price that is in effect for that period. If the Spot price exceeds your specified price, your instance will receive a two-minute notification before it is terminated, and you will not be charged for the partial hour that your instance has run.

If you include a duration requirement with your Spot instances request, your instance will continue to run until you choose to terminate it, or until the specified duration has ended; your instance will not be terminated due to changes in the Spot price.

To compare the current Spot prices against standard On-Demand rates, visit the [Spot Bid Advisor](#).

Manage Your AWS Resources Sign in to the Console

t2.small	1	Variable	2	EBS Only	\$0.026 per Hour
t2.medium	2	Variable	4	EBS Only	\$0.052 per Hour
t2.large	2	Variable	8	EBS Only	\$0.104 per Hour
m4.large	2	6.5	8	EBS Only	\$0.126 per Hour
m4.xlarge	4	13	16	EBS Only	\$0.252 per Hour
m4.2xlarge	8	26	32	EBS Only	\$0.504 per Hour
m4.4xlarge	16	53.5	64	EBS Only	\$1.008 per Hour
m4.10xlarge	40	124.5	160	EBS Only	\$2.52 per Hour
m3.medium	1	3	3.75	1 x 4 SSD	\$0.067 per Hour
m3.large	2	6.5	7.5	1 x 32 SSD	\$0.133 per Hour
m3.xlarge	4	13	15	2 x 40 SSD	\$0.266 per Hour
m3.2xlarge	8	26	30	2 x 80 SSD	\$0.532 per Hour

Compute Optimized - Current Generation

m3.xlarge	4	13	15	2 x 40 SSD	\$0.266 per Hour
-----------	---	----	----	------------	------------------

Secure | https://aws.amazon.com/ec2/spot/pricing/

Menu  Products Solutions Pricing Software More English My Account Sign In to the Console

Pricing	Spot Instances	Defined Duration for Linux	Defined Duration for Windows
Getting Started >	Loading pricing data...		
Spot Bid Advisor >	Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of AWS is subject to Japanese Consumption Tax. Learn more.		
FAQs >			
Testimonials >			

Secure | https://console.aws.amazon.com/console/home?region=us-east-1

Services Resource Groups

AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage)

Recently visited services

All services

- Compute
 - EC2
 - EC2 Container Service
 - Lightsail
 - Elastic Beanstalk
 - Lambda
 - Batch
- Storage
 - S3
 - EFS
 - Glacier
 - Storage Gateway
- Database
 - RDS
 - DynamoDB
 - ElastiCache
 - Redshift
- Networking & Content
 - Delivery
 - VPC
 - CloudFront
 - Direct Connect
 - Route 53
- Developer Tools
 - CodeStar
 - CodeCommit
 - CodeBuild
 - CodeDeploy
 - CodePipeline
 - X-Ray
- Management Tools
 - CloudWatch
 - CloudFormation
 - CloudTrail
 - Config
 - OpsWorks
 - Service Catalog
 - Trusted Advisor
 - Managed Services
- Internet of Things
 - AWS IoT
 - AWS Greengrass
- Contact Center
 - Amazon Connect
- Game Development
 - Amazon GameLift
- Mobile Services
 - Mobile Hub
 - Cognito
 - Device Farm
 - Mobile Analytics
 - Pinpoint
- Security, Identity & Compliance
 - IAM
 - Inspector
 - Certificate Manager
 - Directory Service
 - WAF & Shield
 - Artifact
- Application Services
 - Step Functions
 - SWF
 - API Gateway
 - Elastic Transcoder
- Messaging
 - Simple Queue Service

Helpful tips

Manage your costs
Get real-time billing alerts based on your cost and usage budgets. Start now

Create an organization
Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

Explore AWS

New Product Announcements
View the latest announcements from the AWS Summit - San Francisco. Learn more

Migrate from Oracle to Amazon Aurora
Learn how to migrate from Oracle to Amazon Aurora with minimal downtime. View project

Introducing Amazon Kinesis Analytics
Easily process real-time, streaming data with Amazon Kinesis Analytics. Learn more

AWS Marketplace
Discover, procure, and deploy popular software products that run on AWS. Learn more



Secure | https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#KeyPairs:sort=keyName

Services Resource Groups

Create Key Pair Import Key Pair Delete

Filter by attributes or search by keyword ? 1 to 3 of 3

Key pair name	Fingerprint
Aditya_a_s_windows	9c:86:6c:b8:71:fb:a2:96:b6:e6:87:3e:1d:71:b3:bf:d3:f5:1d:00
CT01	31:35:8b:b3:f4:a7:5b:f1:83:59:d5:22:97:00:3d:17:10:73:ab:ae
zpush1	d5:a8:2a:29:97:a3:96:f6:af:cb:e2:7c:5d:6f:14:18:10:e9:38:98

Select a key pair

EC2 Dashboard Events Tags Reports Limits

Instances Spot Requests Reserved Instances Scheduled Instances Dedicated Hosts

AMIs Bundle Tasks

Volumes Snapshots

Security Groups Elastic IPs Placement Groups

Key Pairs

Network Interfaces

[Secure | https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)

Download PuTTY: latest release (0.69)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)
Download: [Stable](#) · [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.69, released on 2017-04-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.69 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [development snapshots](#), to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

MSI ("Windows Installer")

32-bit:	putty-0.69-installer.msi	(or by FTP)	(signature)
64-bit:	putty-64bit-0.69-installer.msi	(or by FTP)	(signature)

Unix source archive

.tar.gz:	putty-0.69.tar.gz	(or by FTP)	(signature)
----------	-----------------------------------	------------------------------	-----------------------------

Alternative binary files

The installer packages above will provide all of these (except PuTTYtel), but you can download them one by one if you prefer.

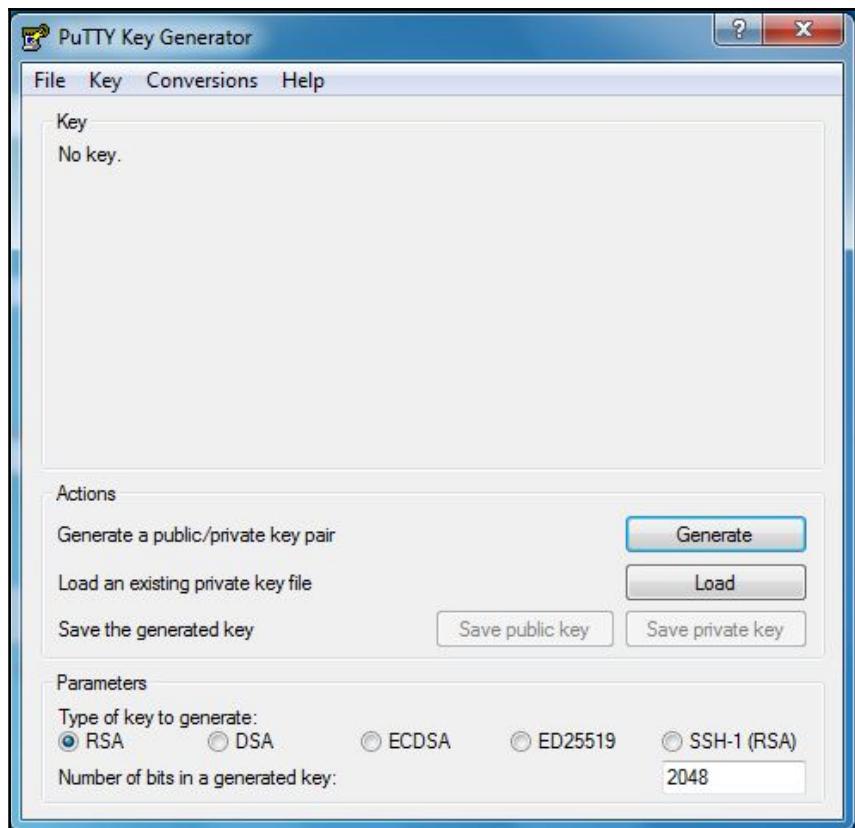
(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

