Traditional relational databases are today ineffective with dealing with the challenges presented by Big Data. A Hadoop-based architecture offers a radical solution, as it is designed specifically to handle huge sets of unstructured data.

This book takes you through the journey of building a modern data lake architecture using HDInsight, a Hadoop-based service that allows you to successfully manage high volume and velocity data in the Microsoft Azure Cloud. Featuring a wealth of practical examples, you'll find tips and techniques to provision your own HDInsight cluster to ingest, organize, transform, and analyze data.

While guided through HDInsight, you'll explore the wider Hadoop ecosystem with plenty of working examples on Hadoop technologies including Hive, Pig, MapReduce, HBase, Storm, and analytics solutions including using Excel PowerQuery, PowerMap, and PowerBI.

Who this book is written for

If you want to discover one of the latest tools designed to produce stunning Big Data insights, this book features everything you need to get to grips with your data. Whether you are a data architect, developer, or a business strategist, HDInsight adds value in everything from development, administration, and reporting.

What you will learn from this book

- Explore core features of Hadoop, including the HDFS and YARN, the new resource manager for Hadoop
- Build your HDInsight cluster in minutes and learn how to administer it using Azure PowerShell
- Discover what's new in Hadoop 2.X and the reference architecture for a modern data lake based on Hadoop
- Find out more about a data lake vision and its core capabilities
- Ingest and organize your data into HDInsight
- Utilize open source software to transform data including Hive, Pig, and MapReduce, and make it available for decision makers
- Get to grips with architectural considerations for scalability, maintainability, and security
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 2 “Enterprise Data Lake using HDInsight”
- A synopsis of the book’s content
- More information on HDInsight Essentials Second Edition

About the Author

**Rajesh Nadipalli** currently manages software architecture and delivery of Zaloni’s Bedrock Data Management Platform, which enables customers to quickly and easily realize true Hadoop-based Enterprise Data Lakes. Rajesh is also an instructor and a content provider for Hadoop training, including Hadoop development, Hive, Pig, and HBase. In his previous role as a senior solutions architect, he evaluated big data goals for his clients, recommended a target state architecture, and conducted proof of concepts and production implementation. His clients include Verizon, American Express, NetApp, Cisco, EMC, and UnitedHealth Group.

Prior to Zaloni, Rajesh worked for Cisco Systems for 12 years and held a technical leadership position. His key focus areas have been data management, enterprise architecture, business intelligence, data warehousing, and Extract Transform Load (ETL). He has demonstrated success by delivering scalable data management and BI solutions that empower business to make informed decisions.

Rajesh authored the first version of the book *HDInsight Essentials, Packt Publishing*, released in September 2013, the first book in print for HDInsight, providing data architects, developers, and managers with an introduction to the new Hadoop distribution from Microsoft.

He has over 18 years of IT experience. He holds an MBA from North Carolina State University and a BSc degree in Electronics and Electrical from the University of Mumbai, India.
I would like to thank my family for their unconditional love, support, and patience during the entire process.

To my friends and coworkers at Zaloni, thank you for inspiring and encouraging me.

And finally a shout-out to all the folks at Packt Publishing for being really professional.
HDInsight Essentials Second Edition

We live in a connected digital era and we are witnessing unprecedented growth of data. Organizations that are able to analyze big data are demonstrating significant return on investment by detecting fraud, improved operations, and reduced time to analyze with a scale-out architecture such as Hadoop. Azure HDInsight is an enterprise-ready distribution of Hadoop hosted in the cloud and provides advanced integration with Excel and .NET without the need to buy or maintain physical hardware.

This book is your guide to building a modern data architecture using HDInsight to enable your organization to gain insights from various sources, including smart-connected devices, databases, and social media. This book will take you through a journey of building the next generation Enterprise Data Lake that consists of ingestion, transformation, and analysis of big data with a specific use case that can apply to almost any organization.

This book has working code that developers can leverage and extend in order to fit their use cases with additional references for self-learning.

What This Book Covers

Chapter 1, Hadoop and HDInsight in a Heartbeat, covers the business value and the reason behind the big data hype. It provides a primer on Apache Hadoop, core concepts with HDFS, YARN, and the Hadoop 2.x ecosystem. Next, it discusses the Microsoft HDInsight platform, its key benefits, and deployment options.

