HADOOP FOR WINDOWS

SUCCINCTLY

BY DAVE VICKERS

SUCCINCTLY EBOOK SERIES

Syncfusion

www.dbooks.org

Hadoop for Windows Succinctly

By
Dave Vickers

Foreword by Daniel Jebaraj



Copyright © 2019 by Syncfusion, Inc.

2501 Aerial Center Parkway
Suite 200
Morrisville, NC 27560
USA
All rights reserved.

Important licensing information. Please read.

This book is available for free download from www.syncfusion.com on completion of a registration form.

If you obtained this book from any other source, please register and download a free copy from www.syncfusion.com.

This book is licensed for reading only if obtained from www.syncfusion.com.

This book is licensed strictly for personal or educational use.

Redistribution in any form is prohibited.

The authors and copyright holders provide absolutely no warranty for any information provided.

The authors and copyright holders shall not be liable for any claim, damages, or any other liability arising from, out of, or in connection with the information in this book.

Please do not use this book if the listed terms are unacceptable.

Use shall constitute acceptance of the terms listed.

SYNCFUSION, SUCCINCTLY, DELIVER INNOVATION WITH EASE, ESSENTIAL, and .NET ESSENTIALS are the registered trademarks of Syncfusion, Inc.

Technical Reviewer: James McCaffrey

Copy Editor: Courtney Wright

Acquisitions Coordinator: Tres Watkins, content development manager, Syncfusion, Inc.

Proofreader: Darren West, content producer, Syncfusion, Inc.

www.dbooks.org

Table of Contents

The Story Behind the *Succinctly* Series of Books **6**

About the Author **8**

Acknowledgements **9**

Introduction **10**

Windows and Linux as environments for Hadoop 11

Distributions of Hadoop for Windows 14

Chapter 1 Installing Hadoop for Windows **20**

Choosing a Hadoop distribution to install 20

Apache Hadoop installation prerequisites 20

Java installation for Hadoop for Windows 20

Apache Hadoop installation 22

Connecting to data sources within HDFS 31

Summary 37

Chapter 2 Enterprise Hadoop for Windows **38**

Physical and virtual enterprise Hadoop distributions for Windows 38

Network setup and installation 40

Suitable network components 40

Suitable server hardware and Windows licensing 41

Security and Active Directory 43

Enterprise Hadoop installation 44

Creating a multi-node Hadoop cluster in Windows 47

Cluster maintenance and management 51

Working with local development and live production clusters 55

Summary 62

Chapter 3 Programming Enterprise Hadoop in Windows 63

Hadoop performance and memory management within Windows Server 63

Hive data types and data manipulation language 69

Enterprise data ingestion and data storage 70

Executing data-warehousing tasks using Hive Query Language over MapReduce.....75

Utilizing Apache Pig and using Sqoop with external data sources 81

Summary.....91

Chapter 4 Hadoop Integration and Business Intelligence (BI) Tools in Windows 93

Hadoop Integration in Windows and SQL Server 2019 CTP 2.0.....93

The choice of BI tools for Hadoop for Windows 103

A new breed of BI for big data.....104

Connecting BI tools to Hadoop in Windows 106

Three features from Hadoop in Linux I'd like to see more of 128

Chapter 5 When Data Scales, Does BI Fail?.....130

Preparing the Dataset.....130

Using Tableau with large datasets in Windows Hadoop 131

Using Azure Data Studio with large datasets in Windows Hadoop 134

Using Arcadia Data with large datasets in Windows Hadoop.....135

Using Syncfusion Dashboard Designer with large datasets in Windows Hadoop.....140

Conclusion 148

Improvements and feature sets 148

Hadoop user Communities for Windows and Linux 148

www.dbooks.org

The Story Behind the *Succinctly* Series of Books

Daniel Jebaraj, Vice President
Syncfusion, Inc.

Staying on the cutting edge

As many of you may know, Syncfusion is a provider of software components for the Microsoft platform. This puts us in the exciting but challenging position of always being on the cutting edge.

Whenever platforms or tools are shipping out of Microsoft, which seems to be about every other week these days, we have to educate ourselves, quickly.

Information is plentiful but harder to digest

In reality, this translates into a lot of book orders, blog searches, and Twitter scans.

While more information is becoming available on the Internet and more and more books are being published, even on topics that are relatively new, one aspect that continues to inhibit us is the inability to find concise technology overview books.

We are usually faced with two options: read several 500+ page books or scour the web for relevant blog posts and other articles. Just as everyone else who has a job to do and customers to serve, we find this quite frustrating.

The *Succinctly* series

This frustration translated into a deep desire to produce a series of concise technical books that would be targeted at developers working on the Microsoft platform.

We firmly believe, given the background knowledge such developers have, that most topics can be translated into books that are between 50 and 100 pages.

This is exactly what we resolved to accomplish with the *Succinctly* series. Isn't everything wonderful born out of a deep desire to change things for the better?

The best authors, the best content

Each author was carefully chosen from a pool of talented experts who shared our vision. The book you now hold in your hands, and the others available in this series, are a result of the authors' tireless work. You will find original content that is guaranteed to get you up and running in about the time it takes to drink a few cups of coffee.

Free forever

Syncfusion will be working to produce books on several topics. The books will always be free. Any updates we publish will also be free.

Free? What is the catch?

There is no catch here. Syncfusion has a vested interest in this effort.

As a component vendor, our unique claim has always been that we offer deeper and broader frameworks than anyone else on the market. Developer education greatly helps us market and sell against competing vendors who promise to “enable AJAX support with one click,” or “turn the moon to cheese!”

Let us know what you think

If you have any topics of interest, thoughts, or feedback, please feel free to send them to us at succinctly-series@syncfusion.com.

We sincerely hope you enjoy reading this book and that it helps you better understand the topic of study. Thank you for reading.

Please follow us on Twitter and “Like” us on Face-book to help us spread the word about the *Succinctly* series!





www.dbooks.org

About the Author

After graduating from the biggest school of Architecture and Town Planning in Europe, Dave Vickers's professional career began at the Acer Engineering Group in London.

Dave has provided technical solutions for multi-national organizations in the telecommunications, financial services, and digital media sectors, among others. He has also been involved in projects including Open Underwriter, the world's leading open source insurance platform.

The absence of turnkey interfaces between content providers and network operators led Dave to create DavincksOne. Its aim is developing solutions that meet the 5G Development and Validation Platform standards for global, industry-specific networks—in particular, the 5G-XCast Media and Entertainment vertical and Object-Oriented Broadcasting solutions.

Acknowledgements

I would like to thank my father, whose ethic for hard work and self-sacrifice made my education possible, and my mother, who has supported me in everything I do, and with who I have shared so much. I dedicate this book to my younger sister Natasha—what she has achieved in her young life is an inspiration— and to our beloved Leonora, gone but not forgotten, and loved by all. To all those generous enough to share their knowledge with me over the years, thank you.

www.dbooks.org

Introduction

Hadoop is a collection of utilities that work together to enable distributed storage and processing of very large datasets. Since its inception, it has almost exclusively been associated with Linux operating systems. An example of this is the number of text books and publications focusing on Hadoop for Linux. Conversely, the number of text books focusing on Hadoop for Windows is almost non-existent.

It's important at this stage to be clear about what I mean when I say “Hadoop for Windows.” Hadoop for Windows refers to Hadoop running directly on the Microsoft Windows operating system, and native support for Windows must be provided.

Hadoop for Windows does not include:

- Hadoop running on Windows via an emulator or virtual machine
- Hadoop accessed via the cloud from a Windows machine
- Hadoop running on Windows via any kind of third-party container system

There are many online examples of people recommending the preceding options to run Hadoop for Windows. They seem unaware that Hadoop can be installed directly onto Windows, so it’s not something they consider. The aim of this book is firstly to make people aware that Hadoop runs perfectly on Windows. The subsequent aim is to guide the reader in the installation and usage of Hadoop on the Windows platform.

Although this book is about Hadoop for Windows, it would not be credible to omit referring to Linux where it’s pertinent to do so. By comparing the two environments as operating systems for Hadoop, we may discover the reasons behind the popularity of Linux. That said, there are some fairly obvious reasons why Hadoop has been so heavily associated with Linux. Hadoop is open source, so an open source operating system such as Linux was always a natural pairing; both are also available free of charge. As important as the role of Linux has been, the role of Microsoft is equally important. Microsoft deployed Hadoop in the cloud via HDInsight on Microsoft Azure.

HDInsight was the result of Hortonworks collaborating with Microsoft, and was based upon the Hortonworks Hadoop distribution. A desktop emulator version was available, and Hortonworks released its own Hadoop distribution for Windows, which is now archived. Since then, Hortonworks has promoted its Hadoop Sandbox for Windows that only runs on a virtual machine. The end of HDInsight for Windows finally came in July 2018, leaving only HDInsight for Linux. Naturally, it raised eyebrows that Microsoft ended HDInsight for Windows, and various questions were raised, including: Why was Hadoop for Windows named HDInsight? If you asked IT professionals if they knew Microsoft had released Hadoop for Windows, how many would know? The reality is that Microsoft has never released a multi-node version of Hadoop for on-premises usage.

The optimum solution may have been to offer the same HDInsight solution on premises that was offered in the cloud. There have been numerous products that haven’t done as well as they could have, as they were primarily cloud-based. IBM Watson Analytics springs to mind—it’s an intelligent piece of software, but unavailable on premises, so it lost on-premises sales.

An on-premises setup puts you in control, but in Azure cloud you can only use Ranger, Kafka, Interactive Query, and Spark on HDInsight for Linux. You can't use them in the retired HDInsight for Windows, nor can you create or resize Windows clusters. Despite this, Microsoft feels that its Hadoop deployment has an edge over competitors by using Azure Storage to store data, instead of on-premises storage or nodes.

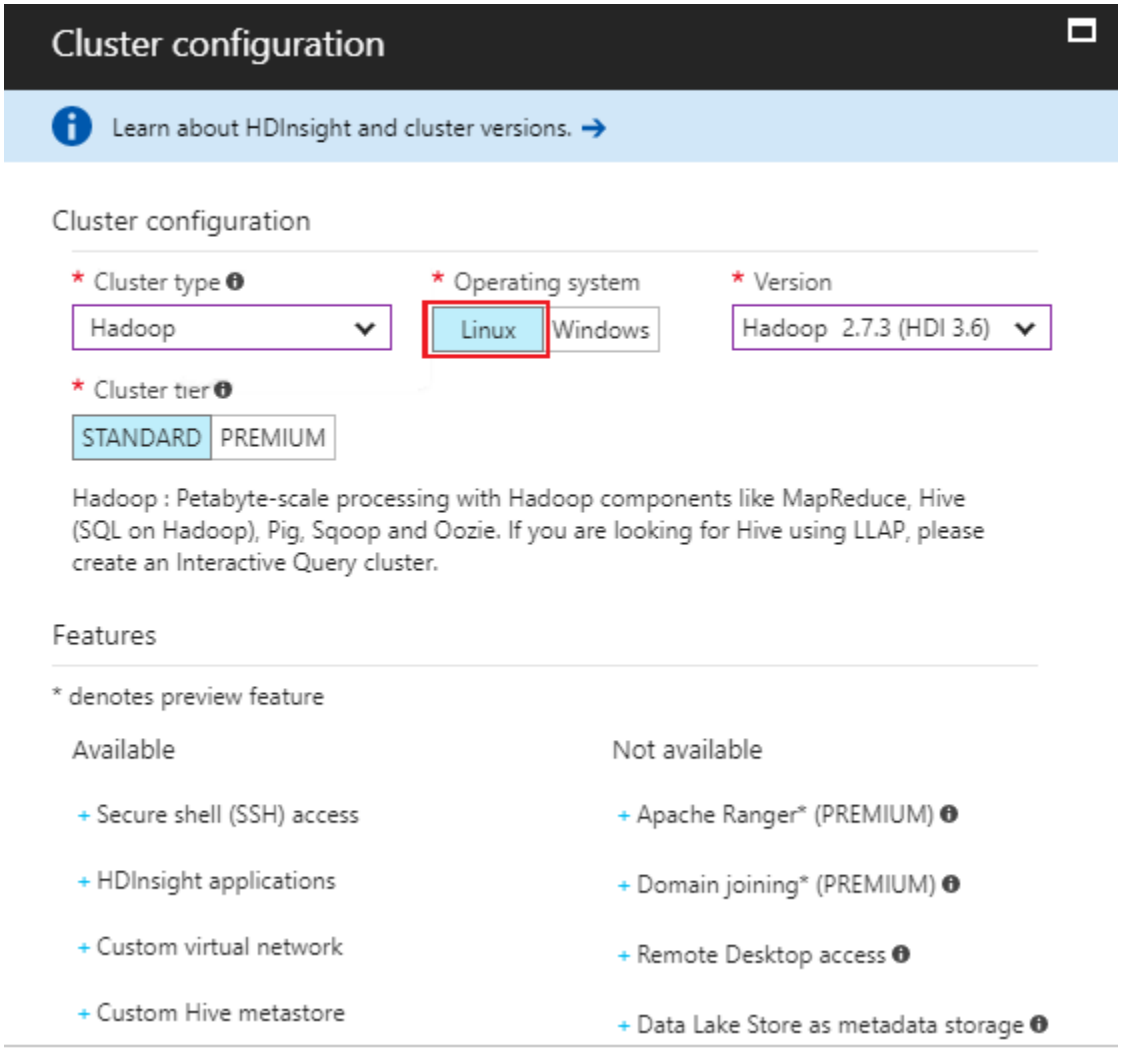


Figure 1: Cluster creation on HDInsight for Windows was retired in July 2018

These factors together may give the impression that running Hadoop for Windows has been a challenge. All the more reason for letting users know they can easily run Hadoop on a supported Windows platform, just like Linux users run Hadoop on supported Linux platforms.