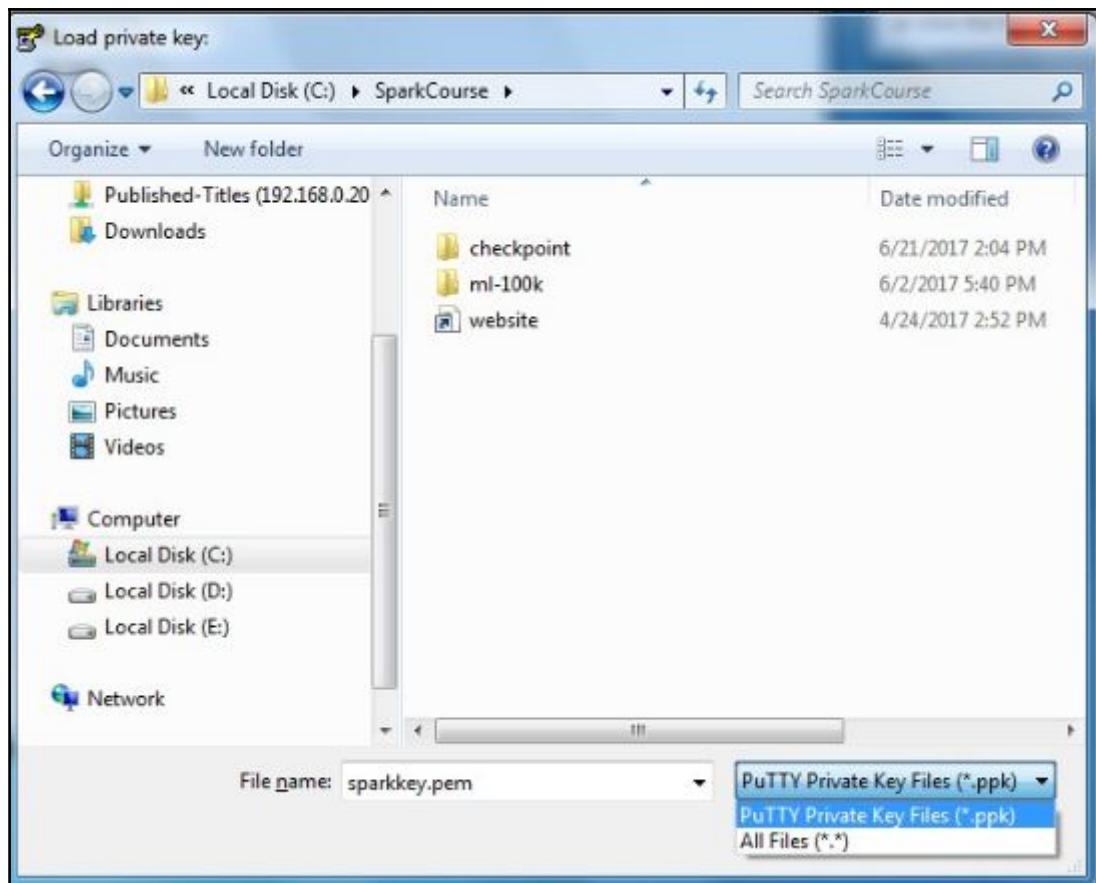
putty.exe (the SSH and Telnet client itself)

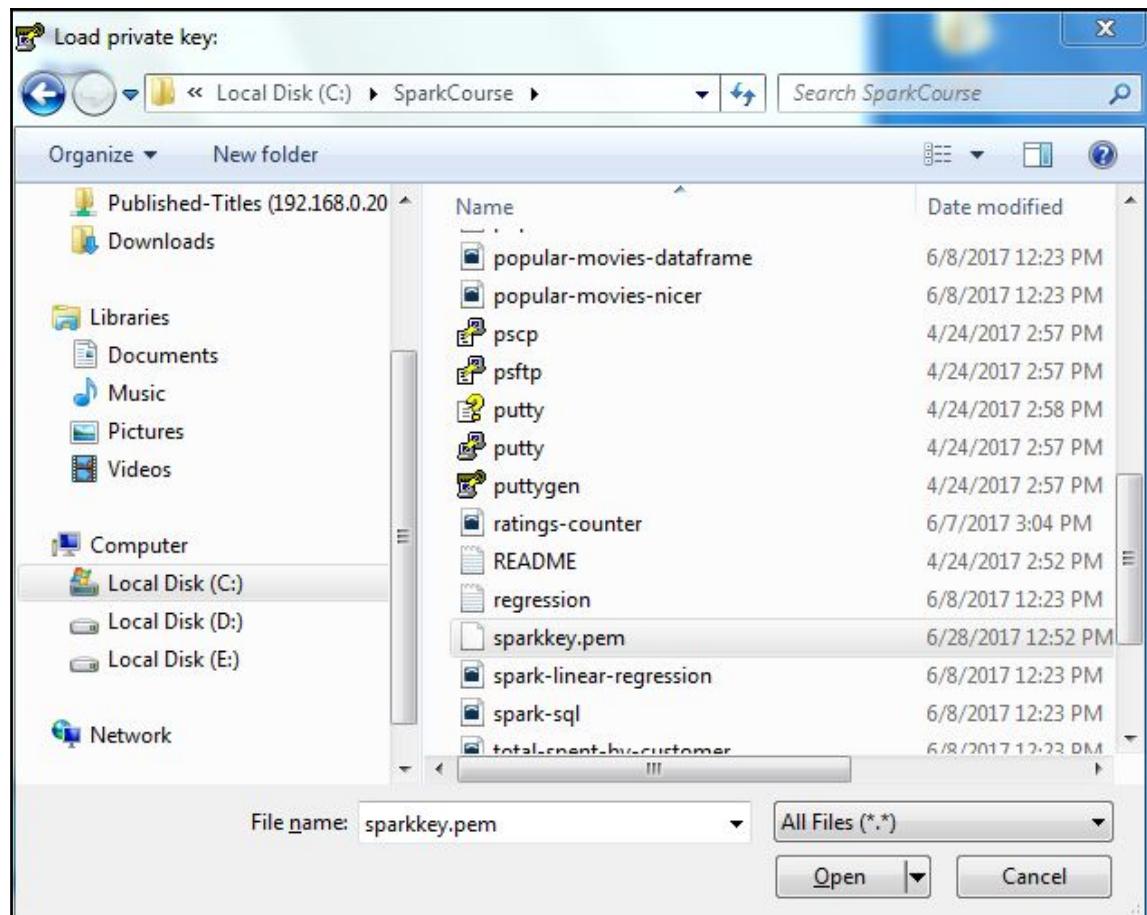
32-bit:	putty.exe	(or by FTP)	(signature)
64-bit:	putty.exe	(or by FTP)	(signature)

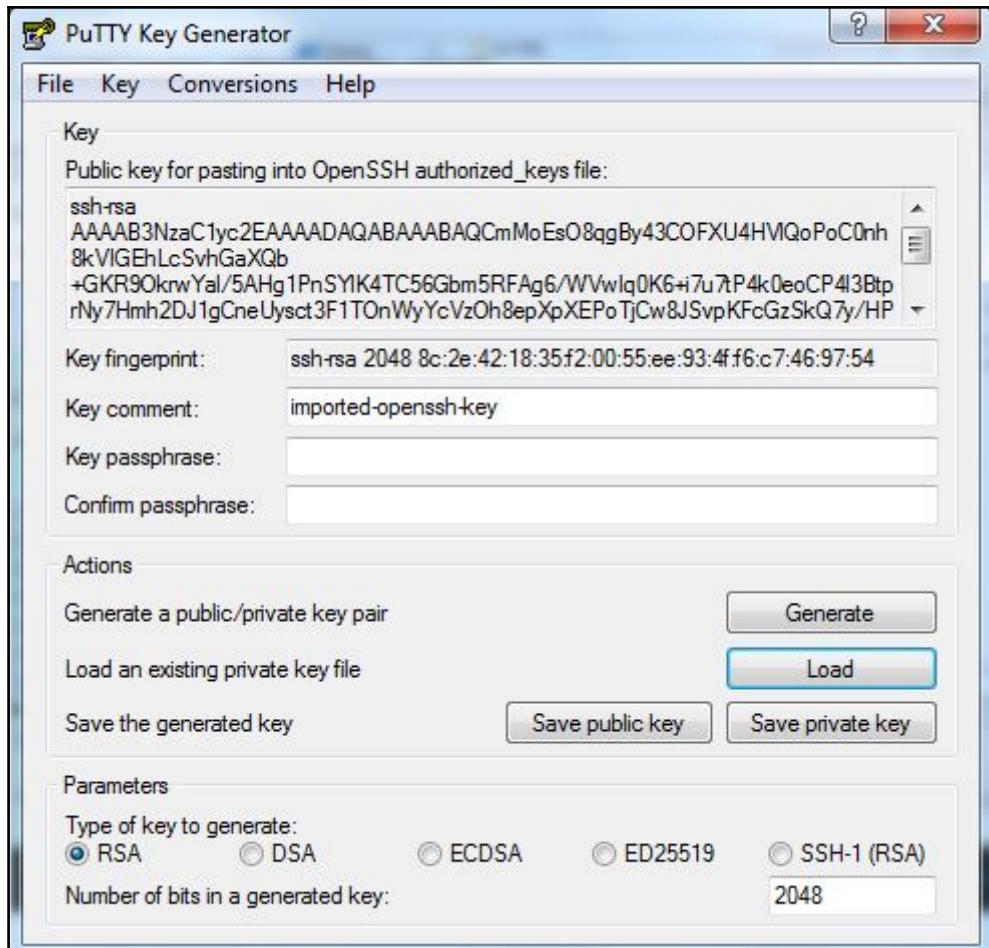
pscp.exe (an SCP client, i.e. command-line secure file copy)

