

Rishi Yadav

# Apache Spark 2.x Cookbook

Cloud-ready recipes to do analytics and data science on  
Apache Spark



**Packt**>

# Apache Spark 2.x Cookbook

Cloud-ready recipes to do analytics and data science on  
Apache Spark

**Rishi Yadav**



**BIRMINGHAM - MUMBAI**

# Apache Spark 2.x Cookbook

Copyright © 2017 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: May 2017

Production reference: 1300517

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK.

ISBN 978-1-78712-726-5

[www.packtpub.com](http://www.packtpub.com)

# Credits

**Author**

Rishi Yadav

**Copy Editor**

Gladson Monteiro

**Reviewer**

Prashant Verma

**Project Coordinator**

Nidhi Joshi

**Commissioning Editor**

Amey Varangaonkar

**Proofreader**

Safis Editing

**Acquisition Editor**

Vinay Argekar

**Indexer**

Pratik Shirodkar

**Content Development Editor**

Jagruti Babaria

**Graphics**

Tania Dutta

**Technical Editor**

Dinesh Pawar

**Production Coordinator**

Sharaddha Falebhai



# About the Author

**Rishi Yadav** has 19 years of experience in designing and developing enterprise applications. He is an open source software expert and advises American companies on big data and public cloud trends. Rishi was honored as one of Silicon Valley's 40 under 40 in 2014. He earned his bachelor's degree from the prestigious Indian Institute of Technology, Delhi, in 1998.

About 12 years ago, Rishi started InfoObjects, a company that helps data-driven businesses gain new insights into data. InfoObjects combines the power of open source and big data to solve business challenges for its clients and has a special focus on Apache Spark. The company has been on the Inc. 5000 list of the fastest growing companies for 6 years in a row. InfoObjects has also been named the best place to work in the Bay Area in 2014 and 2015.

Rishi is an open source contributor and active blogger.

*This book is dedicated to my parents, Ganesh and Bhagwati Yadav; I would not be where I am without their unconditional support, trust, and providing me the freedom to choose a path of my own.*

*Special thanks go to my life partner, Anjali, for providing immense support and putting up with my long, arduous hours (yet again).*

*Our 9-year-old son, Vedant, and niece, Kashmira, were the unrelenting force behind keeping me and the book on track.*

*Big thanks to InfoObjects' CTO and my business partner, Sudhir Jangir, for providing valuable feedback and also contributing with recipes on enterprise security, a topic he is passionate about; to our SVP, Bart Hickenlooper, for taking the charge in leading the company to the next level; to Tanmoy Chowdhury and Neeraj Gupta for their valuable advice; to Yogesh Chandani, Animesh Chauhan, and Katie Nelson for running operations skillfully so that I could focus on this book; and to our internal review team (especially Rakesh Chandran) for ironing out the kinks. I would also like to thank Marcel Izumi for, as always, providing creative visuals. I cannot miss thanking our dog, Sparky, for giving me company on my long nights out. Last but not least, special thanks to our valuable clients, partners, and employees, who have made InfoObjects the best place to work at and, needless to say, an immensely successful organization.*

# About the Reviewer

**Prashant Verma** started his IT career in 2011 as a Java developer at Ericsson, working in the telecom domain. After a couple of years of Java EE experience, he moved into the big data domain and has worked on almost all the popular big data technologies, such as Hadoop, Spark, Flume, Mongo, and Cassandra. He has also played with Scala. Currently, he works with QA Infotech as a lead data engineer, working on solving e-learning problems using analytics and machine learning.

Prashant has also been working as a freelance consultant in his spare time.

*I want to thank Packt Publishing for giving me the chance to review the book as well as my employer and my family for their patience while I was busy working on this book.*

# www.PacktPub.com

For support files and downloads related to your book, please visit [www.PacktPub.com](http://www.PacktPub.com).

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at [www.PacktPub.com](http://www.PacktPub.com) and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at [service@packtpub.com](mailto:service@packtpub.com) for more details.

At [www.PacktPub.com](http://www.PacktPub.com), you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www.packtpub.com/mapt>

Get the most in-demand software skills with Mapt. Mapt gives you full access to all Packt books and video courses, as well as industry-leading tools to help you plan your personal development and advance your career.

## Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

# Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at <https://www.amazon.com/dp/1787127265>.