Chapter 2, Enterprise Data Lake using HDInsight, covers the main points of the current Enterprise Data Warehouse and provides a path for an enterprise Data Lake based on the Hadoop platform. Additionally, it explains a use case built on the Azure HDInsight service.

Chapter 3, HDInsight Service on Azure, walks you through the steps for provisioning Azure HDInsight. Next, it explains how to explore, monitor, and delete the cluster using the Azure management portal. Next, it provides tools for developers to verify the cluster using a sample program and develop it using HDInsight Emulator.

Chapter 4, Administering Your HDInsight Cluster, covers steps to administer the HDInsight cluster using remote desktop connection to the head node of the cluster. It includes management of Azure Blob storage and introduces you to the Azure scripting environment known as Azure PowerShell.

Chapter 5, Ingest and Organize Data Lake, introduces you to an end-to-end Data Lake solution with a near real life size project and then focuses on various options to ingest data to a HDInsight cluster, including HDFS commands, Azure PowerShell, CloudExplorer, and Sqoop. Next, it provides details on how to organize data using
Apache HCatalog. This chapter uses a real life size sample airline project to explain the various concepts.

Chapter 6, Transform Data in the Data Lake, provides you with various options to transform data, including MapReduce, Hive, and Pig. Additionally, it discusses Oozie and Spark, which are also commonly used for transformation. Throughout the chapter, you will be guided with a detailed code for the sample airline project.

Chapter 7, Analyze and Report from Data Lake, provides you with details on how to access and analyze data from the sample airline project using Excel Hive ODBC driver, Excel Power Query, Powerpivot, and PowerMap. Additionally, it discusses RHadoop, Giraph, and Mahout as alternatives to analyze data in the cluster.

Chapter 8, HDInsight 3.1 New Features, provides you with new features that are added to the evolving HDInsight platform with sample use cases for HBase, Tez, and Storm.

Chapter 9, Strategy for a Successful Data Lake Implementation, covers the key challenges for building a production Data Lake and provides guidance on the success path for a sustainable Data Lake. This chapter provides recommendations on architecture, organization, and links to online resources.
Current IT architecture uses an Enterprise Data Warehouse (EDW) as the centralized repository that feeds several business data marts to drive business intelligence and data mining systems. With the advent of smart connected devices and social media that generate petabytes of data, these current relational EDWs are not able to scale and meet the business needs. This chapter will discuss how to build a modern data architecture that extends the EDW with the Hadoop ecosystem.

In this chapter, we will cover the following topics:

- Enterprise Data Warehouse architecture
- Next generation Hadoop-based Data Lake architecture
- The journey to your Data Lake dream
- Tools and technology in the Hadoop ecosystem
- Use case powered by Microsoft HDInsight

Enterprise Data Warehouse architecture

Over the last 3 decades, organizations have built EDW that consolidates data from various sources across the organization to enable business decisions, typically, related to current operational metrics and future what-if analysis for strategy decisions.
The following figure shows you a typical EDW architecture and also shows how information flows from the various source systems to the hands of business users:

Let's take a look at the stack from bottom to top.

**Source systems**
Typical data sources for an EDW are as follows:

- **OLTP databases**: These databases store data for transactional systems such as customer relationship management (CRM), Enterprise resource planning (ERP), including manufacturing, inventory, shipping, and others.
- **XML and Text Files**: Data is also received in the form of text files, which are generally delimited, or XML, or some other fixed format known within the organization.

**Data warehouse**
A data warehouse has two key subcomponents: storage and processing. Let's review these in detail.
Storage
The following are the key data stores for EDW:

- **EDW**: This is the heart of the complete architecture and is a relational database that hosts data from disparate sources in a consistent format such as base facts and dimensions. It is organized by the subject area/domain and preserves history for several years to enable analytics, trends, and ad hoc queries. An EDW infrastructure needs to be robust and scalable to meet the business continuity and growth requirements.

- **Data marts**: Each data mart is a relational database and is a subset of EDW typically, focusing on one subject area such as finance. It queries base facts from EDW and builds summarized facts and stores them as star or snowflake dimensional models.