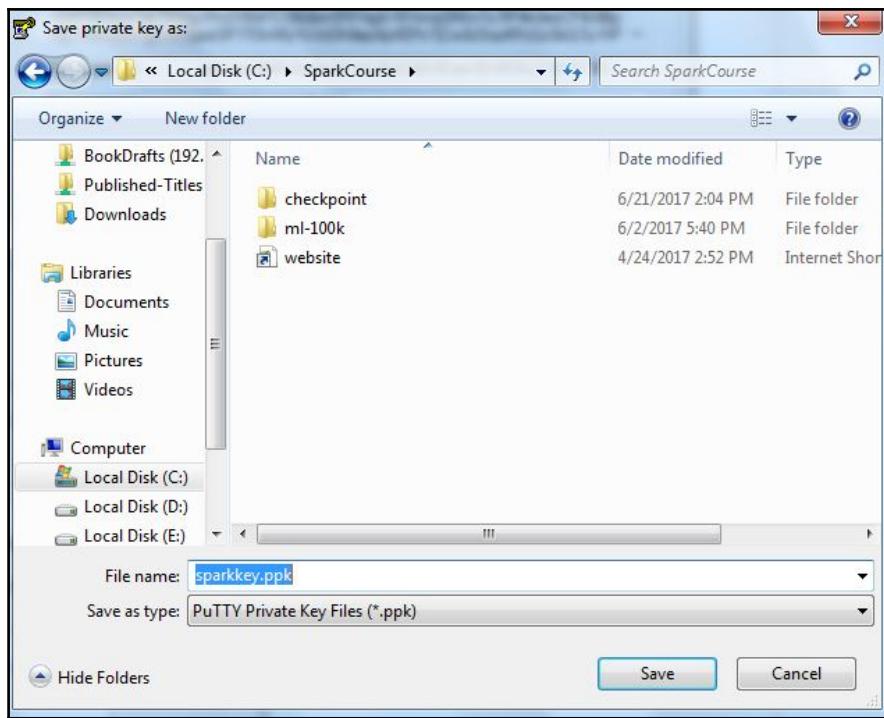
32-bit:	pscp.exe	(or by FTP)	(signature)
64-bit:	pscp.exe	(or by FTP)	(signature)