If you'd like to join our team of regular reviewers, you can e-mail us at [customerreviews@packtpub.com](mailto:customerreviews@packtpub.com). We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

# Table of Contents

<b>Preface</b>	1
<b>Chapter 1: Getting Started with Apache Spark</b>	6
<b>Introduction</b>	6
<b>Leveraging Databricks Cloud</b>	8
How to do it...	9
How it works...	14
Cluster	15
Notebook	15
Table	15
Library	15
<b>Deploying Spark using Amazon EMR</b>	15
What it represents is much bigger than what it looks	15
EMR's architecture	16
How to do it...	16
How it works...	23
EC2 instance types	24
T2 - Free Tier Burstable (EBS only)	25
M4 - General purpose (EBS only)	25
C4 - Compute optimized	26
X1 - Memory optimized	26
R4 - Memory optimized	26
P2 - General purpose GPU	27
I3 - Storage optimized	27
D2 - Storage optimized	27
<b>Installing Spark from binaries</b>	27
Getting ready	28
How to do it...	28
<b>Building the Spark source code with Maven</b>	30
Getting ready	30
How to do it...	31
<b>Launching Spark on Amazon EC2</b>	33
Getting ready	33
How to do it...	34
See also	38
<b>Deploying Spark on a cluster in standalone mode</b>	38
Getting ready	39

How to do it...	39
How it works...	41
See also	43
<b>Deploying Spark on a cluster with Mesos</b>	43
How to do it...	43
<b>Deploying Spark on a cluster with YARN</b>	45
Getting ready	45
How to do it...	45
How it works...	47
<b>Understanding SparkContext and SparkSession</b>	49
SparkContext	49
SparkSession	49
<b>Understanding resilient distributed dataset - RDD</b>	49
How to do it...	50
<b>Chapter 2: Developing Applications with Spark</b>	54
<b>Introduction</b>	54
<b>Exploring the Spark shell</b>	55
How to do it...	56
There's more...	57
<b>Developing a Spark applications in Eclipse with Maven</b>	58
Getting ready	59
How to do it...	59
<b>Developing a Spark applications in Eclipse with SBT</b>	62
How to do it...	62
<b>Developing a Spark application in IntelliJ IDEA with Maven</b>	64
How to do it...	64
<b>Developing a Spark application in IntelliJ IDEA with SBT</b>	66
How to do it...	66
<b>Developing applications using the Zeppelin notebook</b>	66
How to do it...	66
<b>Setting up Kerberos to do authentication</b>	69
How to do it...	70
There's more...	71
<b>Enabling Kerberos authentication for Spark</b>	72
How to do it...	72
There's more...	74
Securing data at rest	74
Securing data in transit	75
<b>Chapter 3: Spark SQL</b>	76

<b>Understanding the evolution of schema awareness</b>	77
Getting ready	77
DataFrames	78
Datasets	79
Schema-aware file formats	79
<b>Understanding the Catalyst optimizer</b>	80
Analysis	81
Logical plan optimization	82
Physical planning	83
Code generation	83
<b>Inferring schema using case classes</b>	84
How to do it...	84
There's more...	85
<b>Programmatically specifying the schema</b>	86
How to do it...	86
How it works...	87
<b>Understanding the Parquet format</b>	88
How to do it...	89
How it works...	90
Partitioning	92
Predicate pushdown	92
Parquet Hive interoperability	92
<b>Loading and saving data using the JSON format</b>	93
How to do it...	94
How it works...	95
<b>Loading and saving data from relational databases</b>	95
Getting ready	95
How to do it...	95
<b>Loading and saving data from an arbitrary source</b>	98
How to do it...	98
There's more...	99
<b>Understanding joins</b>	99
Getting ready	99
How to do it...	99
How it works...	103
Shuffle hash join	103
Broadcast hash join	104
The cartesian join	104
There's more...	104
<b>Analyzing nested structures</b>	104
Getting ready	105

How to do it...	105
<b>Chapter 4: Working with External Data Sources</b>	<b>108</b>
<b>Introduction</b>	109
<b>Loading data from the local filesystem</b>	109
How to do it...	110
<b>Loading data from HDFS</b>	111
How to do it...	112
<b>Loading data from Amazon S3</b>	114
How to do it...	115
<b>Loading data from Apache Cassandra</b>	117
How to do it...	117
How it works	119
CAP Theorem	120
Cassandra partitions	121
Consistency levels	122
<b>Chapter 5: Spark Streaming</b>	<b>124</b>
<b>Introduction</b>	124
Classic Spark Streaming	125
Structured Streaming	126
<b>WordCount using Structured Streaming</b>	127
How to do it...	127
<b>Taking a closer look at Structured Streaming</b>	128
How to do it...	129
There's more...	130
<b>Streaming Twitter data</b>	130
How to do it...	130
<b>Streaming using Kafka</b>	135
Getting ready	137
How to do it...	137
<b>Understanding streaming challenges</b>	138
Late arriving/out-of-order data	139
Maintaining the state in between batches	139
Message delivery reliability	139
Streaming is not an island	140
<b>Chapter 6: Getting Started with Machine Learning</b>	<b>142</b>
<b>Introduction</b>	142
<b>Creating vectors</b>	143
Getting ready	144