Windows and Linux as environments for Hadoop

In addition to the points raised previously, it's important to understand the other reasons that influence people's choice of using Hadoop on Linux or Windows. Requirements for Hadoop on both Windows and Linux reveal less-demanding hardware requirements for Linux.

www.dbooks.org

Table 1: General requirements for Hadoop for Windows and Linux

	Hadoop for Windows	**Hadoop on Linux**
64-bit processor	Quad/Hex/Octo Cores 2 GHz +	4 CPUs
RAM	8 GB, 16 GB to run all features	8 GB
Hard disk space	100 GB	60 GB
Monitor resolution	800 × 600 or higher	640 × 480 or higher
Architecture type	AMD64, Intel x86, x86_64	AMD64, Intel x86, x86_64

Table 2 shows that it's not just Hadoop, but also the Windows operating system, that is creating more demanding requirements. Windows Server also has a cost, whereas Ubuntu Server is free of charge.

Table 2: Windows Server 2016 vs. Linux Ubuntu Server 16.04 LTS minimum requirements

	Windows Server 2016	**Ubuntu Server 16.04LTS**
64-bit processor	1.4 GHz	1 GHz or 300 MHz (minimum)
RAM	512 MB or 2 GB for desktop experience	512 MB or 256 MB minimal installation
Hard disk space	32 GB SATA drive	1.5 GB–5 GB
Monitor resolution	1024 × 768 or higher	640 × 480 or higher
Graphical user interface	Optional. Installed with or without GUI.	Ubuntu Server has no default GUI, external tools available
Architecture type	AMD64, Intel x86_64, x86	AMD64, Intel x86_64, x86 ARM
Cost per server	Windows Standard $883, Datacenter $6,155	$0

These factors will clearly influence people's choices when it comes to acquisition of a big data solution. But in a corporate environment you would need internal or external support for your solution, and this has a cost. The yearly support costs for Ubuntu Advantage support from Canonical highlights the cost of using Linux Server in such an environment.

Table 3: 2018 Ubuntu Advantage Server support – Yearly costs per node in U.S. dollars

	Physical node	Virtual node
Essential support	$225	$75
Standard support	$750	$250
Advanced support	$1,500	$500

Using Linux Ubuntu Desktop in a similar supported environment also has a cost, so essentially, Linux in a corporate environment is not free.

Table 4: 2018 Ubuntu Desktop support – Yearly costs per node in US dollars

	Physical node	Virtual node
Standard support	$150	$150
Advanced support	$300	$300

Finally, Table 5 shows that the zero-cost Ubuntu desktop actually has higher system requirements than Windows 10. This is not surprising, as Windows 10 is now an older system with older basic requirements. Newer releases of Linux, such as Red Hat Enterprise Workstation, offer unlimited RAM, the ability to use a second CPU socket, and four virtualized guests.

Table 5: Windows 10 and Linux Ubuntu 16.04 Desktop recommended minimum requirements

	Windows 10 Desktop	Ubuntu Desktop 16.04 LTS
64-bit processor	1 GHz	2 GHz dual core
RAM	2 GB	2 GB
Hard disk space	20 GB	25 GB
Monitor resolution	800 × 600 or higher	1024 × 768 or higher
Architecture type	AMD64, Intel x86, x86_64, ARM	AMD64, Intel x86, x86_64, ARM
Cost	Home $119, Pro $199	$0

By its very nature, the computer industry does not stand still. The long-held belief that Microsoft is the operating system of choice for most users is still true. At the same time, Red Hat, Inc. became the first open-source provider to reach revenues of more than a billion dollars. This fact cannot be underestimated; the same can be said for Red Hat Enterprise Server, which many corporate users prefer over Windows Server.

www.dbooks.org

Microsoft is aware of these challenges; the reason it's possible to run Apache Hadoop for Windows is because Microsoft made significant changes to the Windows operating system. Official Apache Hadoop releases have included native support for Windows since Hadoop 2.2. This has opened the door for Windows-based Hadoop and Windows-based Hadoop vendors.

With so many data applications running on Windows, it's important that they're able to connect to Hadoop as efficiently as possible. The optimum solution is for Hadoop to be available within the same Windows environment as the applications themselves. Microsoft Power BI, for example, can connect to 80 data sources, including Apache Hadoop File (HDFS) and other big data solutions, without ODBC for Hive. I know of no other business intelligence application with the ability to connect to so many systems.

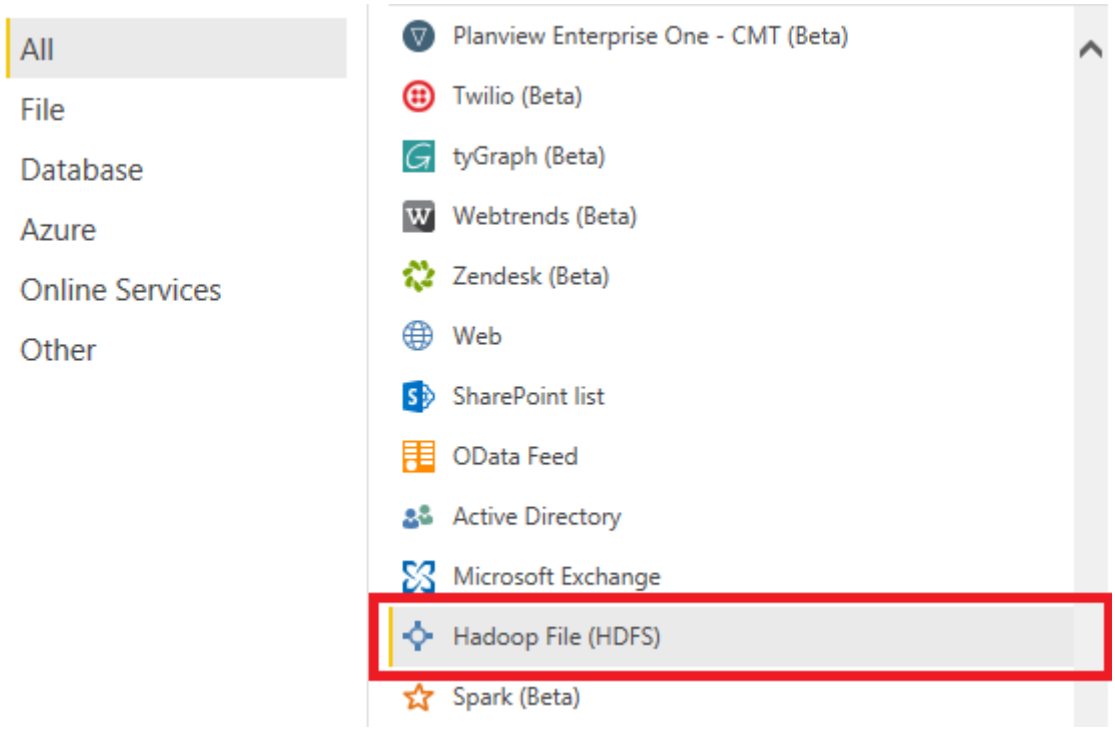


Figure 2: HDFS support in Microsoft Power BI

Distributions of Hadoop for Windows

Apache Hadoop

The current version of Apache Hadoop is 2.9.2, and native support has been provided for Windows since Apache Hadoop 2.2. Apache Hadoop 3.1.1 is also a popular release of the software and good resources include Hadoop Wiki and the Apache Hadoop download page. Windows owners used to automatic installers may find these resources for installing Hadoop problematic. In the wider context, this includes Linux users having problems installing multi-node Hadoop. This has led to companies taking Apache Hadoop and bundling it with other tools to create user-friendly Hadoop installers.

For the best experience, you require the solution to be 100 percent Apache Hadoop-compliant. Apache Hadoop is free and open source, and by installing it yourself, you will learn more than you will by using an automated installer. Once you get past the installation, the functions and commands you use in Apache Hadoop are the same commands you use in other vendors' bundled Hadoop distributions, provided they are Apache Hadoop-compliant.

Microsoft HDInsight

HDInsight for Windows was the Microsoft Hadoop offering based on the Hortonworks Hadoop platform. As mentioned previously, it was available via Microsoft Azure cloud, and on premises via HDInsight Desktop Emulator. It was retired in July 2018 in favor of HDInsight for Linux. You will notice in the following screenshot that the latest HDInsight Emulator was a 2014 release, so as of 2018, was four years old. The age of the last release may give some indication of Microsoft's commitment to on-premises HDInsight.

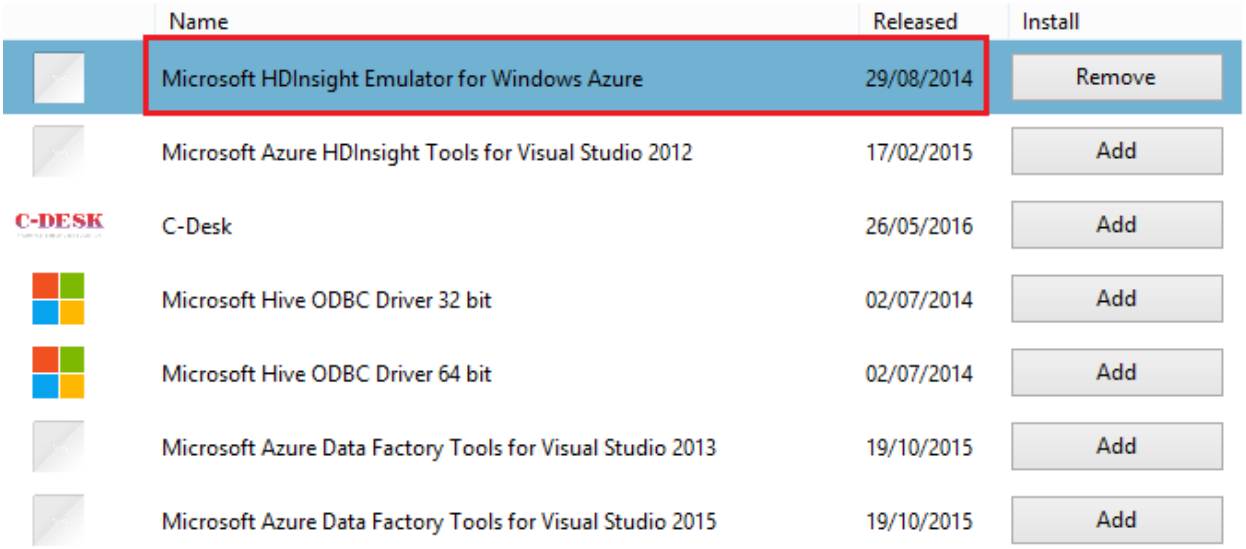


Figure 3: HDInsight Desktop Emulator for Windows

This means the only way to access a Hadoop distribution from Microsoft is cloud-based HDInsight for Linux. This can be a cheaper option than on-premises Hadoop, and it's supported by Canonical, which supports HDInsight 3.4 onward on Ubuntu Server. Ubuntu Server is now a certified guest system for HDInsight access.

The reasons Microsoft gave for HDInsight now only being supported for Linux were:

- A large community and ecosystem for support
- Ability to exercise active development by the open-source community for Hadoop and other big-data technologies
- Faster time-to-market for open-source big data technologies through HDInsight service

If Microsoft makes HDInsight open source, I can see their preceding reasons benefiting HDInsight over time. But placing a non-open source project into an open source community is a different proposition. Remember, there are companies with much bigger on-premises Hadoop installations than your typical Microsoft Azure customer. They may be more interested in other Microsoft big-data innovations with on-premises options.

www.dbooks.org

In November 2018, the public preview of Microsoft Azure Service Fabric Mesh became available. It's an upgrade of Azure Service Fabric that allows you to build applications on shared nodes that scale as the need arises. Many elements of Microsoft Azure already run on Service Fabric, which means that Hadoop is not the only big-data option for Microsoft. It's straightforward to install a one- or five-node Azure Service Fabric system, thanks to the time and resources Microsoft invested in it. If you choose to install it, ensure that there are no warnings or errors, and that the Cluster Health State shows **OK**, as highlighted in Figure 4.