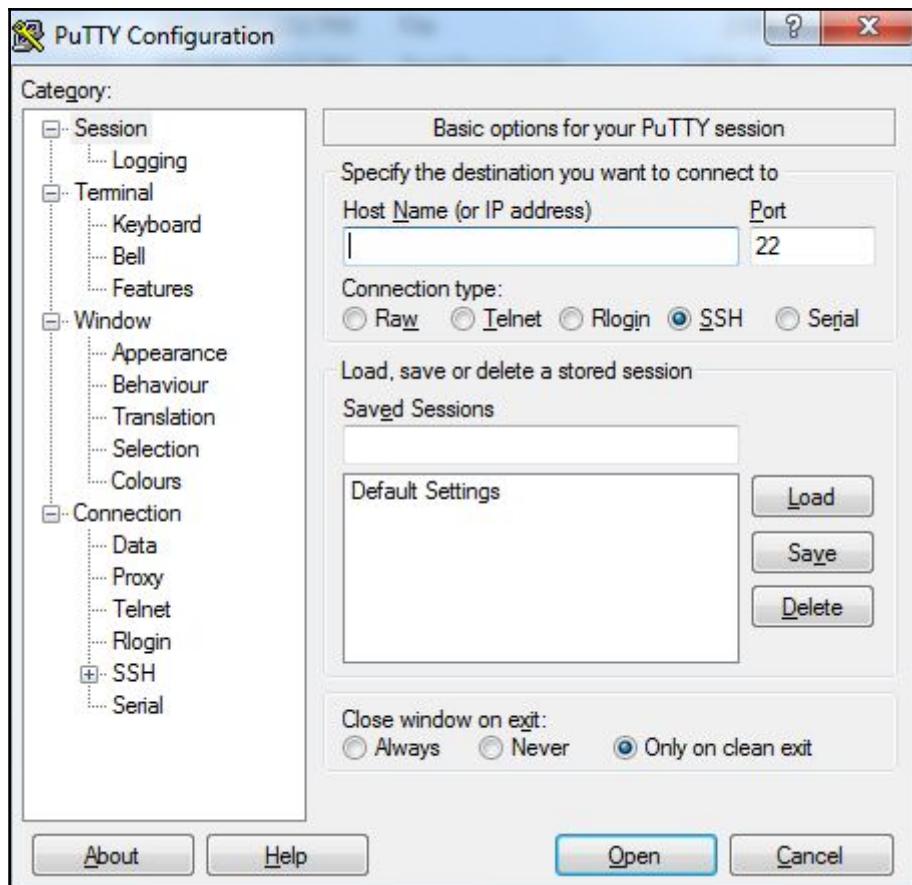


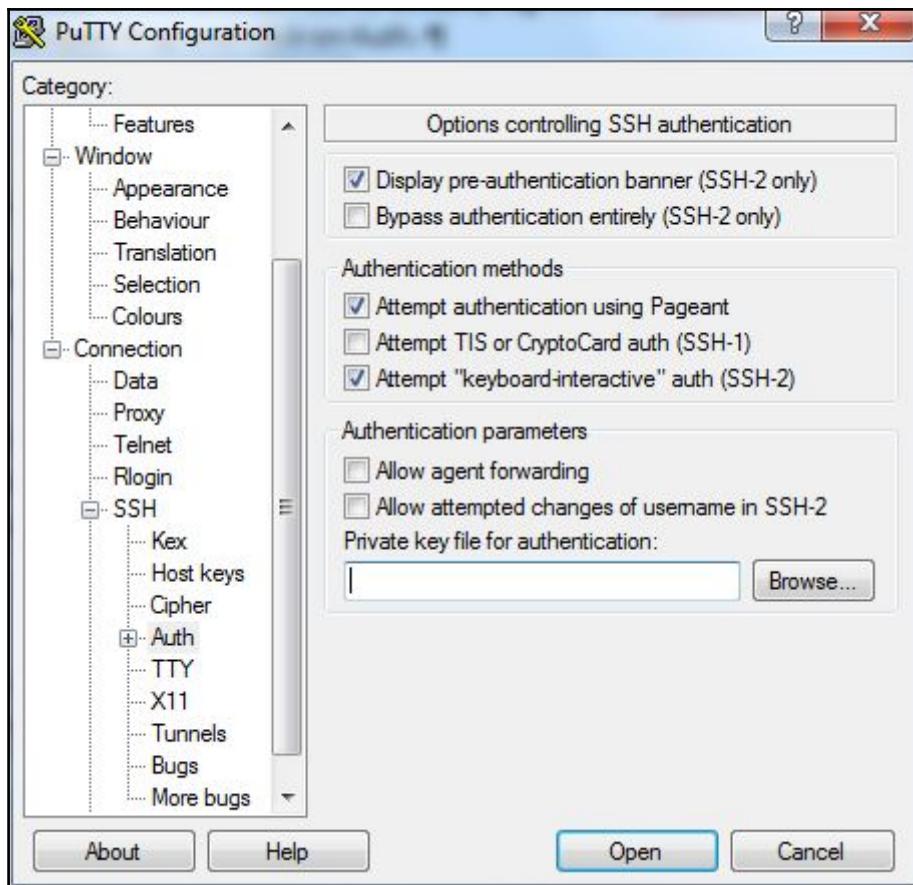


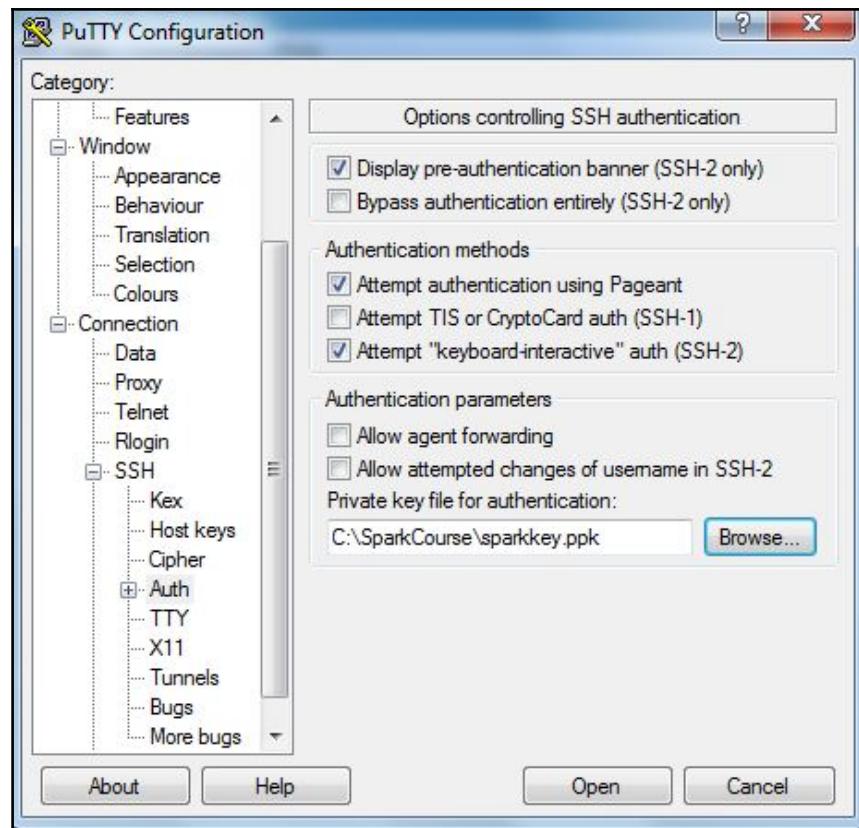


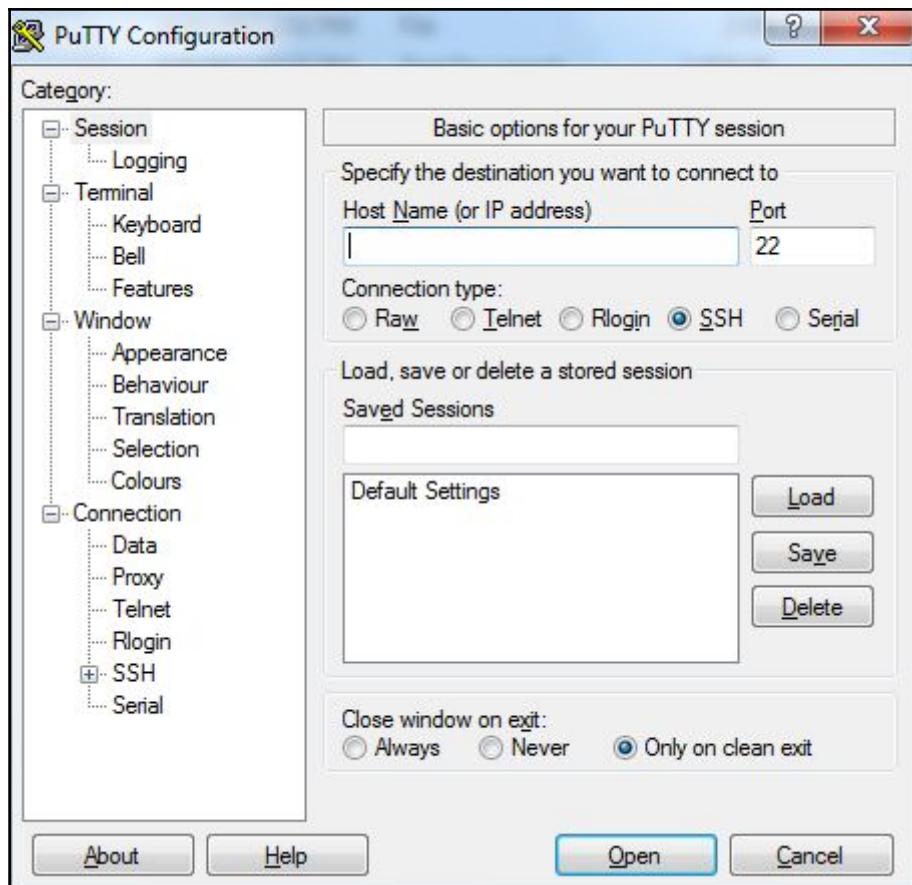












```

import sys
from pyspark import SparkConf, SparkContext
from math import sqrt

#To run on EMR successfully + output results for Star Wars:
#aws s3 cp s3://sundog-spark/MovieSimilarities1M.py ./
#aws s3 cp s3://sundog-spark/ml-1m/movies.dat ./
#spark-submit --executor-memory 1g MovieSimilarities1M.py 260

def loadMovieNames():
    movieNames = {}
    with open("movies.dat") as f:
        for line in f:
            fields = line.split("::")
            movieNames[int(fields[0])] = fields[1].decode('ascii', 'ignore')
    return movieNames

def makePairs((user, ratings)):
    (movie1, rating1) = ratings[0]
    (movie2, rating2) = ratings[1]
    return ((movie1, movie2), (rating1, rating2))

```

The screenshot shows a web browser window with the URL <https://grouplens.org>. The page has a blue header bar with the GroupLens logo and navigation links for 'about', 'datasets', 'publications', and 'blog'. Below the header, the main content area features a large white title card with the text 'Social Computing Research at the University of Minnesota' and a subtitle explaining GroupLens's mission. At the bottom of the page, there is a 'Featured Research' section and a paragraph about recent articles.

groupLens [about](#) [datasets](#) [publications](#) [blog](#)

Social Computing Research at the University of Minnesota

GroupLens advances the theory and practice of social computing by building and understanding systems used by real people

Featured Research

We publish research articles in conferences and journals primarily in the field of computer science, but also in other fields including psychology, sociology, and medicine. See our [blog](#) for research highlights and our [publications](#) page for a comprehensive view of our research contributions. Here are excerpts from recent articles:

MovieLens 1M Dataset

Stable benchmark dataset. 1 million ratings from 6000 users on 4000 movies. Released 2/2003.

- [README.txt](#)
- [ml-1m.zip](#) (size: 6 MB, [checksum](#))

Permalink: <http://grouplens.org/datasets/movielens/1m/>

```
1 import sys
2 from pyspark import SparkConf, SparkContext
3 from math import sqrt
4
5 #To run on EMR successfully + output results for Star Wars:
6 #aws s3 cp s3://sundog-spark/MovieSimilarities1M.py ./
7 #aws s3 cp c3://sundog-spark/ml-1m/movies.dat ./
8 #spark-submit --executor-memory 1g MovieSimilarities1M.py 260
9
10 def loadMovieNames():
11     movieNames = {}
12     with open("movies.dat") as f:
13         for line in f:
14             fields = line.split(":")
15             movieNames[int(fields[0])] = fields[1].decode('ascii', 'ignore')
16     return movieNames
17
18 def makePairs((user, ratings)):
19     (movie1, rating1) = ratings[0]
20     (movie2, rating2) = ratings[1]
21     return ((movie1, movie2), (rating1, rating2))
22
23 def filterDuplicates( (userID, ratings) ):
24     (movie1, rating1) = ratings[0]
25     (movie2, rating2) = ratings[1]
26     return movie1 < movie2
27
28 def computeCosineSimilarity(ratingPairs):
29     numPairs = 0
30     sum_xx = sum_yy = sum_xy = 0
31     for ratingX, ratingY in ratingPairs:
32         sum_xx += ratingX * ratingX
33         sum_yy += ratingY * ratingY
34         sum_xy += ratingX * ratingY
35         numPairs += 1
36
37     numerator = sum_xy
38     denominator = sqrt(sum_xx) * sqrt(sum_yy)
39
40     score = 0
41     if (denominator):
42         score = (numerator / (float(denominator)))
43
44     return (score, numPairs)
45
46
47 conf = SparkConf()
48 sc = SparkContext(conf = conf)
```

The screenshot shows the AWS homepage. At the top, there's a navigation bar with links for 'Menu', 'Amazon web services', 'Products', 'Solutions', 'Pricing', 'Software', 'More', 'English', 'My Account', and a yellow 'Sign In to the Console' button. Below the navigation is a blue banner with the text 'Try AWS with a 10-Minute Tutorial' and a subtext '“Hello, World!” technical documents to help you get hands-on with AWS'. It features three circular icons representing different services. To the right of the banner is a call-to-action box for 'Get Started with AWS for Free' with a 'Create a Free Account' button, information about 'Amazon S3' (5GB storage, 20k Get requests and 2k Put requests), and a link to 'View AWS Free Tier Details'. Below the banner are four service cards: 'AWS DATABASE MIGRATION SERVICE' (OVER 28,000 DATABASES Migrated), 'RUN MICROSERVICES ON AWS' (Learn to build and scale containerized microservices with Amazon ECS), 'AWS TECHCHAT' (Stay informed of the latest round up of AWS news and announcements. Subscribe to AWS TechChat), and 'WINDOWS ON AWS' (Download whitepapers and join live webinars).