How to do it...	144
How it works...	145
<b>Calculating correlation</b>	146
Getting ready	146
How to do it...	147
<b>Understanding feature engineering</b>	147
Feature selection	147
Quality of features	148
Number of features	148
Feature scaling	148
Feature extraction	149
TF-IDF	149
Term frequency	149
Inverse document frequency	150
How to do it...	150
<b>Understanding Spark ML</b>	151
Getting ready	152
How to do it...	153
<b>Understanding hyperparameter tuning</b>	156
How to do it...	156
<b>Chapter 7: Supervised Learning with MLlib — Regression</b>	158
<b>Introduction</b>	158
<b>Using linear regression</b>	162
Getting ready	162
How to do it...	163
There's more...	163
<b>Understanding the cost function</b>	165
There's more...	171
<b>Doing linear regression with lasso</b>	171
Bias versus variance	171
How to do it...	172
<b>Doing ridge regression</b>	173
<b>Chapter 8: Supervised Learning with MLlib — Classification</b>	175
<b>Introduction</b>	175
<b>Doing classification using logistic regression</b>	175
Getting ready	180
How to do it...	181
There's more...	182
What is ROC?	183

<b>Doing binary classification using SVM</b>	183
Getting ready	186
How to do it...	186
<b>Doing classification using decision trees</b>	187
Getting ready	189
How to do it...	191
How it works...	192
There's more...	195
<b>Doing classification using random forest</b>	196
Getting ready	200
How to do it...	200
<b>Doing classification using gradient boosted trees</b>	202
Getting ready	202
How to do it...	202
<b>Doing classification with Naïve Bayes</b>	203
Getting ready	204
How to do it...	205
<b>Chapter 9: Unsupervised Learning</b>	206
<b>Introduction</b>	206
<b>Clustering using k-means</b>	207
Getting ready	209
How to do it...	212
<b>Dimensionality reduction with principal component analysis</b>	215
Getting ready	217
How to do it...	220
<b>Dimensionality reduction with singular value decomposition</b>	222
Getting ready	224
How to do it...	225
<b>Chapter 10: Recommendations Using Collaborative Filtering</b>	227
<b>Introduction</b>	227
<b>Collaborative filtering using explicit feedback</b>	229
Getting ready	230
How to do it...	230
Adding my recommendations and then testing predictions	232
There's more...	233
<b>Collaborative filtering using implicit feedback</b>	234
How to do it...	234
<b>Chapter 11: Graph Processing Using GraphX and GraphFrames</b>	238

<b>Introduction</b>	238
<b>Fundamental operations on graphs</b>	239
Getting ready	239
How to do it...	240
<b>Using PageRank</b>	241
Getting ready	241
How to do it...	241
<b>Finding connected components</b>	243
Getting ready	244
How to do it...	245
<b>Performing neighborhood aggregation</b>	246
Getting ready	247
How to do it...	247
<b>Understanding GraphFrames</b>	249
How to do it...	249
<b>Chapter 12: Optimizations and Performance Tuning</b>	252
<b>Optimizing memory</b>	252
How to do it...	253
How it works...	254
Garbage collection	254
Mark and sweep	255
G1	256
Spark memory allocation	257
<b>Leveraging speculation</b>	257
How to do it...	257
<b>Optimizing joins</b>	258
How to do it...	258
<b>Using compression to improve performance</b>	259
How to do it...	260
<b>Using serialization to improve performance</b>	260
How to do it...	260
There's more...	261
<b>Optimizing the level of parallelism</b>	261
How to do it...	261
<b>Understanding project Tungsten</b>	262
How to do it...	263
How it works...	263
Tungsten phase 1	263
Bypassing GC	264
Cache conscious computation	264

Code generation for expression evaluation	265
Tungsten phase 2	265
Wholesale code generation	265
In-memory columnar format	265

<b>Index</b>	<b>266</b>
--------------	------------

---

# Preface

The success of Hadoop as a big data platform raised user expectations, both in terms of solving different analytics challenges and reducing latency. Various tools evolved over time, but when Apache Spark came, it provided a single runtime to address all these challenges. It eliminated the need to combine multiple tools with their own challenges and learning curves. Using memory for persistent storage besides compute, Apache Spark eliminates the need to store intermediate data on disk and increases processing speed up to 100 times. It also provides a single runtime, which addresses various analytics needs, such as machine-learning and real-time streaming, using various libraries. This book covers the installation and configuration of Apache Spark and building solutions using Spark Core, Spark SQL, Spark Streaming, MLlib, and GraphX libraries.