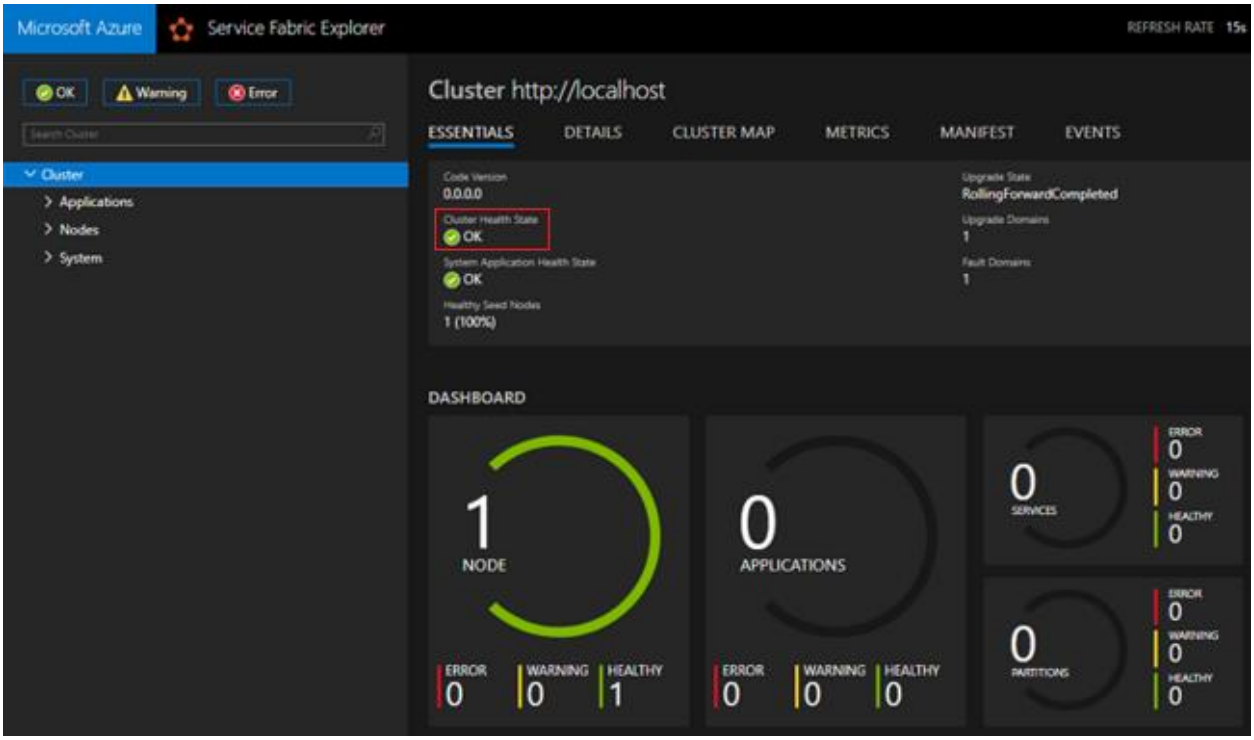


Figure 4: Microsoft Azure Service Fabric Explorer installed locally

Hortonworks

Hortonworks on-premises Hadoop took the form of an offline installer for Windows. As previously mentioned, the product is now archived, and Hortonworks promotes its Sandbox running on VMware or Virtual Box in Windows.

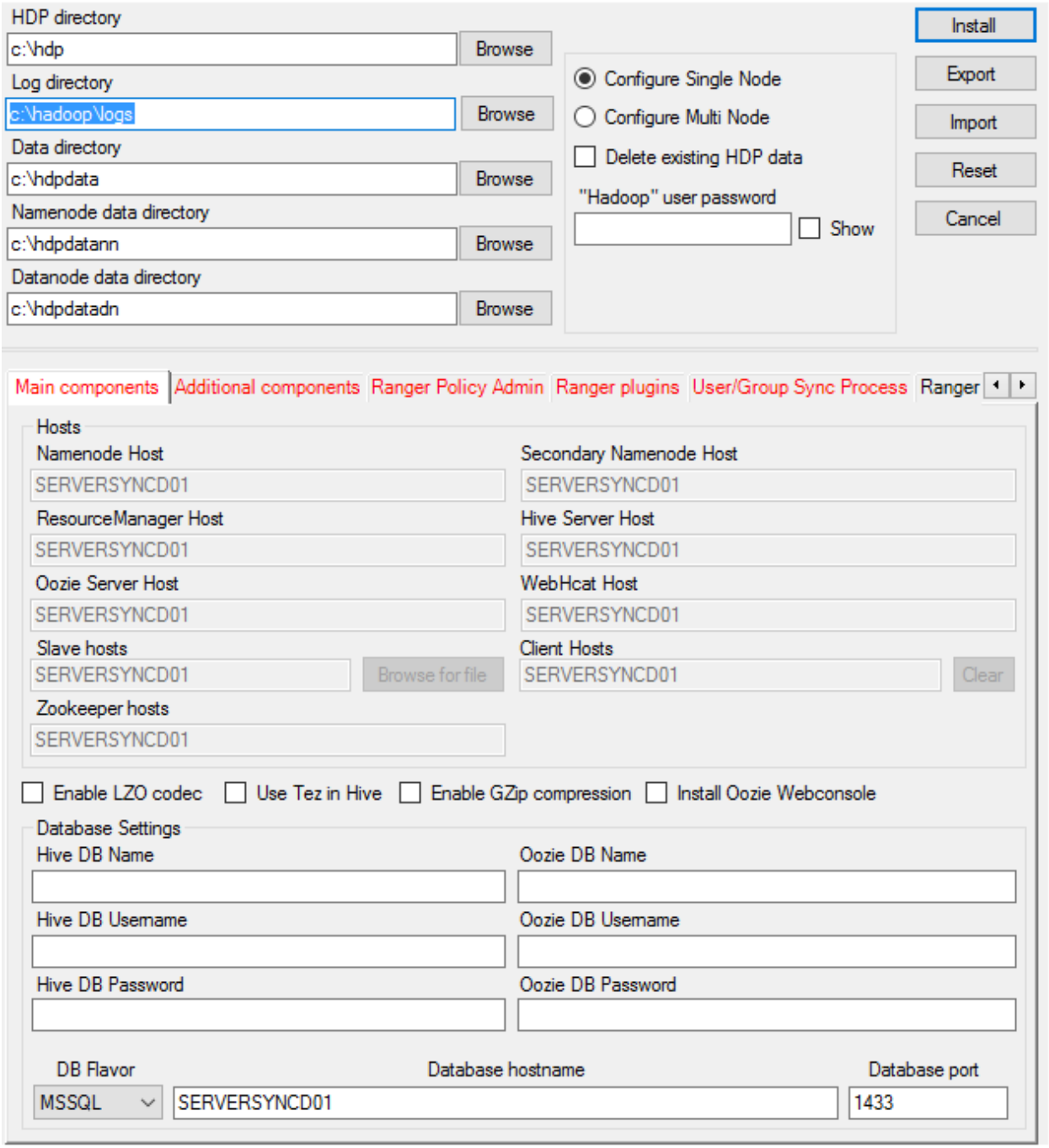


Figure 5: Hortonworks HDP v2.3, a Hadoop installer for Windows

Hortonworks Sandbox is able to bypass issues associated with installing Hadoop on premises. That said, in my experience Hortonworks HDP installer works perfectly on Windows, so why is the Sandbox needed? Sure, it's not the most intuitive installer, but don't let this detract from the positives. I found it very fast and very thorough, and it provided smoke tests to ensure that Pig, Hive, and the Hadoop Ecosystem were working perfectly.

www.dbooks.org

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.1.2.3.4.0-3485      0.15.0.2.3.4.0-3485     hadoop  2018-12-17 12:10:44     2018-12-17 12:13:01     UNK

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   Min
me      Alias   Feature Outputs
job_1545025302968_0003  1       0       47      47      47      47      0       0       0       0       A,B
020/user/hadoop/out-1545048589.log,

Input(s):
Successfully read 272 records (9168 bytes) from: "hdfs://SERVERSYNCD01:8020/user/hadoop/hadoop-1545048589"