Welcome to Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that enables businesses, researchers, data analysts, and developers to easily and cost-effectively process vast amounts of data.

You do not appear to have any clusters. Create one now:

[Create cluster](#)

Secure | https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#quick-create:

Services ▾ Resource Groups ▾

Create Cluster - Quick Options [Go to advanced options](#)

General Configuration

Cluster name [i](#)

Logging [i](#)

S3 folder [i](#)

Launch mode Cluster [i](#) Step execution [i](#)

Software configuration

Release [i](#)

Applications Core Hadoop: Hadoop 2.7.3 with Ganglia 3.7.2, Hive 2.1.1, Hue 3.12.0, Mahout 0.13.0, Pig 0.16.0, and Tez 0.8.4

HBase: HBase 1.3.0 with Ganglia 3.7.2, Hadoop 2.7.3, Hive 2.1.1, Hue 3.12.0, Phoenix 4.9.0, and ZooKeeper 3.4.10

Presto: Presto 0.170 with Hadoop 2.7.3 HDFS and Hive 2.1.1 Metastore

Spark: Spark 2.1.1 on Hadoop 2.7.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.1

Hardware configuration

Instance type [i](#)

Number of instances (1 master and 2 core nodes)

Security and access

EC2 key pair [i](#) [Learn how to create an EC2 key pair.](#)

Create Cluster - Quick Options [Go to advanced options](#)

General Configuration

Cluster name [i](#)

Logging [i](#)

Launch mode Cluster [i](#) Step execution [i](#)

Software configuration

Release emr-5.6.0  

Applications

- Core Hadoop: Hadoop 2.7.3 with Ganglia 3.7.2, Hive 2.1.1, Hue 3.12.0, Mahout 0.13.0, Pig 0.16.0, and Tez 0.8.4
- HBase: HBase 1.3.0 with Ganglia 3.7.2, Hadoop 2.7.3, Hive 2.1.1, Hue 3.12.0, Phoenix 4.9.0, and ZooKeeper 3.4.10
- Presto: Presto 0.170 with Hadoop 2.7.3 HDFS and Hive 2.1.1 Metastore
- Spark: Spark 2.1.1 on Hadoop 2.7.3 YARN with Ganglia 3.7.2 and Zeppelin 0.7.1

Hardware configuration

Instance type m3.xlarge 

Number of instances 3  (1 master and 2 core nodes)

Security and access

EC2 key pair sparkkey   Learn how to create an EC2 key pair.

Permissions Default Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role EMR_DefaultRole 

EC2 instance profile EMR_EC2_DefaultRole 

[Cancel](#)

[Create cluster](#)

Secure | https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-details;j-1H2CSNYHSENXX

Services ▾ Resource Groups ▾

Add step Resize Clone Terminate AWS CLI export

Amazon EMR

Cluster list Security configurations VPC subnets Events Help

Cluster: One Million Ratings Starting

Connections: --
Master public DNS: --
Tags: -- View All / Edit

Summary

ID: J-1H2CSNYHSENXX
Creation date: 2017-06-28 16:25 (UTC+5:30)
Elapsed time:
Auto-terminate: No
Termination Off Change protection:

Configuration Details

Release label: emr-5.6.0
Hadoop Amazon 2.7.3 distribution:
Applications: Ganglia 3.7.2, Spark 2.1.1, Zeppelin 0.7.1
Log URI: --
EMRFS consistent Disabled view:

Network and Hardware

Availability zone: --
Subnet ID: subnet-b481fb8e
Master: Provisioning 1 m3.xlarge
Core: Provisioning 2 m3.xlarge
Task: --

Security and Access

Key name: sparkkey
EC2 instance EMR_EC2_DefaultRole profile:
EMR role: EMR_DefaultRole
Visible to all users: All Change
Security groups for Master:
Security groups for Core & Task:

▶ Monitoring
▶ Hardware
▶ Steps
▶ Configurations

Network and Hardware

Availability zone: --

Subnet ID: subnet-b481fb8e

Master: Provisioning 1 m3.xlarge

Core: Provisioning 2 m3.xlarge

Task: --

Network and Hardware

Availability zone: us-east-1e
Subnet ID: subnet-b481fb8e
Master: Running 1 m3.xlarge
Core: Running 2 m3.xlarge
Task: --

← → ⓘ https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1#cluster-detailsj-1H2CSNYHSENXX

Services Resource Groups

Add step Resize Clone Terminate AWS CLI export

Amazon EMR

Cluster list Security configurations VPC subnets Events

Cluster: One Million Ratings Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Zeppelin, Spark History Server, Ganglia, Resource Manager ... (View All)
Master public DNS: ec2-34-224-17-148.compute-1.amazonaws.com SSH
Tags: -- View All / Edit

SSH

Connect to the Master Node Using SSH

You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on. [Learn more.](#)

Windows Mac / Linux

1. Open a terminal window. On Mac OS X, choose Applications > Utilities > Terminal. On other Linux distributions, terminal is typically found at Applications > Accessories > Terminal.
2. To establish a connection to the master node, type the following command. Replace ~/sparkkey.pem with the location and filename of the private key file (.pem) used to launch the cluster.
`ssh -i ~/sparkkey.pem hadoop@ec2-54-85-206-28.compute-1.amazonaws.com`
3. Type yes to dismiss the security warning.

Clo

SSH

Connect to the Master Node Using SSH

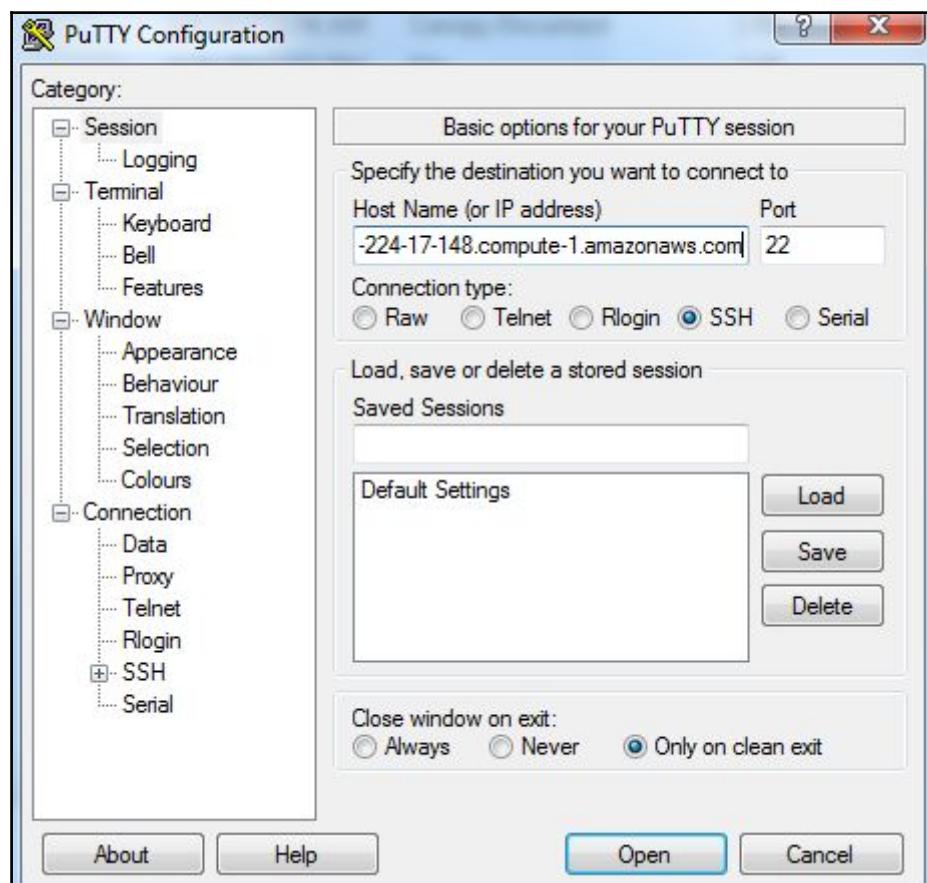
You can connect to the Amazon EMR master node using SSH to run interactive queries, examine log files, submit Linux commands, and so on.