For more information on this book's recipes, please visit  
[infoobjects.com/spark-cookbook](http://infoobjects.com/spark-cookbook).

## What this book covers

Chapter 1, *Getting Started with Apache Spark*, explains how to install Spark on various environments and cluster managers.

Chapter 2, *Developing Applications with Spark*, talks about developing Spark applications on different IDEs and using different build tools.

Chapter 3, *Spark SQL*, covers how to read and write to various data sources.

Chapter 4, *Working with External Data Sources*, takes you through the Spark SQL module that helps you access the Spark functionality using the SQL interface.

Chapter 5, *Spark Streaming*, explores the Spark Streaming library to analyze data from real-time data sources, such as Kafka.

Chapter 6, *Getting Started with Machine Learning*, covers an introduction to machine learning and basic artifacts, such as vectors and matrices.

Chapter 7, *Supervised Learning with MLlib – Regression*, walks through supervised learning when the outcome variable is continuous.

Chapter 8, *Supervised Learning with MLlib – Classification*, discusses supervised learning when the outcome variable is discrete.

Chapter 9, *Unsupervised Learning*, covers unsupervised learning algorithms, such as k-means.

Chapter 10, *Recommendations Using Collaborative Filtering*, introduces building recommender systems using various techniques, such as ALS.

Chapter 11, *Graph Processing Using GraphX and GraphFrames*, talks about various graph processing algorithms using GraphX.

Chapter 12, *Optimizations and Performance Tuning*, covers various optimizations on Apache Spark and performance tuning techniques.

## What you need for this book

There are two ways to work with the recipes in this book:

- The first is to use Databricks Community Cloud at <https://community.cloud.databricks.com>. It is a free notebook provided by Databricks. All the sample data for this book has also been uploaded in the Amazon Web Service S3 bucket, namely `sparkcookbook`.
- The second option is to use InfoObjects Big Data Sandbox, which is a virtual machine built on top of Ubuntu. This software can be downloaded from <http://www.infoobjects.com>.

## Who this book is for

If you are a data engineer, an application developer, or a data scientist who would like to leverage the power of Apache Spark to get better insights from big data, then this is the book for you.

## Sections

In this book, you will find several headings that appear frequently (Getting ready, How to do it..., How it works..., There's more..., and See also).

To give clear instructions on how to complete a recipe, we use these sections as follows:

## Getting ready

This section tells you what to expect in the recipe, and describes how to set up any software or any preliminary settings required for the recipe.

## How to do it...

This section contains the steps required to follow the recipe.

## How it works...

This section usually consists of a detailed explanation of what happened in the previous section.

## There's more...

This section consists of additional information about the recipe in order to make the reader more knowledgeable about the recipe.

## See also

This section provides helpful links to other useful information the recipe.

## Conventions

In this book, you will find a number of text styles that distinguish between different kinds of information. Here are some examples of these styles and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Spark expects Java to be installed and the `JAVA_HOME` environment variable to be set."

A block of code is set as follows:

```
[{ "firstName" : "Bill", "lastName": "Clinton", "age": 70 }  
  {"firstName": "Barack", "lastName": "Obama", "age": 55}]
```

Any command-line input or output is written as follows:

```
scala> val people = spark.sql("select * from person")
```

**New terms** and **important** words are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Click on **Create cluster** and select the last option in the **Applications** option box."



Warnings or important notes appear in a box like this.



Tips and tricks appear like this.

## Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or disliked. Reader feedback is important for us as it helps us develop titles that you will really get the most out of.

To send us general feedback, simply e-mail [feedback@packtpub.com](mailto:feedback@packtpub.com), and mention the book's title in the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide at [www.packtpub.com/authors](http://www.packtpub.com/authors).

## Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the color images of this book

We also provide you with a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output.

You can download this file from:

[https://www.packtpub.com/sites/default/files/downloads/Spark2xCookbook\\_ColorImages.pdf](https://www.packtpub.com/sites/default/files/downloads/Spark2xCookbook_ColorImages.pdf).



## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you could report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the Errata Submission Form link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded to our website or added to any list of existing errata under the Errata section of that title.

To view the previously submitted errata, go to <https://www.packtpub.com/books/content/support> and enter the name of the book in the search field. The required information will appear under the *Errata* section.

## Piracy

Piracy of copyrighted material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works in any form on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at [copyright@packtpub.com](mailto:copyright@packtpub.com) with a link to the suspected pirated material.

We appreciate your help in protecting our authors and our ability to bring you valuable content.

## Questions

If you have a problem with any aspect of this book, you can contact us at [questions@packtpub.com](mailto:questions@packtpub.com), and we will do our best to address the problem.

# 1

# Getting Started with Apache Spark

In this chapter, we will set up Spark and configure it. This chapter contains the following recipes:

- Leveraging Databricks Cloud
- Deploying Spark using Amazon EMR
- Installing Spark from binaries
- Building the Spark source code with Maven
- Launching Spark on Amazon EC2
- Deploying Spark on a cluster in standalone mode
- Deploying Spark on a cluster with Mesos
- Deploying Spark on a cluster with YARN
- Understanding SparkContext and SparkSession
- Understanding Resilient Distributed Datasets (RDD)

## Introduction

**Apache Spark** is a general-purpose cluster computing system to process big data workloads. What sets Spark apart from its predecessors, such as **Hadoop MapReduce**, is its speed, ease of use, and sophisticated analytics.

It was originally developed at *AMPLab, UC Berkeley*, in 2009. It was made open source in 2010 under the BSD license and switched to the Apache 2.0 license in 2013. Toward the later part of 2013, the creators of Spark founded Databricks to focus on Spark's development and future releases.

Databricks offers Spark as a service in the **Amazon Web Services(AWS)** Cloud, called Databricks Cloud. In this book, we are going to maximize the use of AWS as a data storage layer.

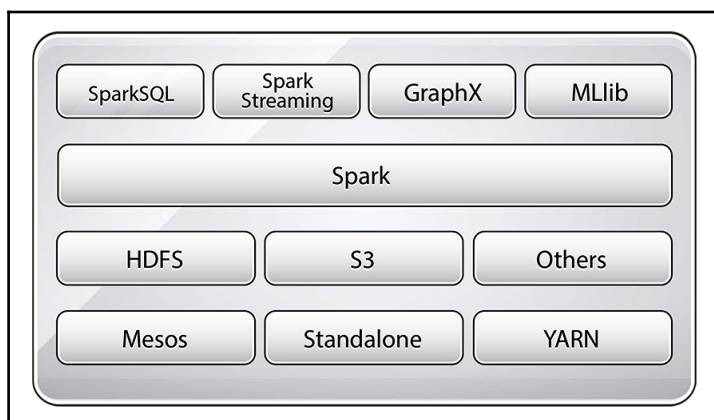
Talking about speed, Spark can achieve subsecond latency on big data workloads. To achieve such low latency, Spark makes use of memory for storage. In MapReduce, memory is primarily used for the actual computation. Spark uses memory both to compute and store objects.

Spark also provides a unified runtime connecting to various big data storage sources, such as HDFS, Cassandra, and S3. It also provides a rich set of high-level libraries for different big data compute tasks, such as machine learning, SQL processing, graph processing, and real-time streaming. These libraries make development faster and can be combined in an arbitrary fashion.

Though Spark is written in Scala--and this book only focuses on recipes on Scala--it also supports Java, Python, and R.

Spark is an open source community project, and everyone uses the pure open source Apache distributions for deployments, unlike Hadoop, which has multiple distributions available with vendor enhancements.

The following figure shows the Spark ecosystem:



Spark's runtime runs on top of a variety of cluster managers, including **YARN** (Hadoop's compute framework), **Mesos**, and Spark's own cluster manager called **Standalone** mode. Alluxio is a memory-centric distributed file system that enables reliable file sharing at memory speed across cluster frameworks. In short, it is an off-heap storage layer in memory that helps share data across jobs and users. Mesos is a cluster manager, which is evolving into a data center operating system. YARN is Hadoop's compute framework and has a robust resource management feature that Spark can seamlessly use.