Output(s):
Successfully stored 272 records (595 bytes) in: "hdfs://SERVERSYNCD01:8020/user/hadoop/out-1545048589.log"
```

Figure 6: Post Installation Smoke Test for Pig on Hortonworks HDP v2.3 for Windows

While the HDP installer provides a classic Hadoop installation from the command line, the interactive Hortonworks Sandbox interface isn't available in Windows. Would the more interactive Sandbox environment running directly on Windows have been a better version of Hadoop to build for Windows? If Windows is anything, it's interactive!

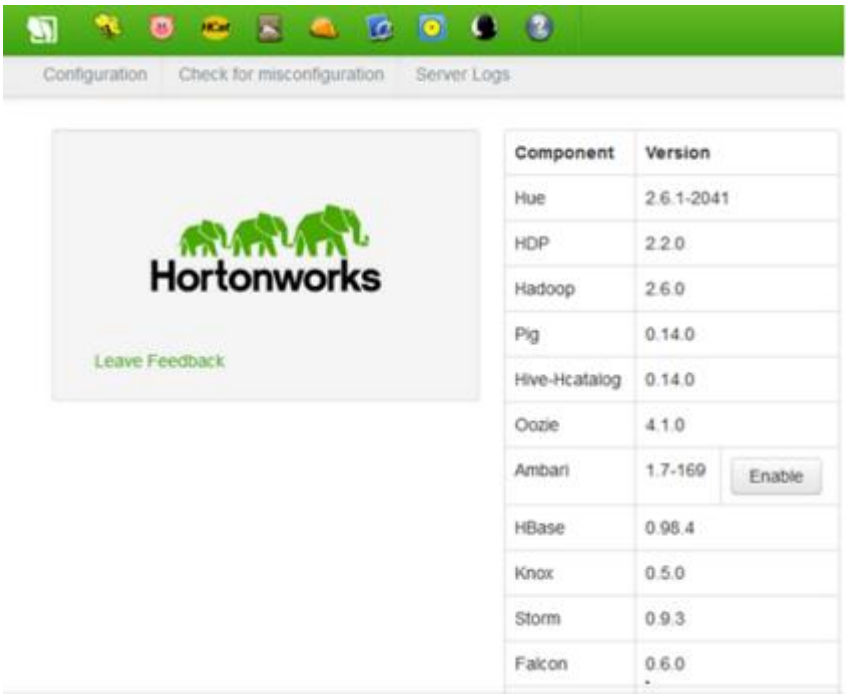


Figure 7: Hortonworks Sandbox, only on Windows via VM

Syncfusion Big Data Platform

Where Hortonworks tried to produce on premises Hadoop for Windows, Syncfusion succeeded. The Syncfusion Big Data Platform runs perfectly on Windows and can be installed in much the same as any other Windows software. It's slightly more complex for multi-node installation, but that is partly the nature of Hadoop. Importantly, with each new release of the platform, there has been significant progress.

Some users of the Hortonworks HDP platform raised issues, such as having to set up nodes manually. With Syncfusion you don't have to—it's made adding nodes and creating clusters straightforward, while adding seamless integration with Active Directory. This gives the user a choice of on-and-off-premises deployment options, and as an offline installer for Hadoop, Syncfusion Big Data Platform is unsurpassed. The online installations from major Hadoop vendors for Linux are impressive, but installing offline isn't always so easy. Syncfusion Big Data platform puts you in control of where, when, and how you want to install Hadoop. It's interesting that the best offline Hadoop installer on any platform is for a Windows platform.

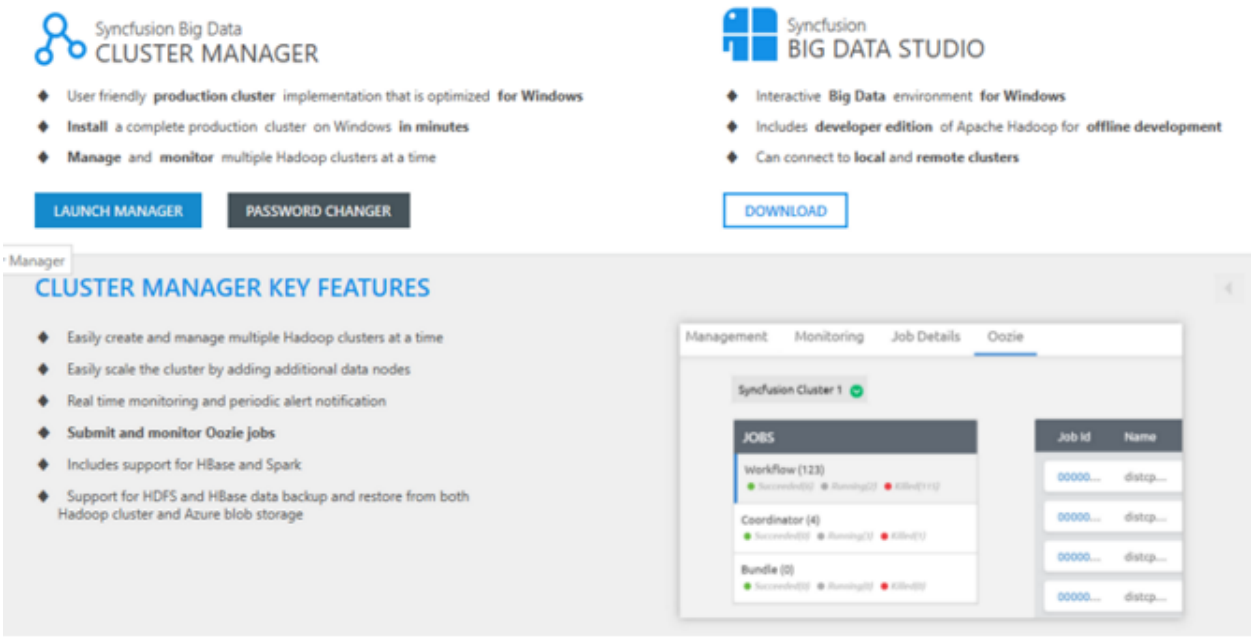


Figure 8: Syncfusion Big Data Platform

www.dbooks.org

Chapter 1 Installing Hadoop for Windows

Choosing a Hadoop distribution to install

In the first chapter I will do an installation of Apache Hadoop for Windows. This is a version of Hadoop that is totally free, and is the basis of all Hadoop distributions. To test whether Hadoop can be installed on pretty much any Windows PC, I will do a local installation on Windows 8. While the screenshots I show are from Windows 8, the process on Windows 10 and Windows Server is similar, and you shouldn't have any trouble making the necessary adjustments. It's important that you don't feel you need the latest shiny, new PC to run Hadoop, though later when we look at multi-node installations, we will use multiple Windows Server 2016 machines. The PC in this exercise meets the requirements shown in Table 1, with 8 GB of RAM, a Quad Core 2.4 GHz AMD processor, and solid-state drives.

Apache Hadoop installation prerequisites

- **JAVA 1.6 or later**: You can download the 64-bit Windows .jdk file **jdk-8u191-windows-x64** from here. It is important to state why a prerequisite is required, so the nature of the dependency on the prerequisite is understood. Hadoop is a Java-based application that creates various dependencies on Java. For example, in a single-node Hadoop installation, there is a single Java process running all Hadoop functions. Without Java, all those functions would be unavailable. It is essential to have the right version and architecture of Java, and a 64-bit JDK higher than 1.6 should always be chosen to install Hadoop for Windows.

- **Hadoop 2.0 or later**: You can download the Hadoop binary file **hadoop-2.9.2.tar.gz** from here.

- **Microsoft Windows**: Windows 7, 8, 10, and Windows Server 2008 and above.

- **Additional prerequisites**: You'll also need a text editor, such as Notepad or Notepad ++, for writing short amounts of code, and Winutils 3.1, which you can download from GitHub.

Java installation for Hadoop for Windows

Run the downloaded Java installer, following the onscreen instructions to complete the installation. Ensure that you right-click on the Java installation file and choose **Run as administrator** from the menu. You will see a User Account Control message asking you to allow the application to make changes to your computer, to which you answer **Yes**. Follow the onscreen prompts to install Java, including the following screen, where you can accept the default installation path.

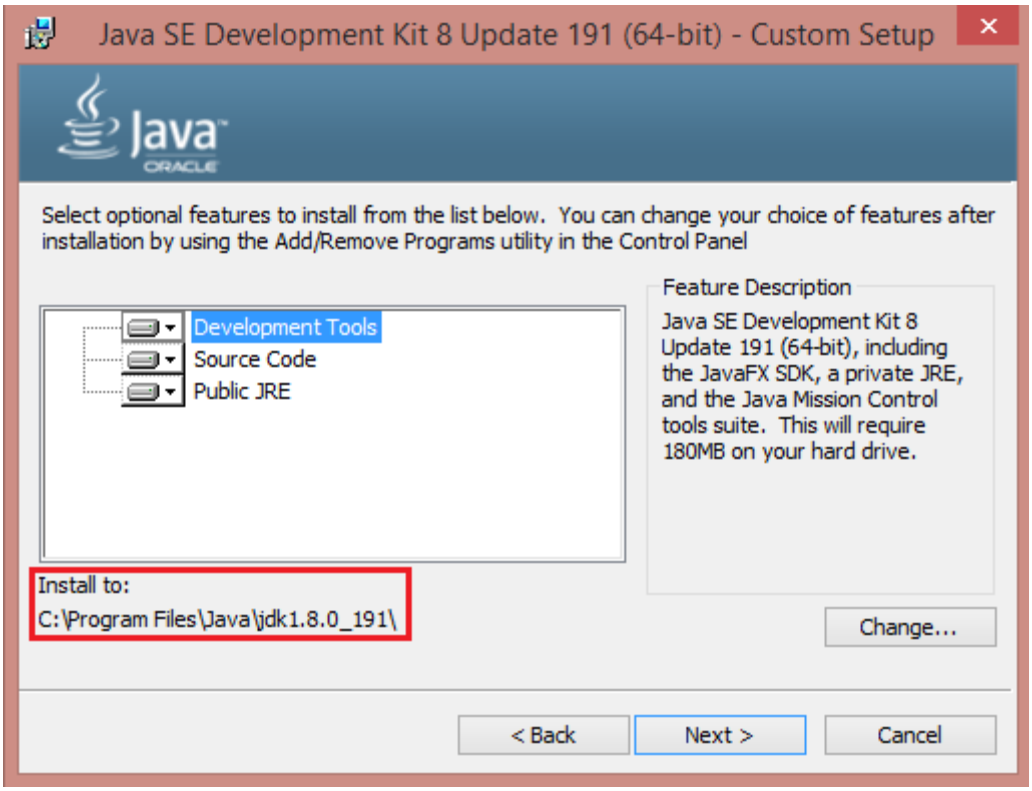


Figure 9: Default Java installation path



Figure 10: Successful Java installation

www.dbooks.org

Ensure that you see the screen informing you that Java has been successfully installed.

Go to **Control Panel** > **System and Security** > **System** and click **Advanced System Settings**, and then click the **System Environment Variables** button. Whether creating a new environment variable for **JAVA_HOME** or editing an existing one, you must alter the Program Files text to text that Hadoop can interpret. On Windows 8, to create a Hadoop-compatible **JAVA_HOME** file instead of entering Program Files, insert **Progra~1** when entering the Java location in the **Variable value** field. On Windows 10 and Windows Server, avoid folder names with blank spaces.

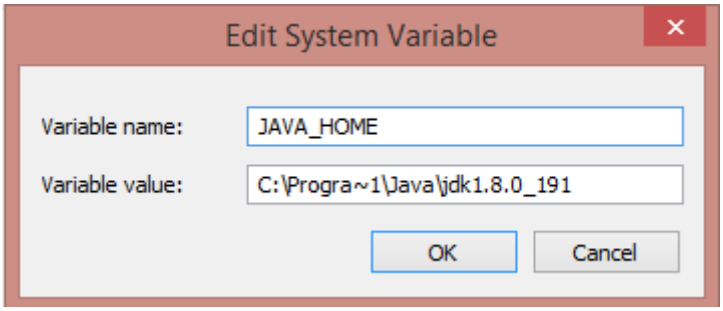


Figure 11: Hadoop compatible Java Home

Please ensure that you add the **JAVA_HOME** to the **Path** variable in System Variables. In this instance, it is done by adding **%JAVA_HOME%\bin** between semi colons in the Path **Variable value** field. Use the **java -version** command from a command prompt to ensure that Java is installed and running correctly.

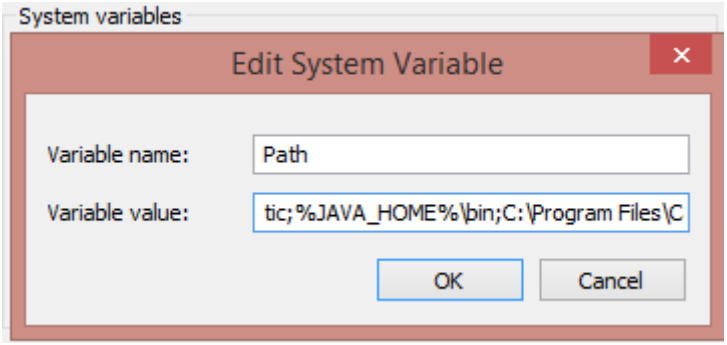


Figure 12: Adding Java Home to the Path Variable

Apache Hadoop installation

1. Create a folder called **C:\hadoop** on your hard drive.

2. Using an application such as 7-Zip File Manager, extract the Hadoop binary file **hadoop-2.9.2.tar.gz** from this website to a directory of your choice, or directly to **C:\hadoop\hadoop-2.9.2**. If you choose to extract the files to a directory of your choice, then you first have to copy the extracted files to **C:\hadoop**. You may find it more convenient to extract them directly to **C:\hadoop**, which will then have an extracted folder in it called hadoop-2.9.2, so you end up with the C:\hadoop\hadoop-2.9.2 folders.

3. You can now create a **HADOOP_HOME** similar to how we created one previously, by going back to **Control Panel** > **System and Security** > **System**, clicking **Advanced System Settings**, and then clicking the **Environment Variables** button. Create the Hadoop home by adding the system variable name **HADOOP_HOME**, with the system variable value being the folder that we extracted the Hadoop binary to, which was **C:\hadoop\hadoop-2.9.2**.

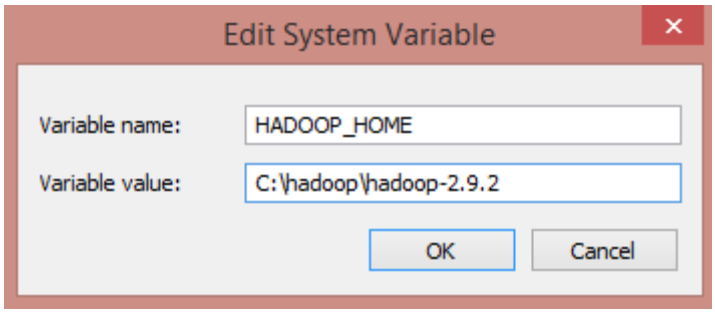


Figure 13: Creating a Hadoop Home

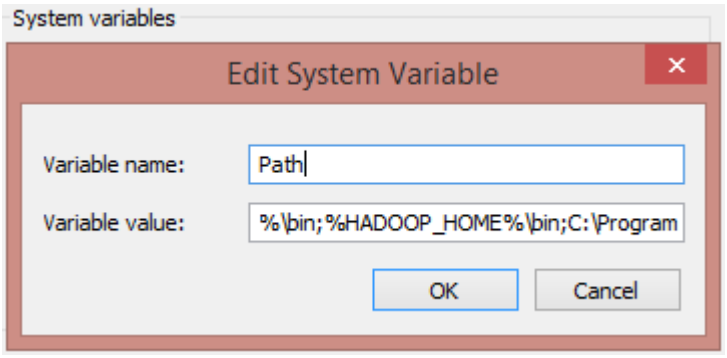


Figure 14: Adding Hadoop Home to the Path variable

We must add the **HADOOP_HOME** file to the **Path** variable in System variables. In this instance, it is done by adding **%HADOOP_HOME%\bin** between semi colons in the **Variable value** field.

In addition, we must add a second **HADOOP_HOME** to the **Path** variable for the folder in Hadoop called **sbin**. This is done by adding **%HADOOP_HOME%\sbin** between semi colons in the **Variable value** field. You should now have Hadoop and Java homes, and two Hadoop path variables.

www.dbooks.org

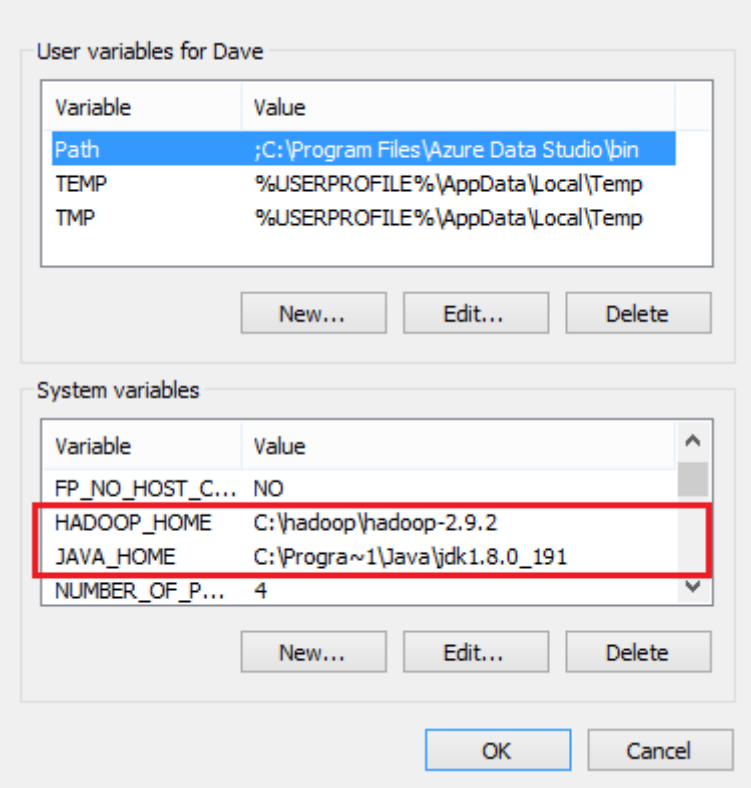


Figure 15: Java and Hadoop homes

The resource page I mentioned previously is an official Apache resource that will assist us in finishing the installation. The area of the site we need first is "Section 3.1. Example HDFS Configuration," which states:

> "Before you can start the Hadoop Daemons you will need to make a few edits to configuration files. The configuration file templates will all be found in c:\deploy\etc\hadoop, assuming your installation directory is c:\deploy."

Since our installation is at C:\hadoop\hadoop-2.9.2, our configuration file templates will be located at C:\hadoop\hadoop-2.9.2\etc\hadoop\. The first file we need to edit is the **core-site.xml** file. The following code listing shows the format of the core-site.xml file, which is the style that we need to adapt. You will need your code editor at this point (I am using Notepad++).

Code Listing 1: The core-site.xml file format

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://0.0.0.0:19000</value>
</property>
</configuration>
```

We need to substitute the name and value elements shown on the core-site.xml file on the Apache Wiki page for values in the installation we are carrying out. The values we require are contained in the following code listing and reflect our current Hadoop installation.