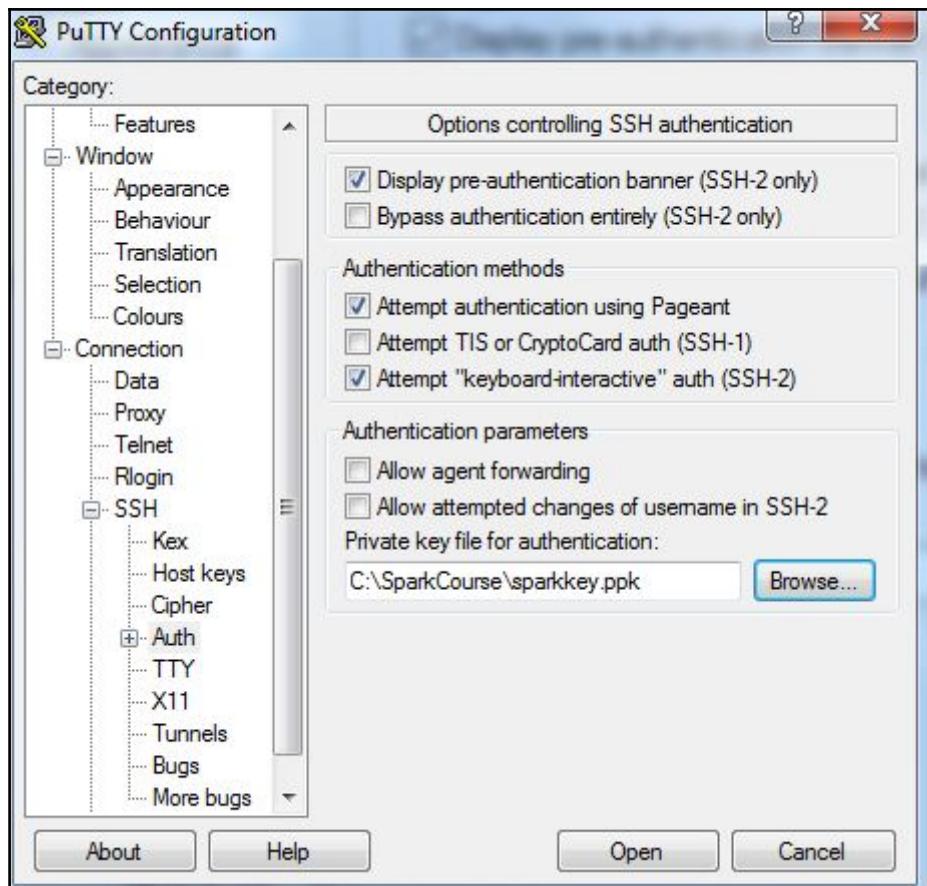
[Learn more](#)

Windows Mac / Linux

1. Download PuTTY.exe to your computer from:
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
2. Start PuTTY.
3. In the Category list, click Session.
4. In the Host Name field, type **hadoop@ec2-34-224-17-148.compute-1.amazonaws.com**.
5. In the Category list, expand Connection > SSH, and then click Auth.
6. For Private key file for authentication, click Browse and select the private key file (**sparkkey.ppk**) used to launch the cluster.
7. Click Open.
8. Click Yes to dismiss the security alert.

[Close](#)





```
[hadoop@ip-172-31-12-235 ~]$ aws s3 cp s3://sundog-spark/MovieSimilarities1M.py  
./
```

```
download: s3://sundog-spark/MovieSimilarities1M.py to ./MovieSimilarities1M.py  
[hadoop@ip-172-31-12-235 ~]$
```

```
[hadoop@ip-172-31-12-235 ~]$ aws s3 cp s3://sundog-spark/ml-1m/movies.dat ./  
download: s3://sundog-spark/ml-1m/movies.dat to ./movies.dat
```

```
[hadoop@ip-172-31-12-235 ~] $
```

```
[hadoop@ip-172-31-12-235 ~]$ spark-submit --executor-memory 1g MovieSimilarities1M.py 260
```

```

Top 10 similar movies for Star Wars: Episode IV - A New Hope (1977)
Star Wars: Episode V - The Empire Strikes Back (1980) score: 0.989791710657 strength: 2355
Sanjuro (1962) score: 0.987715715754 strength: 60
Raiders of the Lost Ark (1981) score: 0.985554827857 strength: 1972
Star Wars: Episode VI - Return of the Jedi (1983) score: 0.984124835993 strength: 2113
Run Silent, Run Deep (1958) score: 0.979146338933 strength: 145 +
Laura (1944) score: 0.978729003724 strength: 187
Close Shave, A (1995) score: 0.978216762084 strength: 436
Wrong Trousers, The (1993) score: 0.978051224484 strength: 596
Captain Horatio Hornblower (1951) score: 0.977892172004 strength: 81
Indiana Jones and the Last Crusade (1989) score: 0.977444002865 strength: 1397

```

Secure | https://console.aws.amazon.com/elasticmapreduce/home?region=us-east-1

Services | Resource Groups | [Create cluster](#) | View details | Clone | Terminate

Amazon EMR

- Cluster list
- Security configurations
- VPC subnets
- Events
- Help

Name	ID	Status	Creation time (UTC+5:30)	Elapsed time
One Million Ratings	J-1H2CSNYHSENXX	Waiting Cluster ready	2017-06-28 16:25 (UTC+5:30)	2 hours, 2 minutes

Add step | Resize | Clone | Terminate | AWS CLI export

Cluster: One Million Ratings Waiting Cluster ready after last step completed.

Connections: Enable Web Connection – Zeppelin, Spark History Server, Ganglia, Resource Manager ... (View All)

Master public DNS: ec2-34-224-17-148.compute-1.amazonaws.com SSH

Tags: -- View All / Edit

Summary

ID: J-1H2CSNYHSENXX

Creation date: 2017-06-28 16:25 (UTC+5:30)

Elapsed time: 2 hours, 3 minutes

Auto-terminate: No

Termination Off Change protection:

Configuration Details

Release label: emr-5.6.0

Hadoop Amazon 2.7.3

distribution:

Applications: Ganglia 3.7.2, Spark 2.1.1, Zeppelin 0.7.1

Log URI: --

EMRFS consistent Disabled view:

Network and Hardware

Availability zone: us-east-1e

Subnet ID: subnet-b481fb8e

Master: Running 1 m3.xlarge

Core: Running 2 m3.xlarge

Task: --

Security and Access

Key name: sparkkey

EC2 instance role EMR_E2_DefaultRole

profile:

EMR role: EMR_DefaultRole

Visible to all users: All Change

Security groups for sg-ce6ef8a9 (ElasticMapReduce-Master: master)

Security groups for sg-cce6ef8ab (ElasticMapReduce-Core & Task: slave)