Apache Spark, initially devised as a replacement of MapReduce, had a good proportion of workloads running in an on-premises manner. Now, most of the workloads have been moved to public clouds (AWS, Azure, and GCP). In a public cloud, we see two types of applications:

- Outcome-driven applications
- Data transformation pipelines

For outcome-driven applications, where the goal is to derive a predefined signal/outcome from the given data, Databricks Cloud fits the bill perfectly. For traditional data transformation pipelines, Amazon's **Elastic MapReduce (EMR)** does a great job.

## Leveraging Databricks Cloud

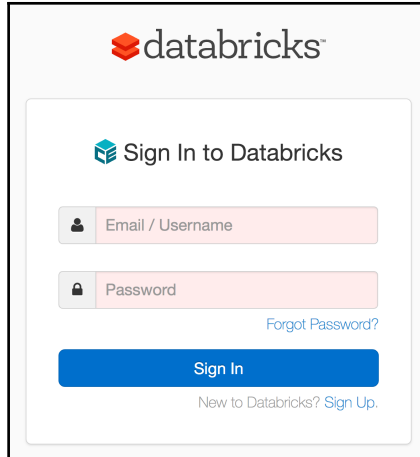
Databricks is the company behind Spark. It has a cloud platform that takes out all of the complexity of deploying Spark and provides you with a ready-to-go environment with notebooks for various languages. Databricks Cloud also has a community edition that provides one node instance with 6 GB of RAM for free. It is a great starting place for developers. The Spark cluster that is created also terminates after 2 hours of sitting idle.



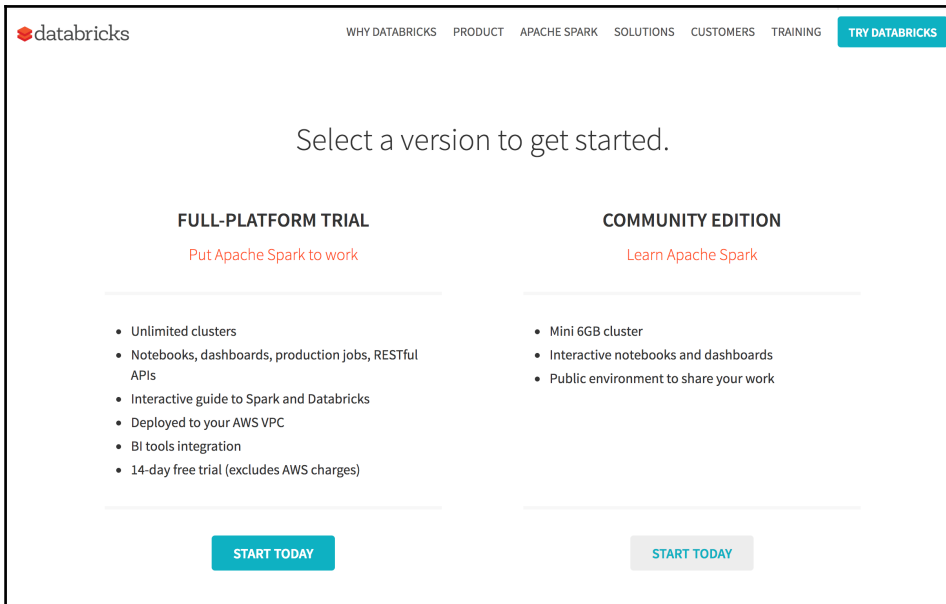
All the recipes in this book can be run on either the `InfoObjects` Sandbox or Databricks Cloud community edition. The entire data for the recipes in this book has also been ported to a public bucket called `sparkcookbook` on S3. Just put these recipes on the Databricks Cloud community edition, and they will work seamlessly.

## How to do it...


1. Go to `https://community.cloud.databricks.com:`

A screenshot of the Databricks sign-in page. At the top is the Databricks logo. Below it is a box titled "Sign In to Databricks". Inside this box are two input fields: "Email / Username" and "Password". Below the password field is a link "Forgot Password?". At the bottom of the box is a blue "Sign In" button and a link "New to Databricks? Sign Up."

2. Click on **Sign Up** :

A screenshot of the Databricks sign-up page. At the top is the Databricks logo and a navigation bar with links: "WHY DATABRICKS", "PRODUCT", "APACHE SPARK", "SOLUTIONS", "CUSTOMERS", "TRAINING", and a "TRY DATABRICKS" button. The main heading is "Select a version to get started." Below this are two columns. The left column is titled "FULL-PLATFORM TRIAL" with the subtext "Put Apache Spark to work". It lists features: "Unlimited clusters", "Notebooks, dashboards, production jobs, RESTful APIs", "Interactive guide to Spark and Databricks", "Deployed to your AWS VPC", "BI tools integration", and "14-day free trial (excludes AWS charges)". At the bottom is a "START TODAY" button. The right column is titled "COMMUNITY EDITION" with the subtext "Learn Apache Spark". It lists features: "Mini 6GB cluster", "Interactive notebooks and dashboards", and "Public environment to share your work". At the bottom is a "START TODAY" button.