- **MDM**: Master data management or MDM is a relational database that stores reference data to ensure consistent reporting across various business units of an organization. Common MDM datasets include products, customers, and accounts. MDM systems require governance to ensure reporting from various data marts that can be correlated and are consistent.

Processing
The following are the key processing mechanisms for EDW:

- **ETL**: Extract, Transform, and Load is a standard data warehouse design pattern that has three key steps: extract from various sources, transform to cleanse, and convert data to the information that is then loaded to various data marts for reporting. There are several tools in the marketplace such as Microsoft SQL Server Integration Services, Informatica, Pentaho, and others. ETL workflows are scheduled typically at daily frequency to update EDW facts and dimensions.

- **SQL-based stored procedures**: This is an alternative to using ETL tools and transform data natively using database features. Most relational databases provide custom stored procedure capabilities such as SQL Server, Oracle, and IBM.
User access

The following are typically used access mechanisms:

- **BI dashboard**: Business intelligence tools access the data from data marts to provide **key performance indicators (KPI)**, scorecards, and dashboards. They allow the business to look at historical trends, current operational performance, perform what-if analysis, and predict future trends. There are several tools in the marketplace, including Microsoft SQL Server Reporting Services, Microsoft Power BI, Oracle Business Intelligence Enterprise Edition (OBIEE), SAP BusinessObjects, Tableau, and so on.

- **Ad hoc analysis**: IT organizations have realized the need to provide business with direct access to certain data for discovery and ad hoc analysis. Excel and several reporting tools fit this need.

- **Operational reports**: These are the day-to-day canned reports required to run the businesses such as daily sales collections, and customer support tickets opened and closed for the current day. These reports are generally required to be near real time and are based on one source system such as customer help desk system. There are several reporting systems such as SQL Server Reporting Services, IBM Cognos, and others that fit this need.

- **Analytics**: Analytical reports look for trends such as how is my customer satisfaction trending over time and the average operational overhead in dollars for support. These reports typically have a one day/week refresh cycle and collect information from multiple sources such as help desk system and PPM. Business intelligence tools typically fit this need.

Provisioning and monitoring

This area of the architecture is responsible for the following functions:

- Managing deployments of ETL code across various environments: development, test, and production
- Monitoring ingestions and jobs to ensure service-level agreements are met
- Operating procedures to recover in case of a failure

Data governance and security

Effective data governance in an enterprise ensures that data is consistent across departments and requires data stewards that identify and enforce the system of records. This requires tools, processes, and people to have high-quality data.
Security on the enterprise data warehouse affects all layers of the architecture and ensures that the right people have the right access to data. There are several technologies that enable fine-grained access at database level, table level, and row level. The filesystem can be encrypted to provide additional level of security.

Pain points of EDW

Based on an IDC research, by 2020, the digital universe will reach 40 ZB (zettabyte), which is a 50-fold growth from the beginning of 2010. (Reference: [http://www.emc.com/about/news/press/2012/20121211-01.htm](http://www.emc.com/about/news/press/2012/20121211-01.htm)). Current IT data architectures based on EDW were not designed to handle this amount of data and are being stretched. The following are the key pain points of EDW:

- **Scale**: A data warehouse was built to handle data in terabytes (TBs) and currently business needs are reaching petabytes (PB). Typically, data warehouse DBAs archive data older than a certain date window like 5 years to address this issue. This data can be a useful information for long-term trends.

- **Cost**: EDW is typically an appliance-based model with proprietary engineered hardware and software supported by vendors such as Teradata, Oracle, IBM, and others. Upgrades are expensive and typically require all servers to be identical.

- **Unstructured data**: Data warehouses struggle to handle unstructured data such as logfiles, social media, and machine data.

- **Timeliness**: To produce business insights, data has to go through several transformations. Current EDW architectures are unable to meet growth and business demands for new insights on their data.

The next generation Hadoop-based Enterprise data architecture

We will now see how modern data architecture addresses the pain points of a legacy EDW and prepares the organization to handle the big wave of data. It is designed to handle both structured and unstructured data in a cost effective and scalable mode. This provides the business with a wide range of new capabilities and opportunities to gain insights.