Code Listing 2: Editing the core-site.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl"href="configuration.xsl"?>

<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

We need to do the same for the **hdfs-site.xml** file template, and the new values we require are in the following code listing.

Code Listing 3: Editing the hdfs-site.xml template

```
<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///C:/Hadoop/hadoop-2.9.2/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>file:///C:/Hadoop/hadoop-2.9.2/datanode</value>
</property>

</configuration>
```

Note: You must create two folders in the C:\Hadoop\hadoop-2.9.2\ folder in Windows Explorer to reflect the *namenode* ***and*** *datanode* ***directories mentioned in***

www.dbooks.org

Code Listing 3. Note that the Hadoop configuration files use forward slashes instead of backward slashes in file paths, even on Windows systems.

Next, we need to edit the **mapred-site.xml** configuration file**;** the values required are shown in the following code listing.

Code Listing 4: Editing the mapred-site.xml configuration file

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>

</configuration>
```

We also need to edit the **yarn-site.xml** configuration file; the values required are provided in the following code listing.

Code Listing 5: Editing the yarn-site.xml configuration file

```
<configuration>


<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.auxservices.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>
```

Next, follow these steps:

1. Replace the bin folder at **C:\hadoop\hadoop-2.9.2\bin** with a bin folder extracted from here.

2. Extract the bin folder from the **apache-hadoop-3.1.0-winutils-master** file downloaded from here.

3. Make a copy of the bin folder at **C:\hadoop\hadoop-2.9.2\bin**, and then delete the folder you made the copy from.

4. Copy the bin folder you extracted from the **apache-hadoop-3.1.0-winutils-master** file to **C:\hadoop\hadoop-2.9.2**; it replaces the bin folder you deleted.

Now we must follow the instructions in section 3.4 of the Hadoop Wiki page, "3.4. Format the FileSystem." This is done by executing the following command (with administrator privileges) from a command shell:

Code Listing 6: Format of the Filesystem

```
hdfs namenode -format
```

You should now see the following on your screen.

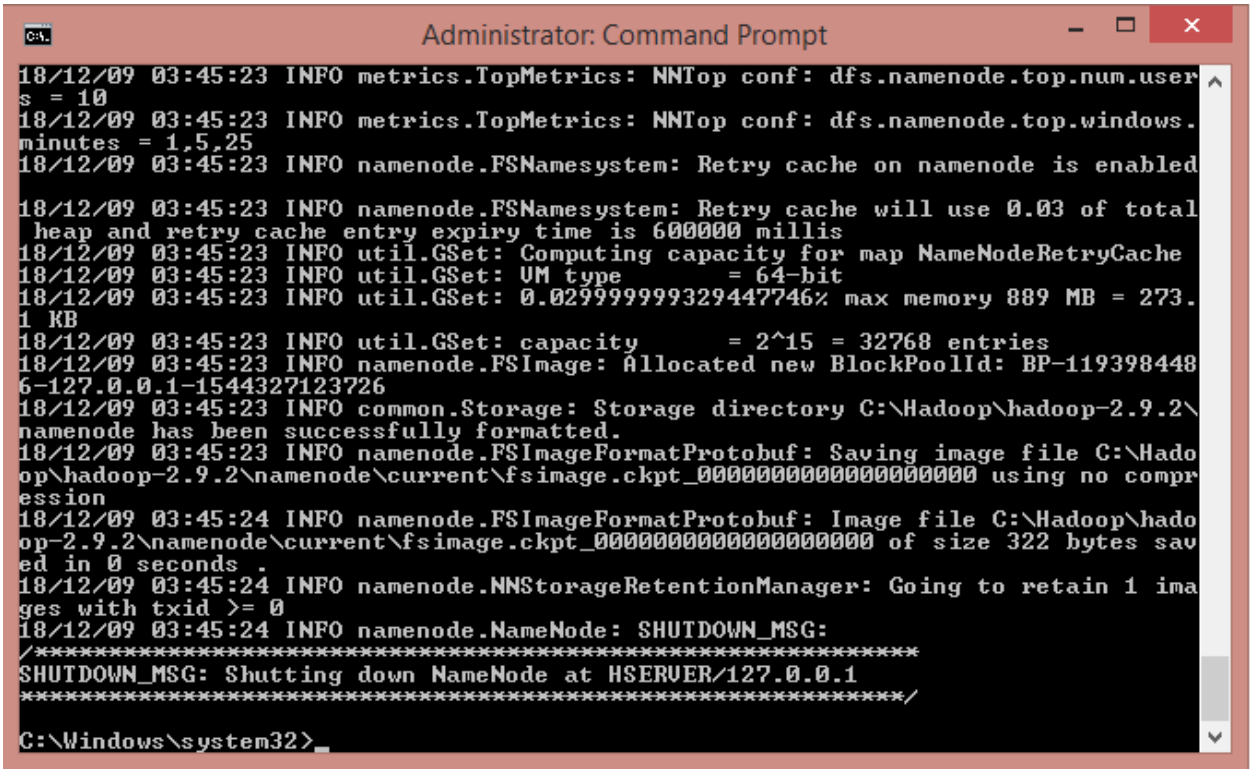


Figure 16: Successful formatting of the Filesystem

You must now copy the **hadoop-yarn-server-timelineservice-2.9.2** file from **C:\hadoop\hadoop-2.9.2\share\hadoop\yarn\timelineservice** to the folder **C:\hadoop\hadoop-2.9.2\share\hadoop\yarn**. We can start Hadoop with the instructions in sections 3.5 and 3.6. of the Hadoop Wiki page, called "3.5. Start HDFS Daemons" and "3.6. Start YARN Daemons and run a YARN job."

You start HDFS daemons by running the following code from the command prompt.

Code Listing 7: Start HDFS Daemons command

```
start-dfs.cmd
```

You start YARN daemons and run a YARN job by running the following code.

www.dbooks.org

Code Listing 8: Start YARN daemons and run a YARN job command

```
start-yarn.cmd
```

You should now see the Hadoop **namenode** and **datanode** successfully started.

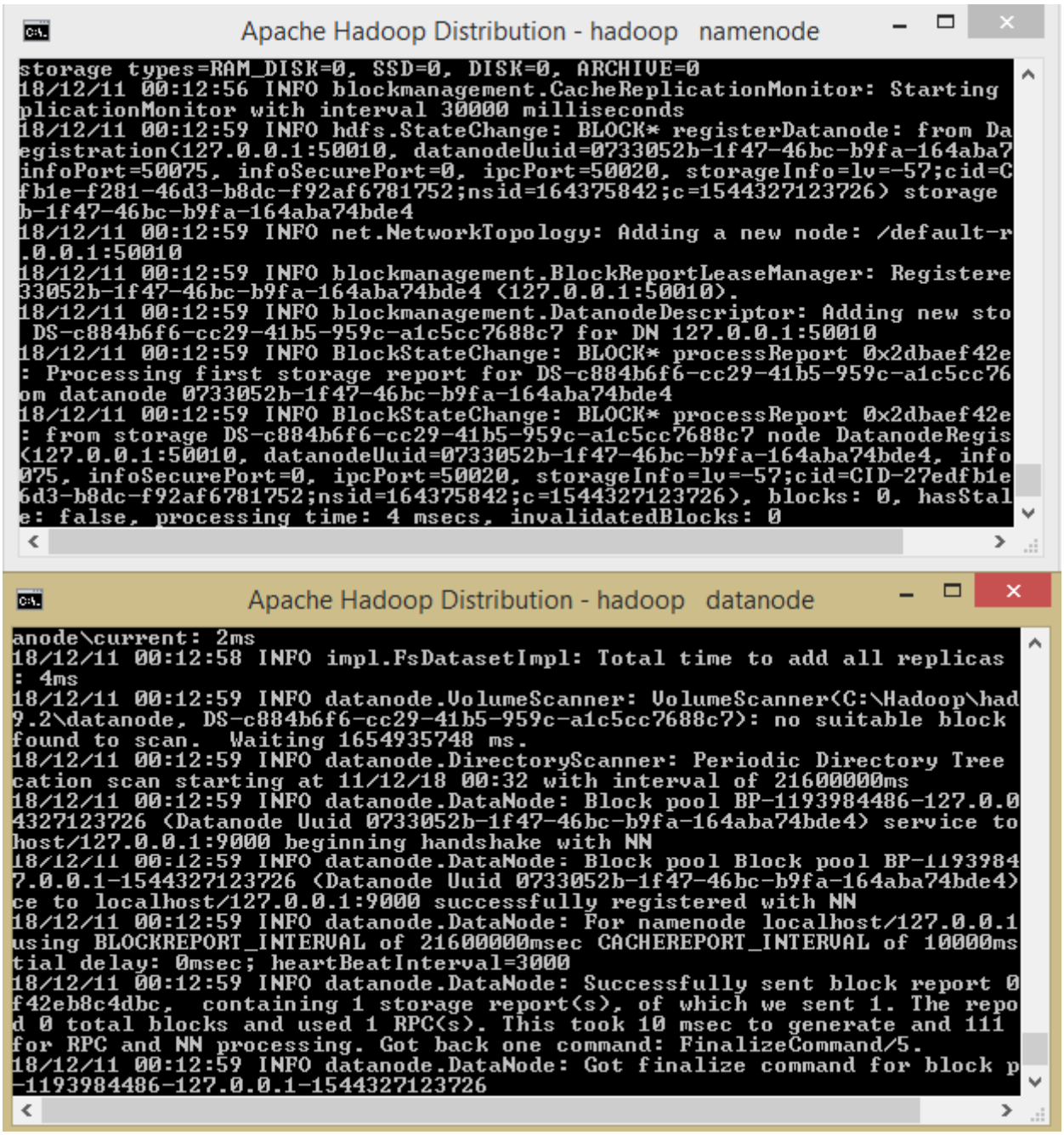


Figure 17: Hadoop namenode and datanode started successfully

In addition, you will see the YARN **resourcemanager** and YARN **nodemanager** successfully started.

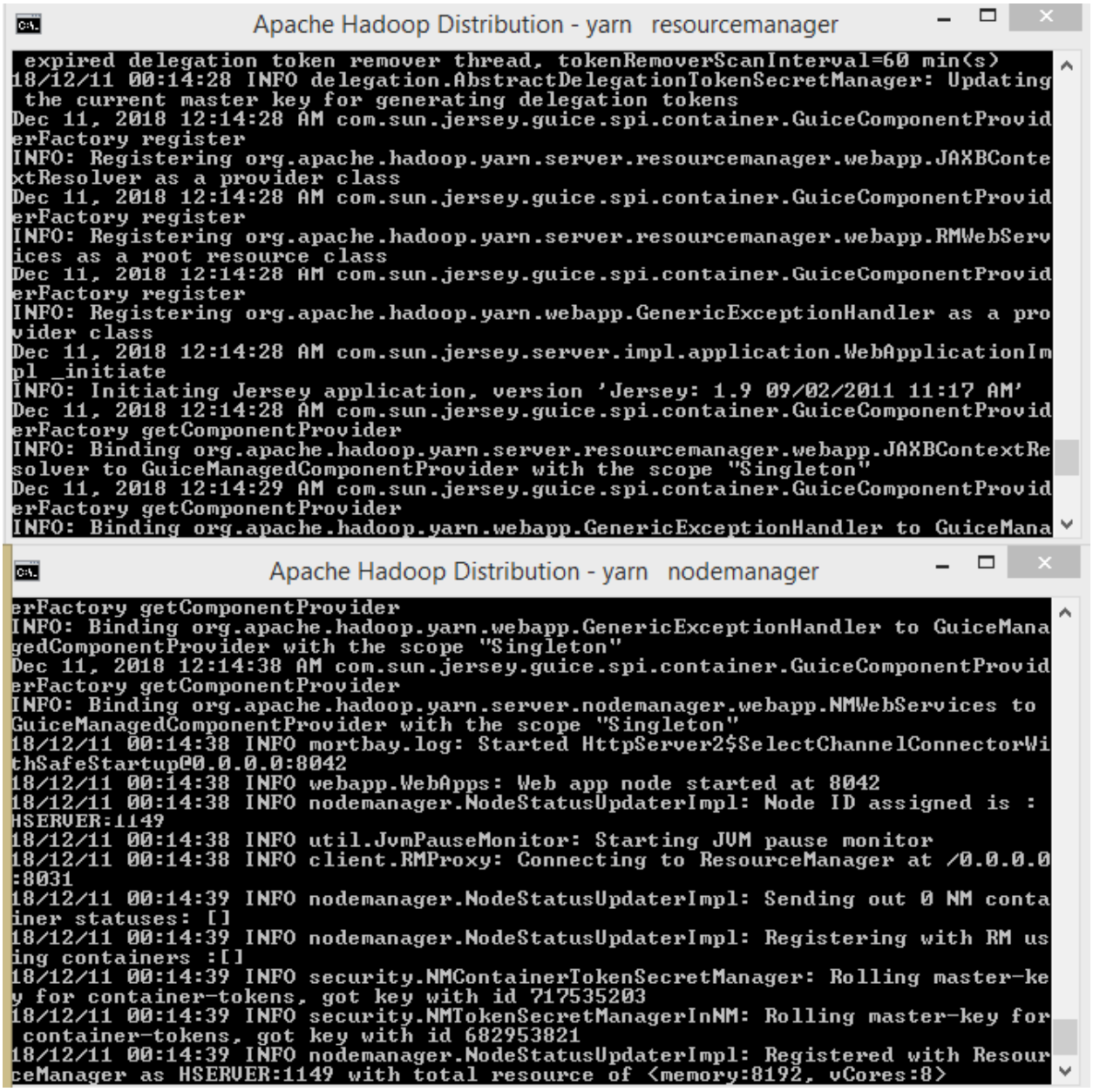


Figure 18: Yarn resourcemanager and Yarn nodemanager successfully started

www.dbooks.org

The finished Hadoop installation directory is shown in the following image. I have added a folder called **datastore**, into which I have placed two text files called ukhousetransactions.txt and ukhousetransactions2.txt.

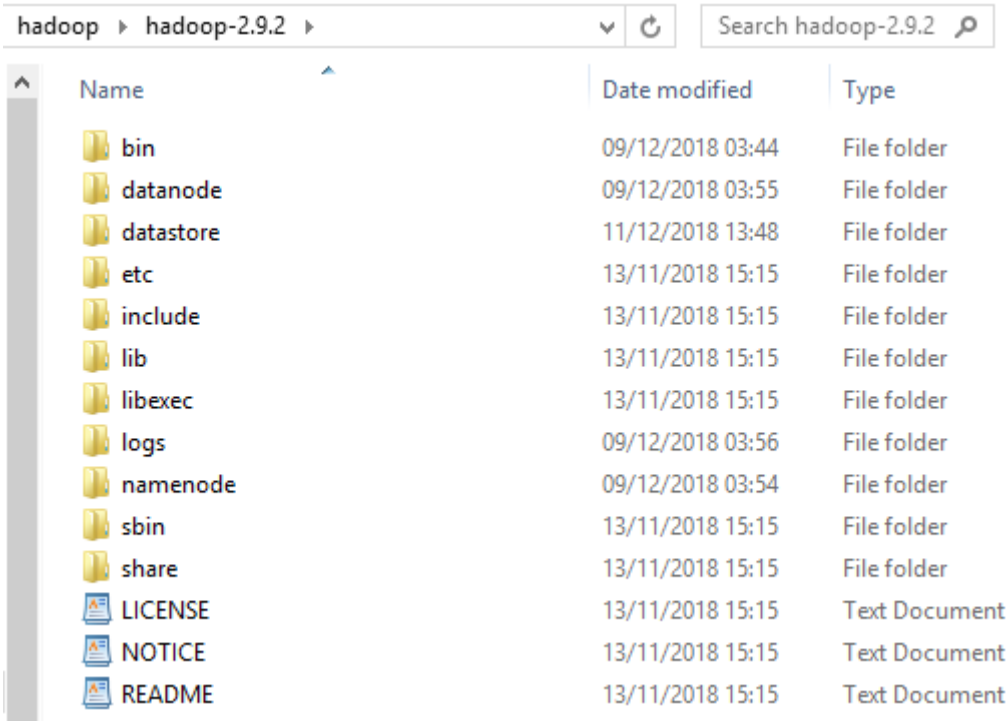


Figure 19: Final Hadoop installation directory

Let's create a directory called **bigdata** by running the following code.

Code Listing 9: Creation of directory called bigdata