Monitoring

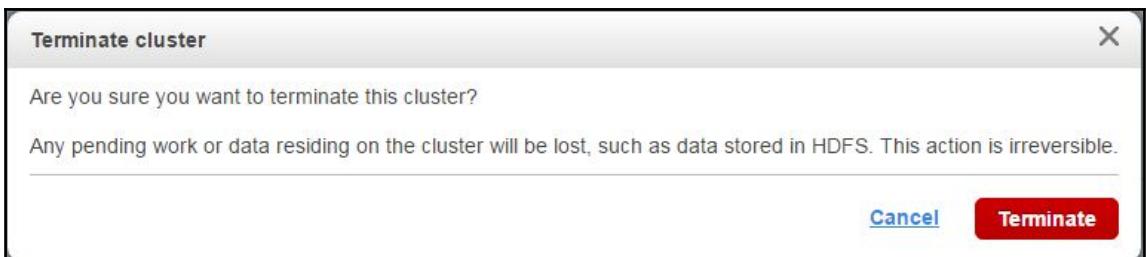
Hardware

Steps

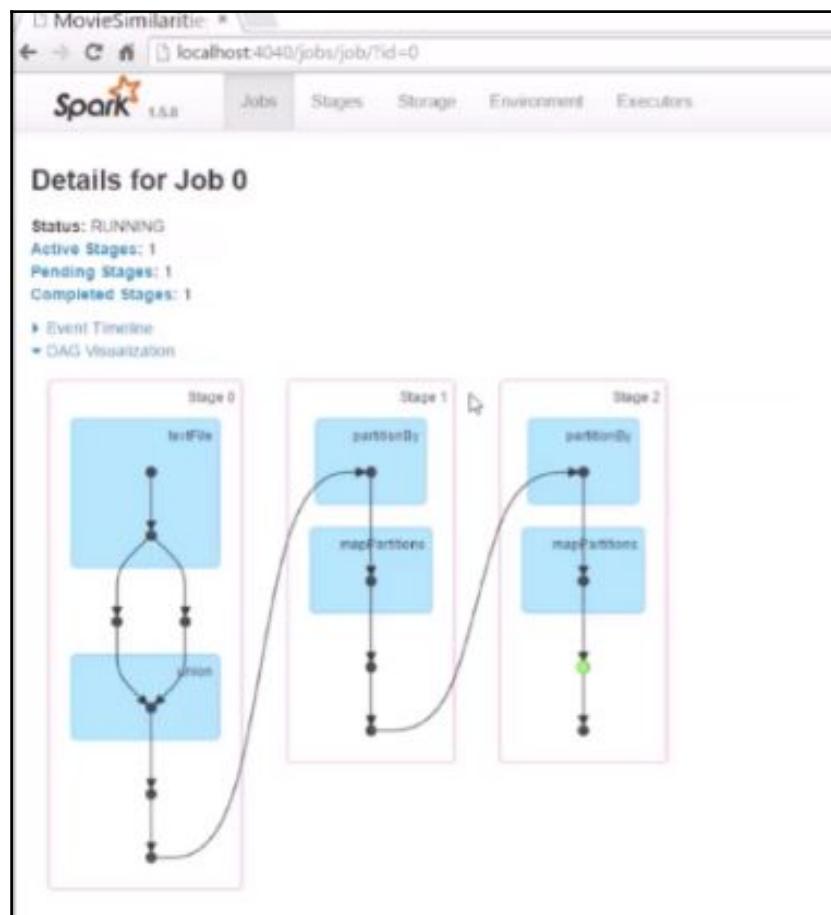
Configurations

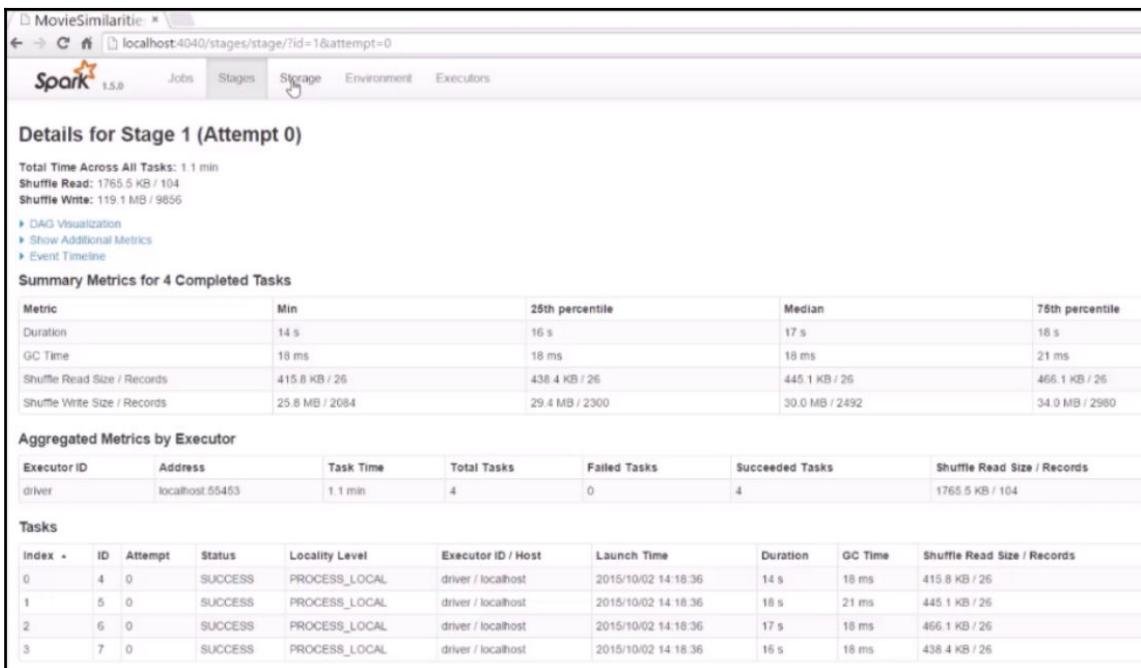
Feedback English

© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



```
(User) c:\SparkCourse>spark-submit movie-similarities.py 50
```





localhost:4040/environment/

Apache Spark 2.1.1 Jobs Stages Storage Environment Executors MovieSimilarities application UI

Environment

Runtime Information

Name	Value
Java Home	C:\jdk\jre
Java Version	1.8.0_131 (Oracle Corporation)
Scala Version	version 2.11.8

Spark Properties

Name	Value
spark.app.id	local-1498653656775
spark.app.name	MovieSimilarities
spark.driver.host	192.168.56.1
spark.driver.port	61784
spark.executor.id	driver
spark.files	file:/c:/SparkCourse/movie-similarity.py
spark.master	local[*]
spark.rdd.compress	True
spark.scheduler.mode	FIFO
spark.serializer.objectStreamReset	100
spark.submit.deployMode	client

System Properties

Name	Value
SPARK_SUBMIT	true
awt.toolkit	sun.awt.windows.WToolkit
file.encoding	Cp1252

localhost:4040/executors/

Apache Spark 2.1.1 Jobs Stages Storage Environment Executors MovieSimilarities application UI

Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Shuffle Input	Shuffle Read	Shuffle Write
Active(1)	3	37 KB / 384.1 MB	0.0 B	4	2	0	6	8	2.5 min (0.1 s)	131.1 KB	0.0 B	71.5 MB
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B
Total(1)	3	37 KB / 384.1 MB	0.0 B	4	2	0	6	8	2.5 min (0.1 s)	131.1 KB	0.0 B	71.5 MB

Executors

Show 20 entries Search:

Executor ID	Address	Status	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Shuffle Input	Shuffle Read	Shuffle Write	Thread Dump
driver	192.168.56.1:62215	Active	3	37 KB / 384.1 MB	0.0 B	4	2	0	6	8	2.5 min (0.1 s)	131.1 KB	0.0 B	71.5 MB	Thread Dump

Showing 1 to 1 of 1 entries Previous 1 Next

Thread dump for executor driver			
Updated at 2017/06/28 18:16:50			
Expand All			
Search: <input type="text"/>			
Thread ID	Thread Name	Thread State	Thread Locks
67	Executor task launch worker for task 0	BLOCKED	Blocked by Thread Some(68) Lock(org.apache.spark.SparkEnv@1984392522) Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@1478623886)
68	Executor task launch worker for task 1	RUNNABLE	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@1231894388), Monitor(java.net.SocksSocketImpl@1452466218)), Monitor(org.apache.spark.SparkEnv@1984392522)
69	Executor task launch worker for task 2	BLOCKED	Blocked by Thread Some(68) Lock(org.apache.spark.SparkEnv@1984392522) Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@602954446)
70	Executor task launch worker for task 3	RUNNABLE	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@3853852), Monitor(java.io.BufferedReaderInputStream@1475785868))
5	Attach Listener	RUNNABLE	
66	context-cleaner-periodic-gc	TIMED_WAITING	
55	dag-scheduler-event-loop	WAITING	
17	dispatcher-event-loop-0	WAITING	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@632441021))
18	dispatcher-event-loop-1	WAITING	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@1586528776))
19	dispatcher-event-loop-2	WAITING	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@423825829))
20	dispatcher-event-loop-3	WAITING	Lock(java.util.concurrent.ThreadPoolExecutor\$Worker@853094394))
56	driver-heartbeater	TIMED_WAITING	
3	Finalizer	WAITING	
52	heartbeat-receiver-event-loop-thread	TIMED_WAITING	
73	Idle Worker Monitor for python	TIMED_WAITING	
1	main	RUNNABLE	