Instead of a complete EDW replacement, this architecture leverages the existing investment by preserving end-user interfaces that require relational stores. In this model, Hadoop becomes the prime data store and EDW is used to store aggregates.
The following figure shows you how to transition from legacy EDW-based solutions to a hybrid Hadoop-based ecosystem, where EDW's role is reduced to hosting the aggregated data enabling queries via well-established tools that are relational in nature:

Let's take a look at the stack from bottom to top.
**Source systems**
The following are the data sources in the next generation architecture:

- **OLTP**: These databases store data for transactional systems such as CRM, ERP, including manufacturing, inventory, shipping, and others
- **XML and text files**: Data is also received in the form of text files, which are generally delimited, or XML, or some other fixed format known within the organization
- **Unstructured**: Information from various websites, word documents, PDF documents, and other forms that don't have fixed structure or semantics
- **Machine-generated data**: Data captured from automated systems such as telemetry is used primarily for monitoring and performance
- **Audio, video, and images**: Audio, video recordings, and images that are difficult to analyze due to their binary formats
- **Web clicks and logs**: Click stream and logs from websites that provide you with valuable information about consumer behavior
- **Social media**: Messages, tweets, and posts on several social media platforms such as Twitter, Facebook, and Google that provide you with consumer sentiments

**Data Lake**
This is the heart of the architecture that includes storage and compute.

**Storage**
The following are the key data stores for a Data Lake:

- **Hadoop HDFS**: HDFS is a core component of Hadoop that provides a data store that can scale as per the business needs and run on any commodity hardware and is 100 percent open source. In this new architecture, all the source data first lands to HDFS and is then processed and exported to other databases or applications.
- **Hadoop HBase**: HBase is a distributed and scalable NoSQL database that provides low latency option on Hadoop. It uses HDFS to store data files and is hosted on Hadoop cluster.
**Enterprise Data Lake using HDInsight**

- **Hadoop MPP databases**: MPP stands for massively parallel processing where data can be stored in HDFS and access to MPP can be through SQL or APIs enabling easier integration to existing applications. We are seeing a lot of innovations in this area.

- **Legacy EDW and DM**: This architecture leverages current investment on EDW, DM, and MDM. The size of EDW, however, is reduced as HDFS takes the heavy lifting and only the summary data is hosted in EDW.

**Processing**

The following are the processing mechanisms for Data Lake:

- **Hadoop Batch (MapReduce)**: MapReduce is a core Hadoop component and is a good fit to replace ETL batch jobs. MapReduce has built-in fault tolerance and runs on the same HDFS data nodes that can scale when the demand increases.

- **Hadoop Streaming (Storm)**: Storm allows the distributed real-time computation system on top of the Hadoop cluster. A good use of this technology is the real-time security alerts on dashboards that require low latency and cannot wait for a complete batch execution.

- **Hadoop Real time (Tez)**: Tez is an extensible framework that allows developers to write native YARN applications that can handle workloads ranging from interactive to batch. Additionally projects such as Hive and Pig can run over Tez and benefit from performance gains over MapReduce.

- **Hadoop Oozie workflows**: This enables creation of workflow jobs to orchestrate Hive, Pig, and MapReduce tasks.

- **Legacy ETL and stored procedures**: This block in the architecture represents the legacy ETL code that will gradually shrink as Hadoop ecosystem builds more capabilities to handle various workloads.

**User access**

This part of the architecture remains identical to the traditional data warehouse architecture with BI dashboards, operational reports, analytics, and ad hoc queries. This architecture does provide additional capabilities such as fraud detection, predictive analytics, 360 views of customers, and longer duration of history reports.
Provisioning and monitoring
The new architecture will also require provisioning and monitoring capabilities such as the EDW-based architecture that includes managing deployments, monitoring jobs, and operations.

A Data Lake architecture does have additional components from the Hadoop stack that add to the complexity and we need new tools that typically come with the Hadoop distribution such as Ambari.

Data governance, security, and metadata
Data governance process and tools are built for EDW-based architecture that can be extended to the Data Lake base architecture.

Current tools for security on Hadoop-based Data Lake are not sophisticated but will improve in the next few years as adoption of Hadoop is gaining steam.

The core of Hadoop is essentially a filesystem whose management requires metadata for organizing, transforming, and publishing information from these files. This requires a new metadata component for the Data Lake architecture. Apache HCatalog does have some basic metadata capabilities but needs to be extended to capture operational and business-level metadata.