```
hadoop fs -mkdir /bigdata
```

The next section of code, **`hadoop fs -ls /`**, confirms the directory has been created by listing its contents.

Code Listing 10: Listing of Directory Contents

```
C:\Windows\system32>hadoop fs -ls /
Found 1 items
drwxr-xr-x   - Dave supergroup          0 2018-12-11 13:52 /bigdata
```

We can copy the **ukhousetransactions.txt** file and the **ukhousetransactions2.txt** to the HDFS after first changing directory to the **datastore** folder, by using the change directory command: **`cd C:\hadoop\hadoop-2.9.2\datastore`**. Both tables have the same data and are used to check that duplicate records can be identified and removed by any tool that accesses them.

Try to remember the code you used for the first file to copy the second file, **ukhousetransactions2.txt,** from memory. If you're new to Hadoop, you'll get used to using the command line much quicker if you remember the basic commands.

Code Listing 11: Copying Text Files to the Hadoop Distributed File System (HDFS)

```
hadoop fs -put ukhousetransactions.txt /bigdata
hadoop fs -put ukhousetransactions2.txt /bigdata
```

We can now list the files we have copied with the following command.

Code Listing 12: Listing of copied files to HDFS

```
C:\hadoop\hadoop-2.9.2\datastore>hadoop fs -ls -R /
```

We access our Hadoop installation by going to http://localhost:8088/ or http://localhost:50070.

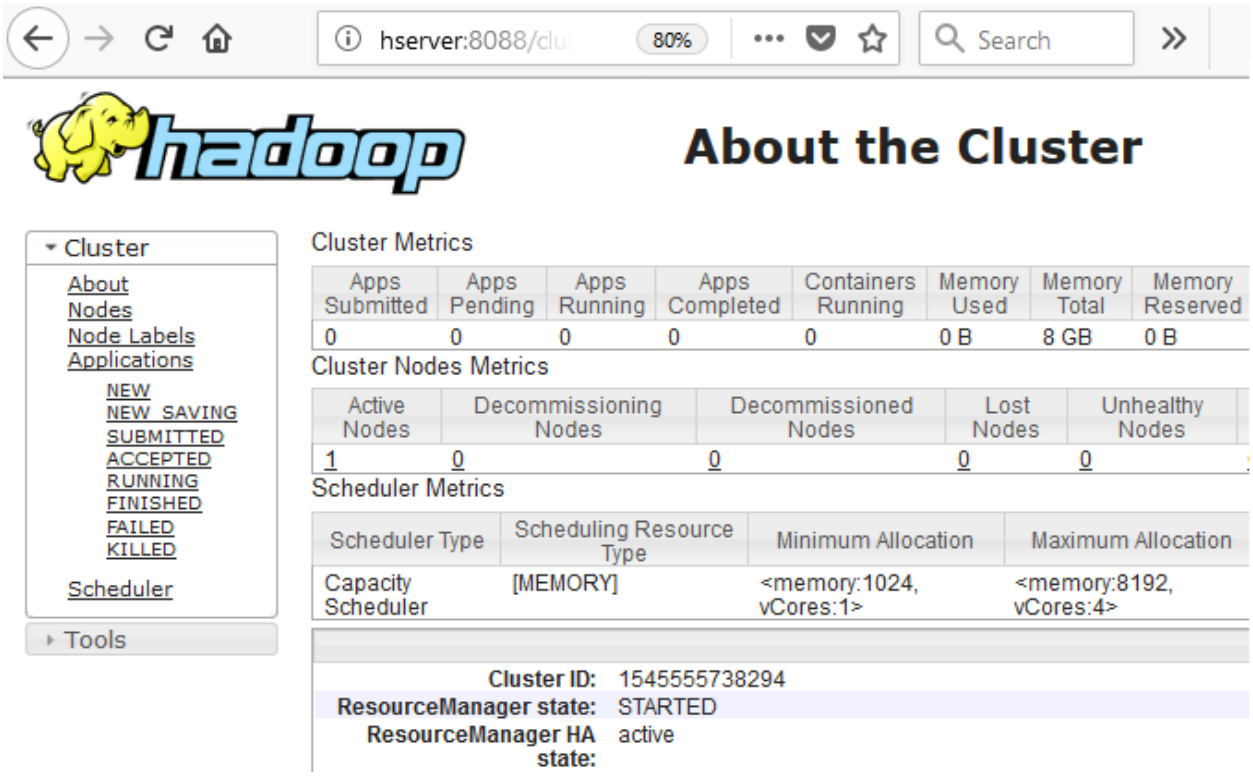


Figure 20: Apache Hadoop is now running on Microsoft Windows

Connecting to data sources within HDFS

Earlier, in Figure 2, I mentioned the HDFS connector for Microsoft Power BI.

www.dbooks.org

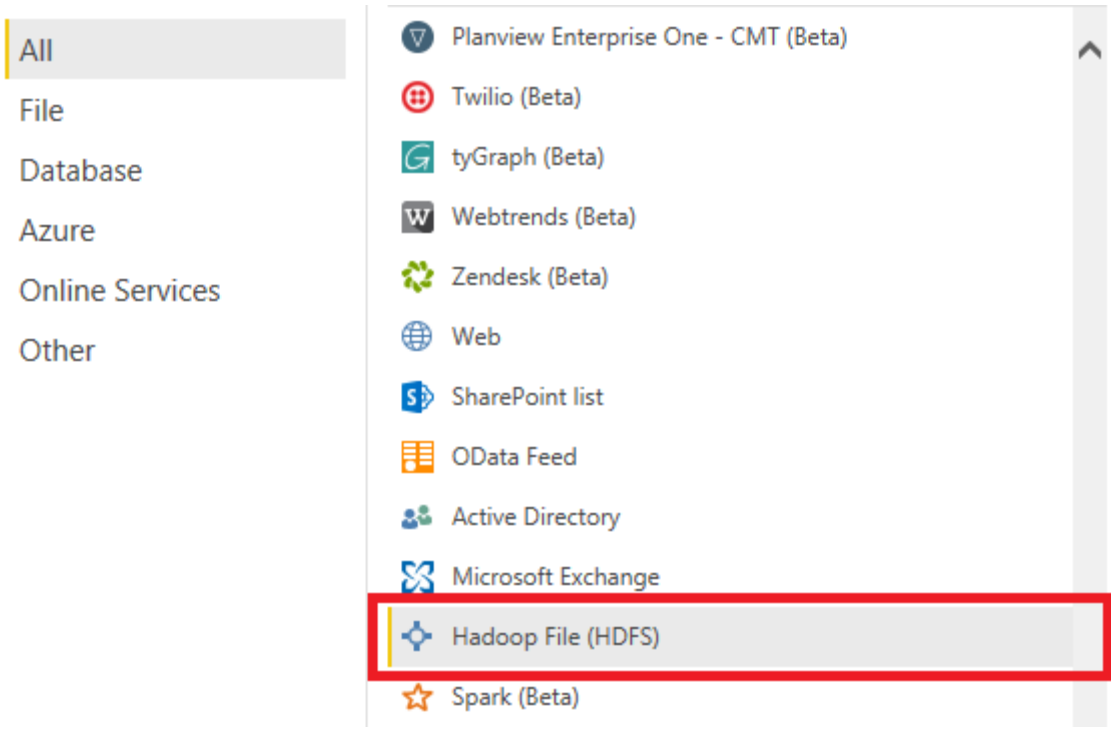


Figure 21: HDFS connector for Power BI

Now that we have Hadoop installed, we can revisit this by selecting the **Hadoop File (HDFS)** connection in Power BI and clicking **Connect**. The aim is to connect to HDFS without a Hive ODBC driver.

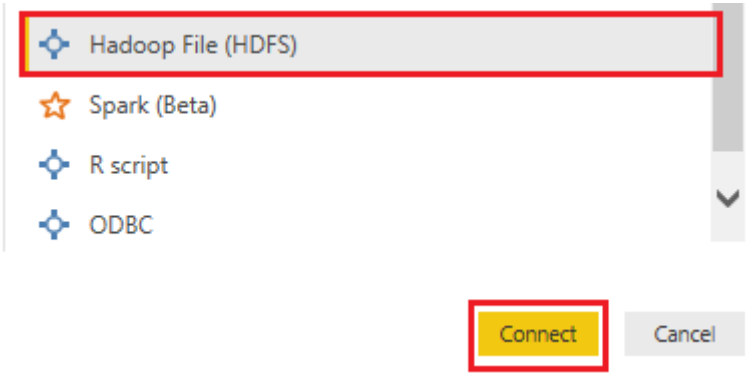


Figure 22: Connecting to HDFS from Power BI

Enter **localhost** as the name of the server that HDFS is installed on, then click **OK**.

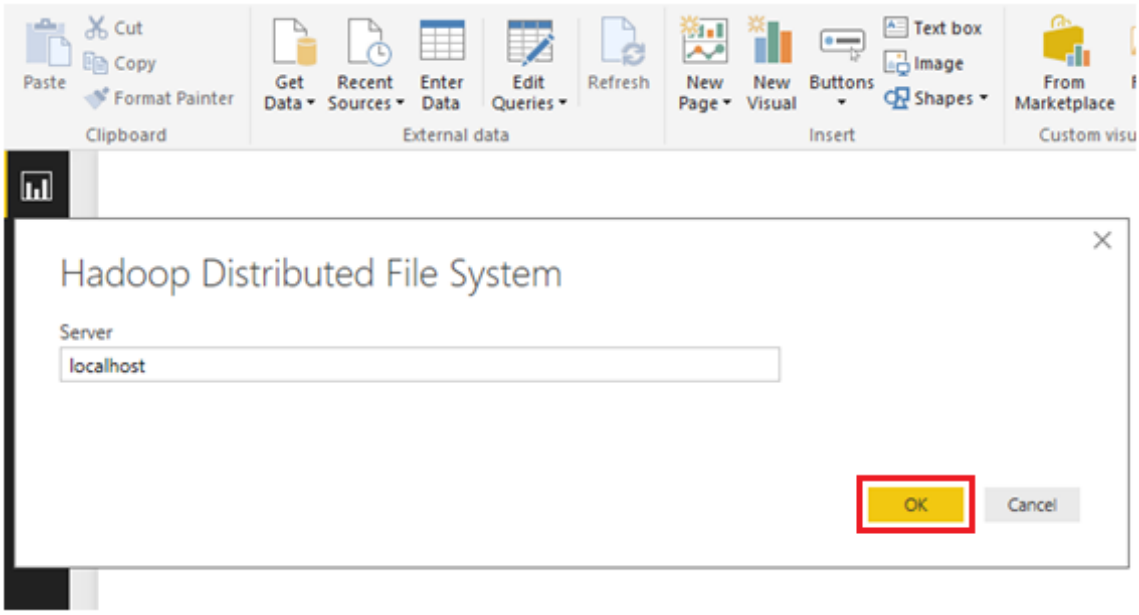


Figure 23: Connecting to HDFS from Power BI in Windows

The first time you get past the preceding screen, you may see a screen similar to the one shown in Figure 24, asking for your preferred method of security access. This can also happen if you don't type in **localhost** on the preceding screen, but instead, enter **localhost:9000**, for example. You simply need to enter **localhost** then click **OK**, as shown in the previous screen. If you see a screen similar to Figure 24 in response to entering **localhost**, click **Connect**.

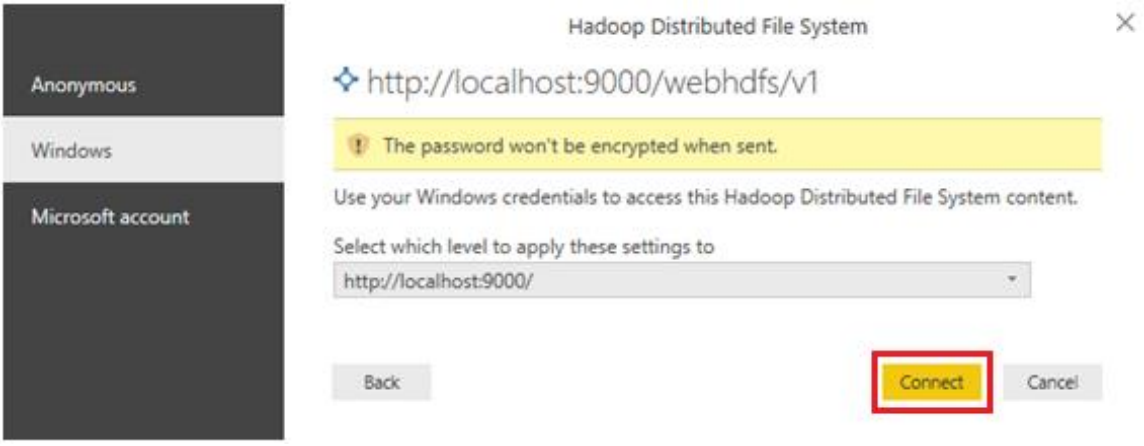


Figure 24: Security Access Options for HDFS on Power BI

The files we copied to HDFS are now accessible in Power BI. Click **Load** to continue.

www.dbooks.org

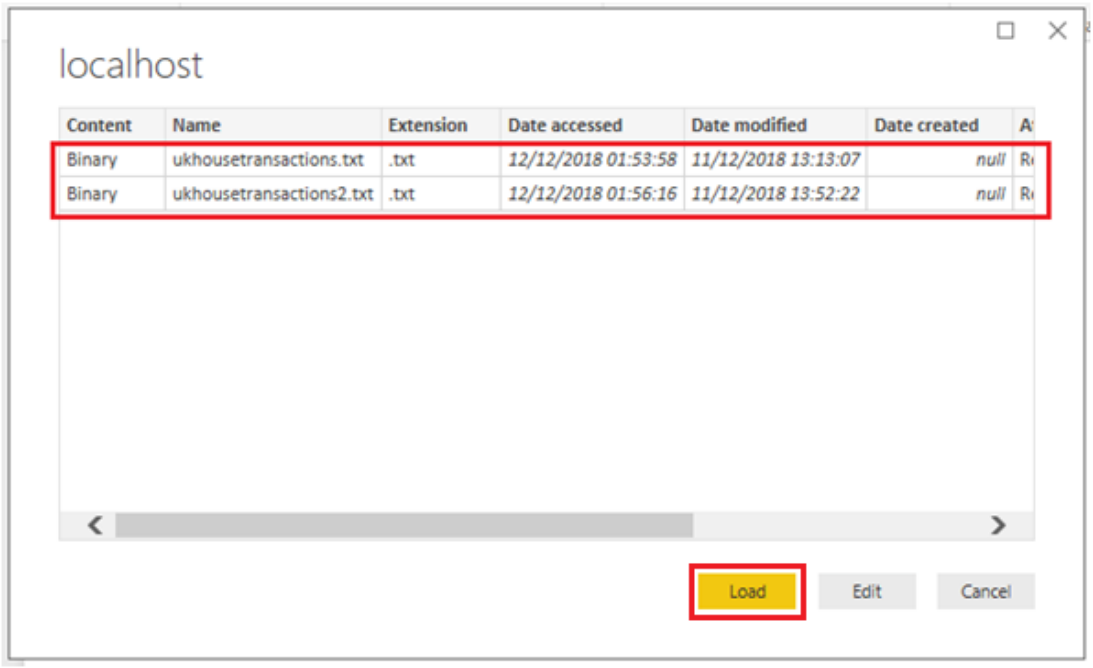


