**Big Data Trunk Spark Workshop**

****

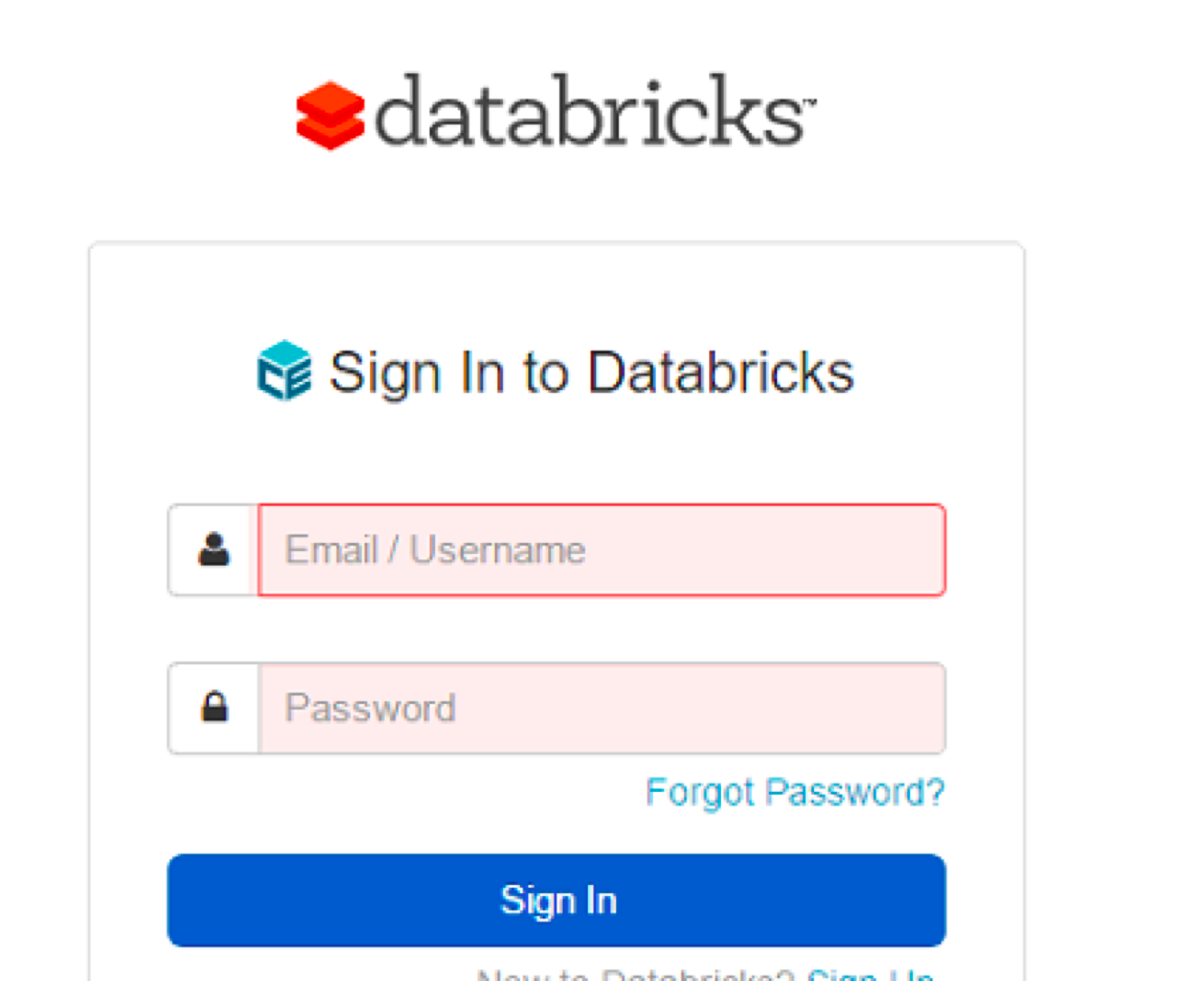
**Lab 1**

**Create account on Databricks cloud using below link**

**Sign-up for Databricks Community Edition**

**<https://accounts.cloud.databricks.com/registration.html#signup/community>**

**<https://community.cloud.databricks.com/login.html>**

****

**Lab 2 - WordCount**

WordCount program in spark

====================================================================================

**Instructions**:This document explains a simple wordcount program in Spark

**Databricks Notebook**

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2987202527269392/1559169996017265/3091033648572544/latest.html>

**Step by step solution**

<https://drive.google.com/open?id=0BwmmpwU6fMMkaWVJLUVMRkt0Nzg>

// Databricks notebook source

val x = sc.parallelize(List("spark", "rdd", "example", "sample", "example"), 3)

val wordcount = x.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(\_ + \_)

wordcount.collect

**Lab 3 - Spark RDD Part 1**

Spark RDD Part 1

====================================================================================

**Instructions**:This document explains Spark RDD Basic Action and Transformations

**Databricks Notebook**

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2987202527269392/3306018232269556/3091033648572544/latest.html>