Journey to your Data Lake dream
Hadoop's HDFS and YARN are the core components for the next generation Data Lake; there are several other components that need to be built to realize the vision. In this section, we will see the core capabilities that need to be built in order to enable an Enterprise Data Lake. The following are the key components that need to be built for an effective Data Lake:

Let us look into each component in detail.
Ingestion and organization
Data Lake based on HDFS has a scalable and distributed filesystem that requires a scalable ingestion framework and software that can take in structured, unstructured, and streaming data.

A managed Data Lake requires data to be well-organized and this requires several kinds of metadata. The following are key metadata that require management:

- **File inventory**: What, when, and who about files ingested to Hadoop?
- **Structural metadata**: What is the structure of a file such as XML, HL7, CSV, and TSV?

Hadoop does work well with Avro sequence file where the metadata and data are stored together.

- **User-defined information**: Additional information that the user is provided with such as comments and tags for a file.

Transformation (rules driven)
A key design consideration for next generation Data Lakes is easy for building new transformations as demanded by business. This requires a workflow to orchestrate the sequence of rules-based transformations; where business users can author rules.

Access, analyze, and report
To democratize access to data from a Data Lake, the following patterns of access need to be enabled:

- **Direct data access**: Access to data via programs and ad hoc queries for power users via tools
- **BI tools**: Access via BI tools for dashboards and reports
Tools and technology for Hadoop ecosystem

Next generation architecture includes Hadoop-based projects that complement the traditional RDBMS systems. The following table highlights key projects that are organized by Data Lake capabilities:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingest</td>
<td>Flume</td>
<td>This is a distributed and reliable software to collect large amounts of data from different sources such as logfiles in a streaming fashion in Hadoop.</td>
</tr>
<tr>
<td>Ingest</td>
<td>Sqoop</td>
<td>This tool is designed to transfer data between Hadoop and RDBMS such as Oracle, Teradata, and SQL Server.</td>
</tr>
<tr>
<td>Organize</td>
<td>HCatalog</td>
<td>This tool stores metadata for Hadoop, including file structures and formats. It provides an abstraction and interoperability across various tools such as Pig, MapReduce, Hive, Impala, and others.</td>
</tr>
<tr>
<td>Transform</td>
<td>Oozie</td>
<td>This is a workflow scheduler system to manage Apache Hadoop jobs, which can be MapReduce, Hive, Pig, and others. It provides developers greater control over complex jobs and also makes it easier to repeat those at predetermined intervals.</td>
</tr>
<tr>
<td>Transform</td>
<td>YARN</td>
<td>With Hadoop 2.x, YARN takes over all the resource management capabilities on top of Hadoop so that it can now serve broad types of workloads such as batch, interactive, streaming, in-memory, and others.</td>
</tr>
<tr>
<td>Transform and Access</td>
<td>Pig</td>
<td>A scripting language such as Python that abstracts MapReduce and is useful for data scientists.</td>
</tr>
<tr>
<td>Transform</td>
<td>Spark</td>
<td>This is a fast and general compute engine for Hadoop with a directed acyclic graph (DAG) execution engine that supports complex data flows and in-memory computing. There are number of tools and projects that now leverage Spark as their execution engine, for example, Talend, which is an ETL tool.</td>
</tr>
<tr>
<td>Transform</td>
<td>Tez</td>
<td>Tez is a framework that allows complex DAG of tasks for processing data using native YARN APIs. It is designed to empower end users by providing expressive dataflow definitions and has significant performance gains over MapReduce.</td>
</tr>
<tr>
<td>Capability</td>
<td>Tool</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Access</td>
<td>Hive</td>
<td>Data warehouse infrastructure that provides SQL-like access on HDFS. This is suited for ad hoc queries on Hadoop that abstract MapReduce.</td>
</tr>
<tr>
<td>Access</td>
<td>Excel</td>
<td>With HDInsight, you can connect Excel to HDFS via Hive queries using ODBC to analyze and visualize your data.</td>
</tr>
<tr>
<td>Access</td>
<td>Mahout</td>
<td>This framework allows machine learning and data mining on top of Hadoop.</td>
</tr>
<tr>
<td>Operations</td>
<td>Ambari</td>
<td>A Web-based provisioning, managing, and monitoring tool for Hadoop cluster. It also provides a dashboard with cluster health and the ability to see current running jobs in a user-friendly manner.</td>
</tr>
</tbody>
</table>

**Use case powered by Microsoft HDInsight**

Let's take a look at a practical use case powered by Microsoft HDInsight that demonstrates the value of next generation Data Lake architecture.