Figure 25: Loading files from HDFS into Power BI without Hive ODBC

Now all the tools in Power BI can be used on the data and queries run against it.

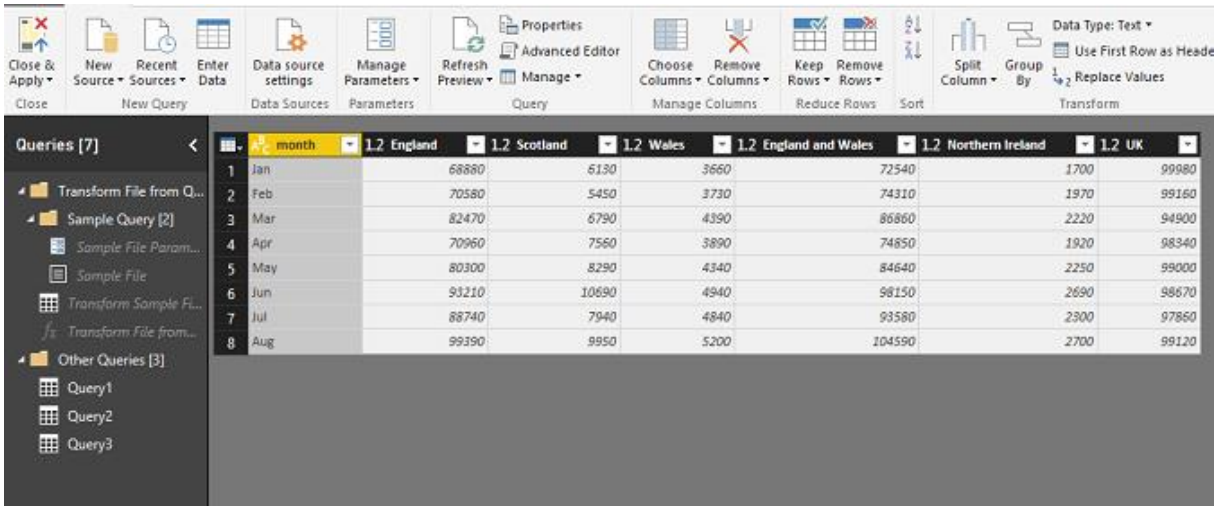
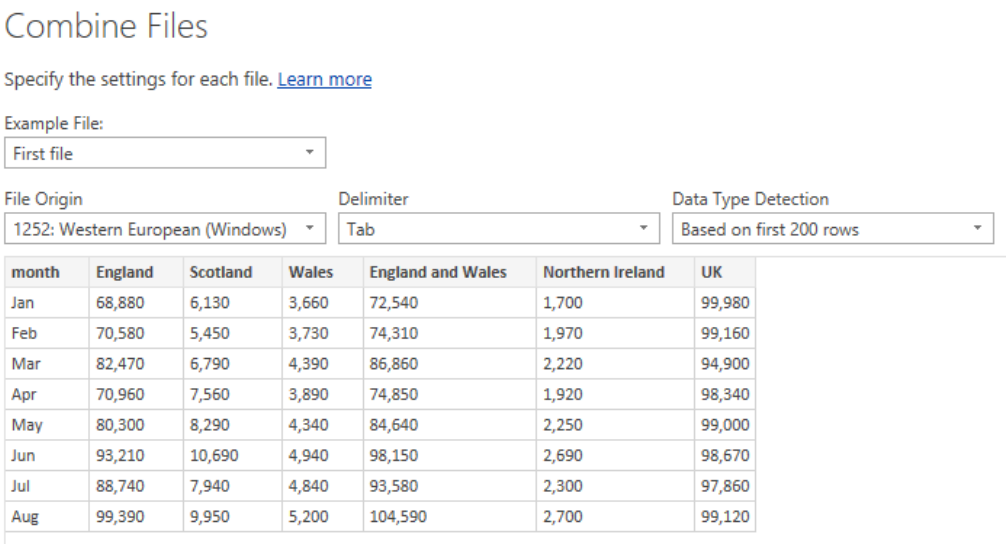


Figure 26: HDFS Data in the Power BI Environment

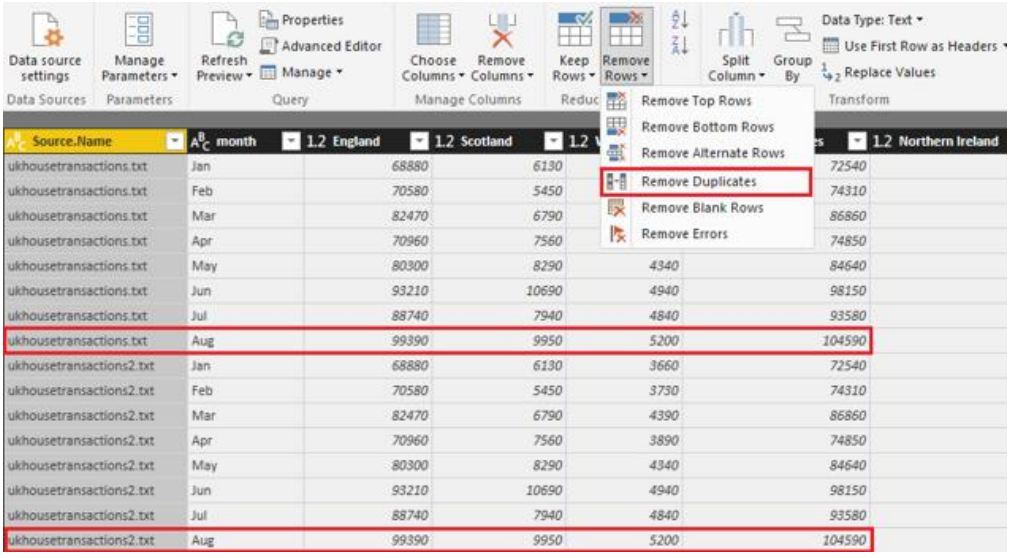
In addition, we have the ability to combine files in Power BI for multiple-file loading from HDFS.



month	England	Scotland	Wales	England and Wales	Northern Ireland	UK
Jan	68,880	6,130	3,660	72,540	1,700	99,980
Feb	70,580	5,450	3,730	74,310	1,970	99,160
Mar	82,470	6,790	4,390	86,860	2,220	94,900
Apr	70,960	7,560	3,890	74,850	1,920	98,340
May	80,300	8,290	4,340	84,640	2,250	99,000
Jun	93,210	10,690	4,940	98,150	2,690	98,670
Jul	88,740	7,940	4,840	93,580	2,300	97,860
Aug	99,390	9,950	5,200	104,590	2,700	99,120

Figure 27: Combining multiple files from HDFS in Power BI

We also have the ability to remove duplicates from the UK house transactions files we loaded into HDFS.



Source.Name	month	England	Scotland		Northern Ireland
ukhousetransactions.txt	Jan	68880	6130		72540
ukhousetransactions.txt	Feb	70580	5450		74310
ukhousetransactions.txt	Mar	82470	6790		86860
ukhousetransactions.txt	Apr	70960	7560		74850
ukhousetransactions.txt	May	80300	8290	4340	84640
ukhousetransactions.txt	Jun	93210	10690	4940	98150
ukhousetransactions.txt	Jul	88740	7940	4840	93580
ukhousetransactions.txt	Aug	99390	9950	5200	104590
ukhousetransactions2.txt	Jan	68880	6130	3660	72540
ukhousetransactions2.txt	Feb	70580	5450	3730	74310
ukhousetransactions2.txt	Mar	82470	6790	4390	86860
ukhousetransactions2.txt	Apr	70960	7560	3890	74850
ukhousetransactions2.txt	May	80300	8290	4340	84640
ukhousetransactions2.txt	Jun	93210	10690	4940	98150
ukhousetransactions2.txt	Jul	88740	7940	4840	93580
ukhousetransactions2.txt	Aug	99390	9950	5200	104590

Figure 28: Removing duplicates from the house transactions files in Power BI

Now we can create dashboards from the HDFS data; the integration is such that we can automatically convert numeric text to numeric values. This allows us to ask math-based questions using free text, a function seen previously in the online BI tool IBM Watson Analytics.

www.dbooks.org

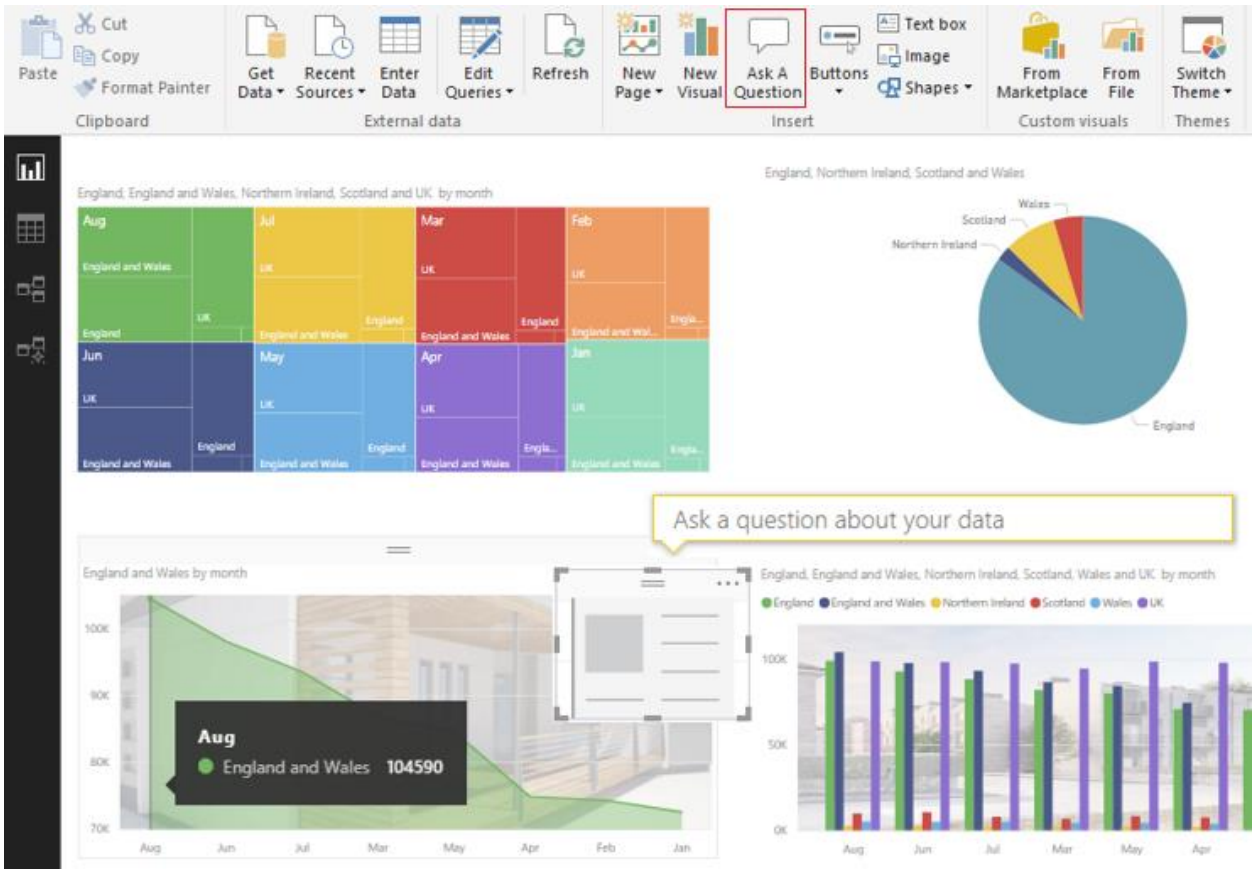


Figure 29: Creating a Dashboard from HDFS data in Microsoft Power BI

I started typing the question "How many England and Wales Aug" and before I could type the full sentence it automatically calculated 105k.This is a rounding up of the correct answer of 104590 as shown next. It automatically offered the option to select data for the other months.

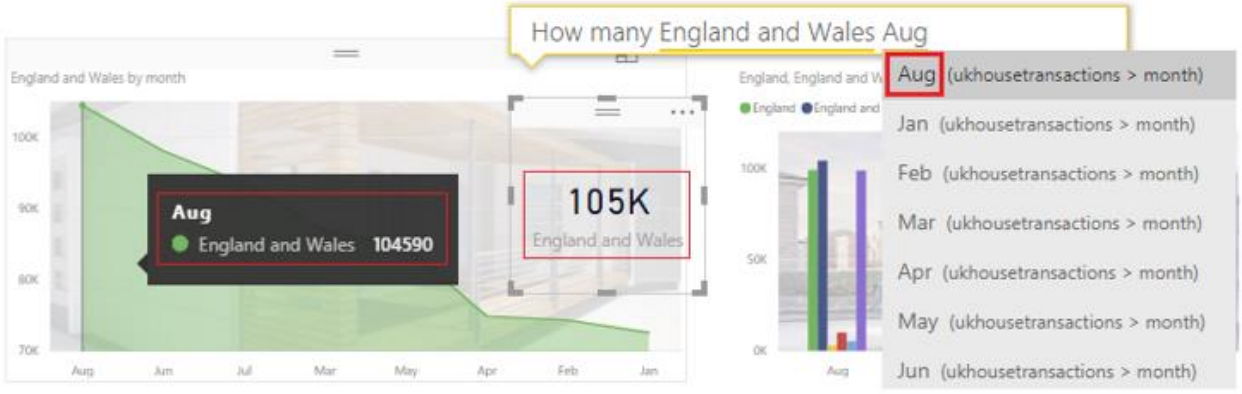


Figure 30: Power BI calculating a math answer from a free text question

Everything we required to do this exercise runs in the same Windows environment. As good as this is, how would you recommend it to peers? HDInsight is only online for Linux, and HDP is archived or a VM, leaving manual Apache Hadoop or Syncfusion Big Data Platform.

Summary