Stages for All Jobs										
Active Stages: 1		Pending Stages: 2								
Active Stages (1)		Pending Stages (2)								
Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write		
0	join at c:/SparkCourse/movie-similarities.py:55 +details (kill)	2017/06/28 18:19:29	7 s	3/4	64.0 KB			1273.0 KB		
Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write		
2	sortByKey at c:/SparkCourse/movie-similarities.py:92 +details Unknown Unknown			0/4						
1	groupByKey at c:/SparkCourse/movie-similarities.py:67 +details Unknown Unknown			0/4						

Chapter 5

```
1 |from pyspark.sql import SparkSession
2 |from pyspark.sql import Row
3 |
4 |import collections
5 |
6 # Create a SparkSession (Note, the config section is only for Windows!)
7 spark = SparkSession.builder.config("spark.sql.warehouse.dir", "file:///C:/temp").appName("SparkSQL").getOrCreate()
8 |
9 |def mapper(line):
10 |    fields = line.split(',')
11 |    return Row(ID=int(fields[0]), name=str(fields[1].encode("utf-8")), age=int(fields[2]), numFriends=int(fields[3]))
12 |
13 lines = spark.sparkContext.textFile("fakefriends.csv")
14 people = lines.map(mapper)
15 |
16 # Infer the schema, and register the DataFrame as a table.
17 schemaPeople = spark.createDataFrame(people).cache()
18 schemaPeople.createOrReplaceTempView("people")
19 |
20 # SQL can be run over DataFrames that have been registered as a table.
21 teenagers = spark.sql("SELECT * FROM people WHERE age >= 13 AND age <= 19")
22 |
23 # The results of SQL queries are RDDs and support all the normal RDD operations.
24 for teen in teenagers.collect():
25     print(teen)
26 |
27 # We can also use functions instead of SQL queries:
28 schemaPeople.groupBy("age").count().orderBy("age").show()
29 |
30 spark.stop()
31
```

```
Row(ID=21, age=19, name=u'Miles', numFriends=268)
Row(ID=52, age=19, name=u'Beverly', numFriends=269)
Row(ID=54, age=19, name=u'Brunt', numFriends=5)
Row(ID=106, age=18, name=u'Beverly', numFriends=499)
Row(ID=115, age=18, name=u'Dukat', numFriends=397)
Row(ID=133, age=19, name=u'Quark', numFriends=265)
Row(ID=136, age=19, name=u'Will', numFriends=335)
Row(ID=225, age=19, name=u'Elim', numFriends=106)
Row(ID=304, age=19, name=u'Will', numFriends=404)
Row(ID=341, age=18, name=u'Data', numFriends=326)
Row(ID=366, age=19, name=u'Keiko', numFriends=119)
Row(ID=373, age=19, name=u'Quark', numFriends=272)
Row(ID=377, age=18, name=u'Beverly', numFriends=418)
Row(ID=404, age=18, name=u'Kasidy', numFriends=24)
Row(ID=409, age=19, name=u'Nog', numFriends=267)
Row(ID=439, age=18, name=u'Data', numFriends=417)
Row(ID=444, age=18, name=u'Keiko', numFriends=472)
Row(ID=492, age=19, name=u'Dukat', numFriends=36)
Row(ID=494, age=18, name=u'Kasidy', numFriends=194)
```

age	count
18	8
19	11
20	5
21	8
22	7
23	10
24	5
25	11
26	17
27	8
28	10
29	12
30	11
31	8
32	11
33	12
34	6
35	8
36	10
37	9

```

1 from pyspark.sql import SparkSession
2 from pyspark.sql import Row
3 from pyspark.sql import functions
4
5 def loadMovieNames():
6     movieNames = {}
7     with open("ml-100k/u.ITEM") as f:
8         for line in f:
9             fields = line.split('|')
10            movieNames[int(fields[0])] = fields[1]
11    return movieNames
12
13 # Create a SparkSession (the config bit is only for Windows!)
14 spark = SparkSession.builder.config("spark.sql.warehouse.dir", "file:///C:/temp").appName("PopularMovies").getOrCreate()
15
16 # Load up our movie ID -> name dictionary
17 nameDict = loadMovieNames()
18
19 # Get the raw data
20 lines = spark.sparkContext.textFile("file:///SparkCourse/ml-100k/u.data")
21 # Convert it to a RDD of Row objects
22 movies = lines.map(lambda x: Row(movieID =int(x.split()[0]),)

```

movieID	count
50	584
258	509
100	508
181	507
294	485
286	481
288	478
1	452
300	431
121	429
174	420
127	413
56	394
7	392
98	390
237	384
117	378
172	368
222	365
313	350

only showing top 20 rows

Star Wars (1977): 584
Contact (1997): 509
Fargo (1996): 508
Return of the Jedi (1983): 507
Liar Liar (1997): 485
English Patient, The (1996): 481
Scream (1996): 478
Toy Story (1995): 452
Air Force One (1997): 431
Independence Day (ID4) (1996): 429

Chapter 6

0	50	5	881250949
0	172	5	881250949
0	133	1	881250949

```
(User) c:\SparkCourse>spark-submit movie-recommendations-als.py 0
```

```
Ratings for user ID 0:  
Star Wars (1977): 5.0  
Empire Strikes Back, The (1980): 5.0  
Gone with the Wind (1939): 1.0  
  
Top 10 recommendations:  
Love in the Afternoon (1957) score 6.42090083536  
Roommates (1995) score 6.39431215726  
Burnt Offerings (1976) score 6.38702183096  
Lost in Space (1998) score 6.38680899253  
Endless Summer 2, The (1994) score 6.30275992511  
Primary Colors (1998) score 6.03035775839  
Drunks (1995) score 5.92894606542  
Cronos (1992) score 5.71380632161  
unknown score 5.676838214  
Double Team (1997) score 5.65588319517
```

```
(User) c:\SparkCourse>
```

```
Ratings for user ID 0:  
[Stage 280:>  
[Stage 280:=====] (0 + 2) / 2]  
[Stage 280:=====] (1 + 1) / 2]  
  
star Wars (1977): 5.0  
Empire Strikes Back, The (1980): 5.0  
Gone with the Wind (1939): 1.0  
  
Top 10 recommendations:  
Roommates (1995) score 7.87966702947  
I'll Do Anything (1994) score 7.57841013131    ↗  
Shall We Dance? (1937) score 7.23874848332  
Don't Be a Menace to South Central while Drinking Your Juice in the Hood (1996)  
score 6.72436905195  
Low Down Dirty Shame, A (1994) score 6.13396930989  
Army of Darkness (1993) score 5.98367809308  
Underneath, The (1995) score 5.9643946162  
Lord of Illusions (1995) score 5.95305643224  
Hard Eight (1996) score 5.93277528025  
In the Line of Duty 2 (1987) score 5.88337368104  
  
(Canopy 64bit) c:\SparkCourse>
```

```
Top 10 recommendations:  
War, The (1994) score 6.65716239806  
Low Down Dirty Shame, A (1994) score 6.44548993994  
Lost in Space (1998) score 6.27515939994  
Love in the Afternoon (1957) score 5.60112839882  
Schizopolis (1996) score 5.56638126463  
Meet John Doe (1941) score 5.11439598351  
Star Wars (1977) score 5.04373210278  
Addiction, The (1995) score 4.96306972202  
Empire Strikes Back, The (1980) score 4.92202227603  
Fast, Cheap & Out of Control (1997) score 4.91837364129
```

```
<User> c:\SparkCourse>spark-submit spark-linear-regression.py
```

```
(0.8061518555150741, 1.19)  
(0.956678266497083, 1.25)  
(0.8634952501748869, 1.27)  
(0.9423424178321297, 1.34)  
(1.049861282819279, 1.36)  
(0.9136707205022232, 1.44)  
(1.0785329801491854, 1.45)  
(1.0928688288141386, 1.52)  
(1.1358763748089984, 1.53)  
(1.1502122234739516, 1.54)  
(1.1502122234739516, 1.55)  
(1.1358763748089984, 1.59)  
(1.0928688288141386, 1.74)  
(1.3939216507781564, 1.78)  
(1.2648990127935775, 1.82)  
(1.3007386344559606, 1.85)  
(1.2290593911311944, 1.93)  
(1.5086084400977824, 1.95)  
(1.479936742767876, 1.98)  
(1.329410331785867, 2.0)  
(1.594623532087502, 2.08)
```

```
<User> c:\SparkCourse>
```