**Problem statement**

The Virginia Bioinformatics Institute collaborates with institutes across the globe to locate undetected genes in a massive genome database that leads to exciting medical breakthroughs such as cancer therapies. This database size is growing exponentially across the 2,000 DNA sequencers and is generating 15 petabytes of genome data every year. Several universities lack storage and compute resources to handle this kind of workload in a timely and cost-effective manner.

**Solution**

The institute built a solution on top of Windows Azure HDInsight service to perform DNA sequencing analysis in the cloud. This enabled the team to analyze petabytes of data in a cost-effective and scalable manner. Let's take a look at how the Data Lake reference architecture applies to this use case. The following figure shows you the solution architecture:
Let's review the stack in detail.

**Source systems**
Source data for the initiative are various DNA sequence files and additional text files that are generally from researchers' desktop machines.

**Storage**
All big data was stored in the Azure cloud ecosystem thereby reducing the need for an in-house data center. Azure Blob storage was used to persist data, independent of HDInsight cluster; this resulted in additional cost savings by only paying for the HDInsight compute services on-demand without the loss of any data.
Processing
For processing the DNA sequence files, Virginia Tech built an application called SeqInCloud based on Broad Institute's Genome Analysis Toolkit (GATK). GATK is a toolkit for analyzing sequencing data with a main focus on variant discovery and genotyping.

SeqInCloud runs GATK on HDInsight MapReduce for maximum portability and scalability. It also has features for data partitioning, data transfer, and storage optimization on Windows Azure.

User access
The end customers for this project were data scientists for which the team built another custom application called CloudFlow. This is a workflow management framework installed on a researcher's PC and is used for all the interactions with Azure HDInsight Service. CloudFlow allows you to compose a pipeline with several MapReduce segments that utilize both client and cloud resources and allow user-defined plugins for data transformation.

The users can also access Azure Blob storage directly from a browser that allows collaboration across different stakeholders.

Benefits
The solution based on HDInsight for DNA sequencing has several benefits out of which the key ones are listed as follows:

- **Scalable solution**: As the cluster is on Azure cloud, it can be scaled to higher number of nodes if the demand and data volumes increase.
- **Shorter time for analysis**: With HDInsight, the team was able to analyze DNA sequences in an effective and timely manner.
- **Significant cost savings**: When compared to millions of dollars required to build a data center to handle this work load, the Azure HDInsight service has proven to be a cost-effective alternative. Since data is stored in Azure Blob storage, the cluster can be torn down but data is still preserved for future reference thereby further reducing the compute expenses.
• **Improved collaboration:** Since data is stored in the cloud, sharing public datasets became really easy with other researchers spread across the country. Additionally, data can be accessed by various types of end devices, which include mobile phones and tablets. This approach leads to new opportunities such as genome analysis at a hospital, which could lead to faster, prescribed treatments.

• **Easy visualization impact:** With HDInsight, business users benefited from quick visualization using Power BI dashboards with Office 365/Excel 2013 and were able to analyze data in a better and faster way.

The National Human Genome Research Institute (NHGRI) has been tracking costs associated with DNA sequencing since 2001 and the following figure shows you the trend due to such innovations in universities such as Virginia Tech:

![Cost per Genome Graph](http://www.genome.gov/sequencingcosts/)


Enterprise Data Lake using HDInsight

Summary

Most organizations have built expensive EDWs as the centralized data repositories serving critical business decisions. The relational EDW-based architecture is struggling to handle the data growth and ability to provide near real-time metrics. Hadoop-based Data Lake has emerged as a cost-effective alternative to EDW providing access to real-time information to business users in a more agile fashion.

Microsoft HDInsight Azure-based service is well-positioned to enable a modern Data Lake on the cloud thereby further reducing operational and data center costs.

In the next chapter, we will build, configure, and monitor a new HDInsight cluster, which is the first step in building a Data Lake.
Where to buy this book