3. Choose **COMMUNITY EDITION** (or full platform):



Sign Up for Databricks Community Edition

First Name \*

Last Name \*

Company Name \*

To select, begin typing.

Work Email \*

Password \*

Confirm Password \*

Phone Number


What is your intended use case? \*

- Please Select -

How would you describe your role? \*

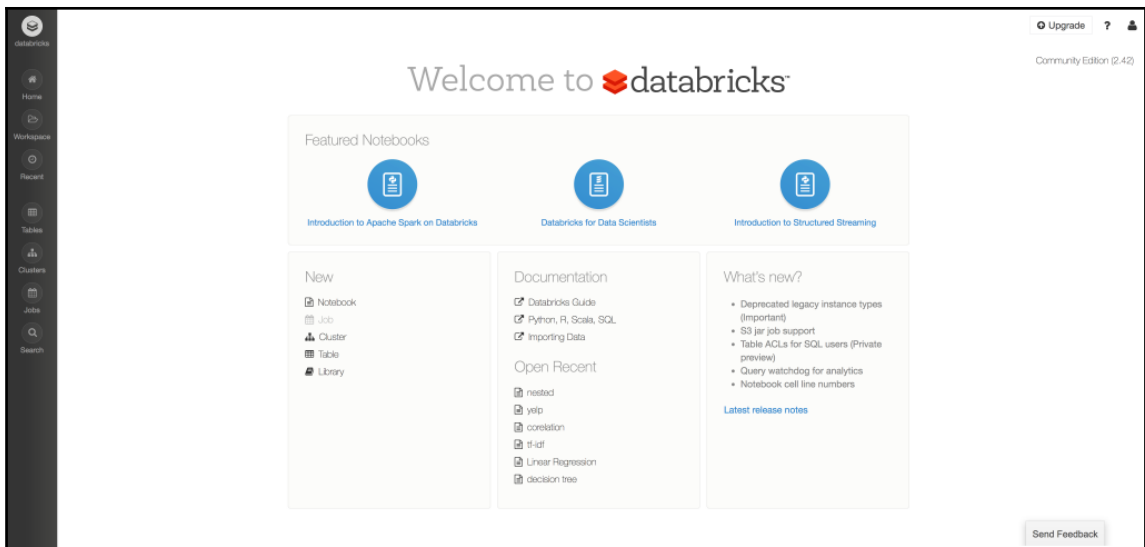
- Please Select -

☐ I'm not a robot

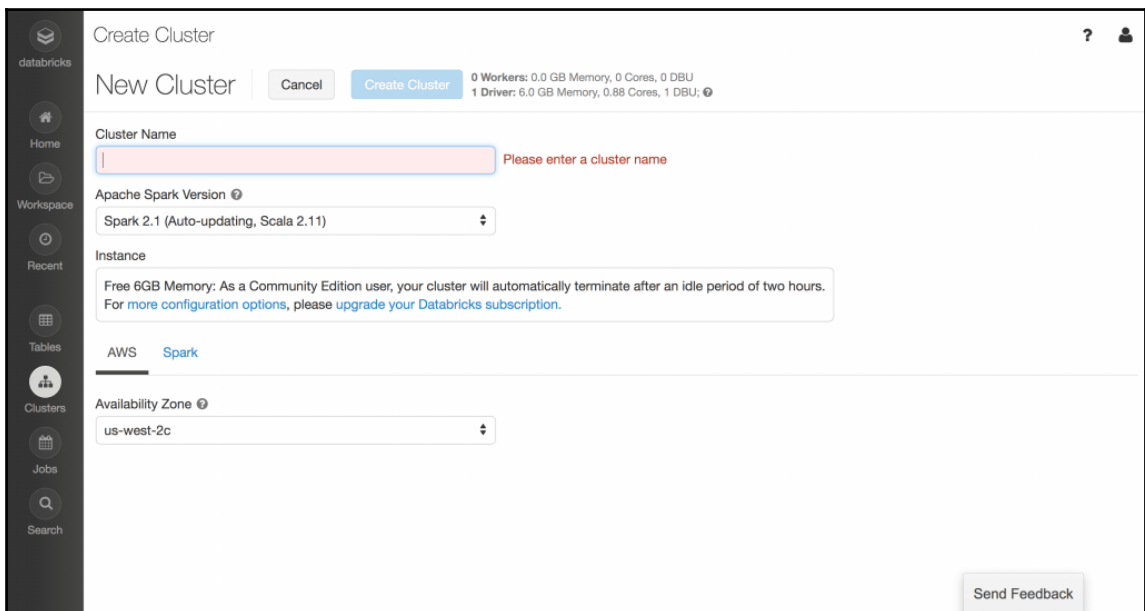
  
reCAPTCHA  
Privacy - Terms

Sign Up

4. Fill in the details and you'll be presented with a landing page, as follows:



5. Click on **Clusters**, then **Create Cluster** (showing community edition below it):



6. Enter the cluster name, for example, `myfirstcluster`, and choose **Availability Zone** (more about AZs in the next recipe). Then click on **Create Cluster**:

The screenshot shows the Databricks Clusters management interface. On the left is a sidebar with navigation icons for Home, Workspace, Recent, Tables, Clusters, Jobs, and Search. The main panel is titled 'Clusters' and has a '+ Create Cluster' button. It is divided into two sections: 'Active Clusters' and 'Terminated Clusters'. The 'Active Clusters' section contains a table with one cluster, 'myfirstcluster', which is in a 'Pending' state (indicated by a green spinning icon). The 'Terminated Clusters' section is currently empty.

Name	Memory	Type	State	Nodes	Spark	Libraries	Notebooks	Default Cluster	Actions
myfirstcluster	6 GB	Community Optimized, Spark 2.1 (Auto-updating, Scala 2.10)	Pending			--	--	Make Default	

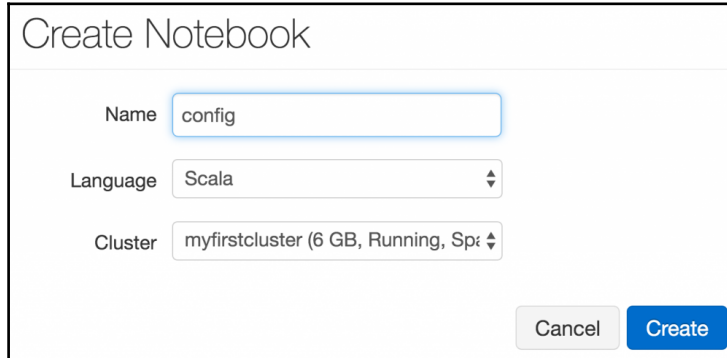
7. Once the cluster is created, the blinking green signal will become solid green, as follows:

This screenshot shows the Databricks Clusters page after the cluster 'myfirstcluster' has started. The cluster's state is now 'Running', indicated by a solid green dot. The 'Nodes' column shows '1 On-demand' node. The 'Spark' column now has links for 'Spark UI' and 'Logs'. The 'Default Cluster' column shows 'Default'. The 'Terminated Clusters' section remains empty.

Name	Memory	Type	State	Nodes	Spark	Libraries	Notebooks	Default Cluster	Actions
myfirstcluster	6 GB	Community Optimized, Spark 2.1 (Auto-updating, Scala 2.10)	Running	1 On-demand	Spark UI Logs	--	--	Default	



8. Now go to **Home** and click on **Notebook**. Choose an appropriate notebook name, for example, `config`, and choose **Scala** as the language:



9. Then set the AWS access parameters. There are two access parameters:
- `ACCESS_KEY`: This is referred to as `fs.s3n.awsAccessKeyId` in SparkContext's Hadoop configuration.
  - `SECRET_KEY`: This is referred to as `fs.s3n.awsSecretAccessKey` in SparkContext's Hadoop configuration.
10. Set `ACCESS_KEY` in the `config` notebook:

```
sc.hadoopConfiguration.set("fs.s3n.awsAccessKeyId", "<replace  
with your key>")
```

11. Set `SECRET_KEY` in the `config` notebook:

```
sc.hadoopConfiguration.set("fs.s3n.awsSecretAccessKey", "  
<replace with your secret key>")
```

12. Load a folder from the `sparkcookbook` bucket (all of the data for the recipes in this book are available in this bucket:

```
val yelpdata =  
  spark.read.textFile("s3a://sparkcookbook/yelpdata")
```

13. The problem with the previous approach was that if you were to publish your notebook, your keys would be visible. To avoid the use of this approach, use **Databricks File System (DBFS)**.