**Step by step solution**

<https://drive.google.com/open?id=0BwmmpwU6fMMkdk1zRW9wdWdLNEU>

val x = sc.parallelize(List("spark", "rdd", "example", "sample", "example"), 3)

val y = x.map(x => (x, 1))

y.collect

val y = x.map((\_, 1))

y.collect

val y = x.map(x => (x, x.length))

y.collect

val x = sc.parallelize(List("spark rdd example", "sample example"), 2)

val y = x.groupBy(\_.charAt (0))

y.collect

val y = x.map(x => x.split(" ")) // split(" ")

y.collect

val y = x.flatMap(\_.split(" "))

y.collect

val x = sc.parallelize(Array(("a", 1), ("b", 1), ("a", 1),

("a", 1), ("b", 1), ("b", 1),

("b", 1), ("b", 1)), 3);

val y = x.reduceByKey((accum, n) => (accum + n))

y.collect

val y = x.reduceByKey(\_ + \_)

y.collect

val x = sc.parallelize(1 to 10, 2)

val y = x.filter(e => e%2==0)

y.collect

val y = x.filter(\_ % 2 == 0)

y.collect

val x = sc.parallelize(Array("Joseph", "Jimmy", "Tina",

"Thomas", "James", "Cory",

"Christine", "Jackeline", "Juan"), 3);

val y = x.groupBy(word => word.charAt(0));

y.collect

val y = x.groupBy(\_.charAt(0))

y.collect

**Lab 4 - Spark RDD Part 2**

Spark RDD Part 2

====================================================================================

Instructions:This document explains Spark RDD Action and Transformations(Filter,Map,RDD)

**Databricks Notebook**

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2987202527269392/3306018232269576/3091033648572544/latest.html>

**Step by step solution**

<https://drive.google.com/open?id=0BwmmpwU6fMMkWG4zei0tNFJJc2c>

var num = Array(5,10,15,20,25)

val y = num.map(num=> num\*num)

var num = Array(5, 10, 15, 20, 25)

val y = num.filter(e => e!=15)

val x = sc.parallelize(List("spark rdd example", "sample example"), 2)

val y = x.map(x => x.split(" "))

y.collect

val y = x.flatMap(\_.split(" "))

y.collect

val x = sc.parallelize(Array("John", "Tom", "Tina", "Rahil", "James", "Mary", "Arnold", "jany", "juhi"),

3)

val y = x.groupBy(word => word.charAt(0))

y.collect

**Lab 5 - Spark SQL, Data Frames**

**Databricks Notebook**

<https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/2987202527269392/1559169996017271/3091033648572544/latest.html>

**import** org.apache.spark.sql.functions.**\_**  
**val** numDS = spark.range(0, 100000, 5)  
display(numDS.orderBy(desc("id")))

**val** langPercentDF = spark.createDataFrame(List(("Scala", 35), ("SQL", 15), ("Python", 25), ("R", 5), ("Java", 20)))  
 .toDF("language", "percent")  
display(langPercentDF.orderBy(desc("percent")))

langPercentDF.write.saveAsTable("languages")

//range always returns at Dataset  
**val** employeeDS = spark.range(0, 500000).  
 select($"id".as("employee\_id"),   
 (rand() \* 3).cast("int").as("dep\_id"),   
 (rand() \* 40 + 20).cast("int").as("age"))  
 .cache()  
//create a temporary table in memory  
employeeDS.createOrReplaceTempView("employees")  
employeeDS.count()

display(spark.catalog.listTables)

**%sql** **select** employee\_id, age **from** employees **where** age > 50

**Lab 6 - Data Frames**

<https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-dataframes-python.html>

**Lab 7 - Data Sets**

https://docs.databricks.com/spark/latest/dataframes-datasets/introduction-to-datasets.html

**Bonus - challenges**

<http://dbricks.co/sswksh1>

Twitter 2016 Election

<https://www.youtube.com/watch?v=35Y-rqSMCCA>

<https://docs.cloud.databricks.com/docs/latest/featured_notebooks/2016%20Election%20Tweets.html>

**Other resources**

Introduction to Databricks

<https://community.cloud.databricks.com/?o=2987202527269392#notebook/154972344372474>

Databricks documentation

<https://docs.databricks.com/spark/latest/training/index.html>

Upcoming Databricks book ( Free)

<https://cdn2.hubspot.net/hubfs/438089/notebooks/eBook/Apache-Spark-The-Definitive-Guide-Excerpts-R1.pdf?t=1496778488662&utm_campaign=Definitive%20Guide%20Spark%20book&utm_source=hs_automation&utm_medium=email&utm_content=52725068&_hsenc=p2ANqtz-9aXqz5DrRWXH8dnNZZL6ga-BfnBfqWsYtGjyxZMkOZV4ttc7MkPTHYFy0fTUmVwBAPHsoebAiCT5_4nNZ7FbJt-TAiOw&_hsmi=52725068>

**Full training programs**

<http://www.bigdatatrunk.com/course/big-data-trunk-mentorship-program/>

