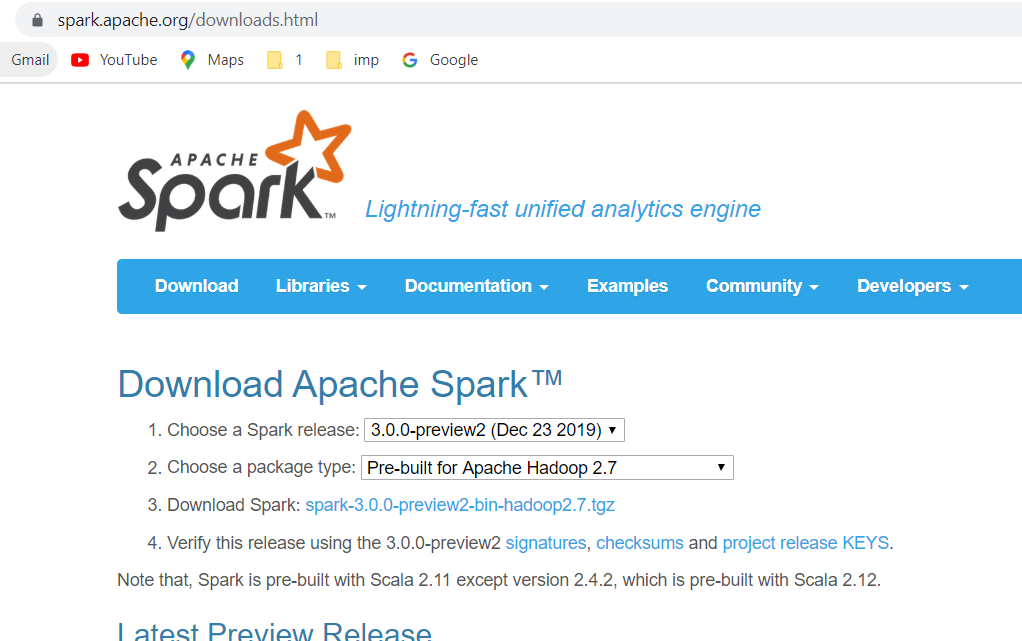
**Useful information Spark**

1. Download Spark from below url.

<https://spark.apache.org/downloads.html>



1. Run the spark-shell.cmd from bin folder to start the Spark.
2. Download winutils.exe from <http://public-repo-1.hortonworks.com/hdp-win-alpha/winutils.exe>.
3. SetUp your HADOOP\_HOME environment variable on the OS level or programmatically:

//System.setProperty("hadoop.home.dir", "full path to the folder with winutils");

HADOOP\_HOME=D:\Practice\spark-scala\{bin\winutil.exe}

Important points :

* Sc (spart Context ),spark(Spark Session)

Scala>:help

scala> :type sc

org.apache.spark.SparkContext

scala> :type spark

org.apache.spark.sql.SparkSession

scala> :history

1 jhj

2 :help

3 type sc

4 type:sc

5 :type sc

6 :type spark

7 :history

**RDD:**

1. RDD Immutable objects
2. It supports any language (scala,java,pythom)
3. Creation of RDD.
4. **Parallelize method**

* scala> val intArry = Array(41,56,85,9,85)

intArry: Array[Int] = Array(41, 56, 85, 9, 85)

scala> val intRDD = sc.**parallelize**(intArry)

intRDD: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[5] at parallelize at <console>:26

Or

scala> val intRDD = sc.parallelize(intArry,2)

intRDD: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[7] at parallelize at <console>:26

* **Reading data:**

scala> intRDD.first()

res6: Int = 41

scala> intRDD.take(3)

res7: Array[Int] = Array(41, 56, 85)

scala> intRDD.collect()

res10: Array[Int] = Array(41, 56, 85, 9, 85)

scala> intRDD.collect().foreach(println)

41

56

85

9

85

scala> intRDD.**partitions**.size

res14: Int = 8

1. **Using Text file**

**scala> val fileRDD = sc.textFile("C:\\Users\\DELL\\Downloads\\output-onlinerandomtools.txt",7);**

**fileRDD: org.apache.spark.rdd.RDD[String] = C:\Users\DELL\Downloads\output-onlinerandomtools.txt MapPartitionsRDD[3] at textFile at <console>:24**

**scala> fileRDD.partitions.size**

**res0: Int = 7**

**scala> fileRDD.take(6).foreach(println)**

**abc yuhuh uyuyu uiuiu**

**vv vgbg tyh gggg**

**t ff fff ffdfdfdzXzXzXzXzXZ**

**t fqwqf fff ffdfdfd**

**t ffqwqw ffaf ffdfdfd**

**t ffaw fffa ffdfdfdzxzXZXZXzxzXzX**

**scala> fileRDD.first()**

**res2: String = abc yuhuh uyuyu uiuiu**

**scala>**

**Transformations:**

* **filter:**

scala> val data = Array("Hi how are you","HI","Hi hsg shjsh sjhsdj","Ji hj","sakdh askjdhaskjd askjdsah");

data: Array[String] = Array(Hi how are you, HI, Hi hsg shjsh sjhsdj, Ji hj, sakdh askjdhaskjd askjdsah)

scala> val dataRdd = sc.parallelize(data);

dataRdd: org.apache.spark.rdd.RDD[String] = ParallelCollectionRDD[4] at parallelize at <console>:26

scala> val frdd = dataRdd.filter(l=>l.length>10);

frdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[5] at filter at <console>:25

scala> frdd.collect.foreach(println);

Hi how are you

Hi hsg shjsh sjhsdj

sakdh askjdhaskjd askjdsah

* **map:**

scala> val mapRdd = dataRdd.map(l=>l.split(" "));

mapRdd: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[6] at map at <console>:25

scala> mapRdd.collect();

res6: Array[Array[String]] = Array(Array(Hi, how, are, you), Array(HI), Array(Hi, hsg, shjsh, sjhsdj), Array(Ji, hj), Array(sakdh, askjdhaskjd, askjdsah))

* **flatMap:**

scala> val fmapRdd = dataRdd.flatMap(l=>l.split(" "));

fmapRdd: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[8] at flatMap at <console>:25

scala> fmapRdd.collect();

res7: Array[String] = Array(Hi, how, are, you, HI, Hi, hsg, shjsh, sjhsdj, Ji, hj, sakdh, askjdhaskjd, askjdsah)

* **distinct:**

scala> val intData = Array(1,1,5,8,9,4,4,9,1,55,77,9);

intData: Array[Int] = Array(1, 1, 5, 8, 9, 4, 4, 9, 1, 55, 77, 9)

scala> val intRDD = sc.parallelize(intData);

intRDD: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[13] at parallelize at <console>:26

scala> val diRdd = intRDD.distinct();

diRdd: org.apache.spark.rdd.RDD[Int] = MapPartitionsRDD[16] at distinct at <console>:25

scala> diRdd.collect();

res11: Array[Int] = Array(8, 1, 9, 4, 77, 5, 55)

Cloud era download:

<https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.13.0-0-virtualbox.zip>

Mount fine in Cloud era:

>mkdir Cloudera

[root@quickstart workspace]# mount -t vboxsf Cloudera /home/cloudera/workspace/Cloudera/

User Creation:

>sudo useradd DELL

>hdfs dfs -ls /user

>sudo -u hdfs hdfs dfs -mkdir /user/DELL

>sudo -u DELL hdfs dfs -mkdir /user/DELL/jars

* yarn application -kill application\_id

spark-submit --master yarn --class Test test1\_2.11-0.1.jar