Despite learning about the Linux dominance of Hadoop and questions surrounding the ability of Windows to run Hadoop, we were able to install Apache Hadoop 2.9.2 on Windows 8 quite easily. We did this with no changes to Windows, other than installing the required software. We then created a directory within HDFS to store files and accessed them with ease from Power BI.

This was an advantage over having to access Hadoop on Linux from an external Windows system running BI tools. Using the command prompt in Windows felt no different than using Hadoop in Linux. The memory management was good, and seldom went above 4 GB of RAM; there was no deterioration of the system performance at any time. You can freely download Hadoop and a free version of Power BI, and you probably already have Windows. This is the strength of Hadoop for Windows: the lack of disruption. Everything else you need already runs on Windows. It's all already there, so it makes sense to invite Hadoop into Windows rather than move to Linux.

www.dbooks.org

Chapter 2 Enterprise Hadoop for Windows

Physical and virtual enterprise Hadoop distributions for Windows

Virtual machines have allowed big-data vendors from Cloudera to MapR to package Hadoop distributions that run on Windows. Generally, these distributions are fine for testing or development, but less suited to production environments. While you can load a Linux Virtual Machine that runs Hadoop for Windows, it's still Hadoop running on Linux. What we need is multi-node enterprise Hadoop with Hive, Pig, and Sqoop running within Windows itself.

Three vendors have released software that falls into this category: Apache, Hortonworks, and Syncfusion. We've already done an Apache installation, and Hortonworks is in archive, so Syncfusion and Microsoft remain. Why do I include Microsoft? It's because their enterprise-level changes to Windows Server made multi-node Hadoop installations possible. They have gone even further with their flagship enterprise database product, SQL Server 2019. Though not yet released, the technology preview of SQL Server 2019 has taken Hadoop integration in Windows to new levels. We will look more closely at that product in Chapter 4. All this progression would not have been possible without Apache themselves, as can be seen at Apache.org.

The next two figures highlight this, with the first discussing a patch for running Hadoop for Windows without Cygwin. This was an important milestone in terms of Hadoop being able to run on Windows. On the left-hand side in Figure 31, there are numerous other issues that have been raised and logged.

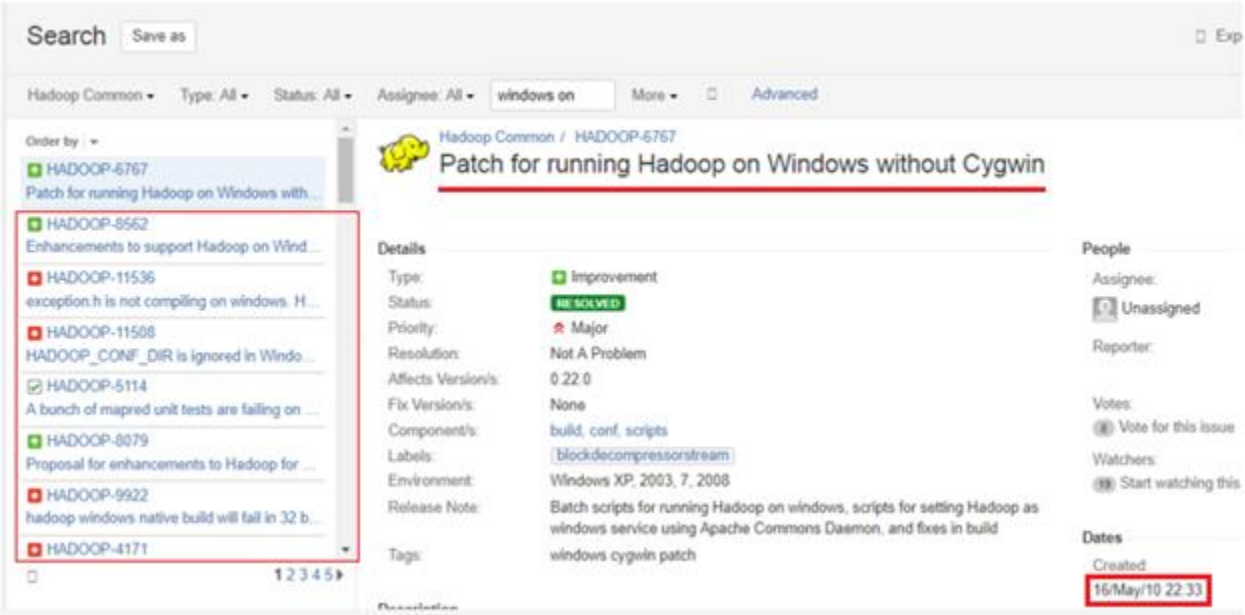


Figure 31: Apache.org shows issues for Hadoop for Windows

Figure 32 shows an enhancement to support Hadoop natively on Windows Server and Windows Azure environments. This highlights an expectation to run Hadoop natively on Windows, the same way Hadoop runs on Linux. Managing user expectations are vital in this area.

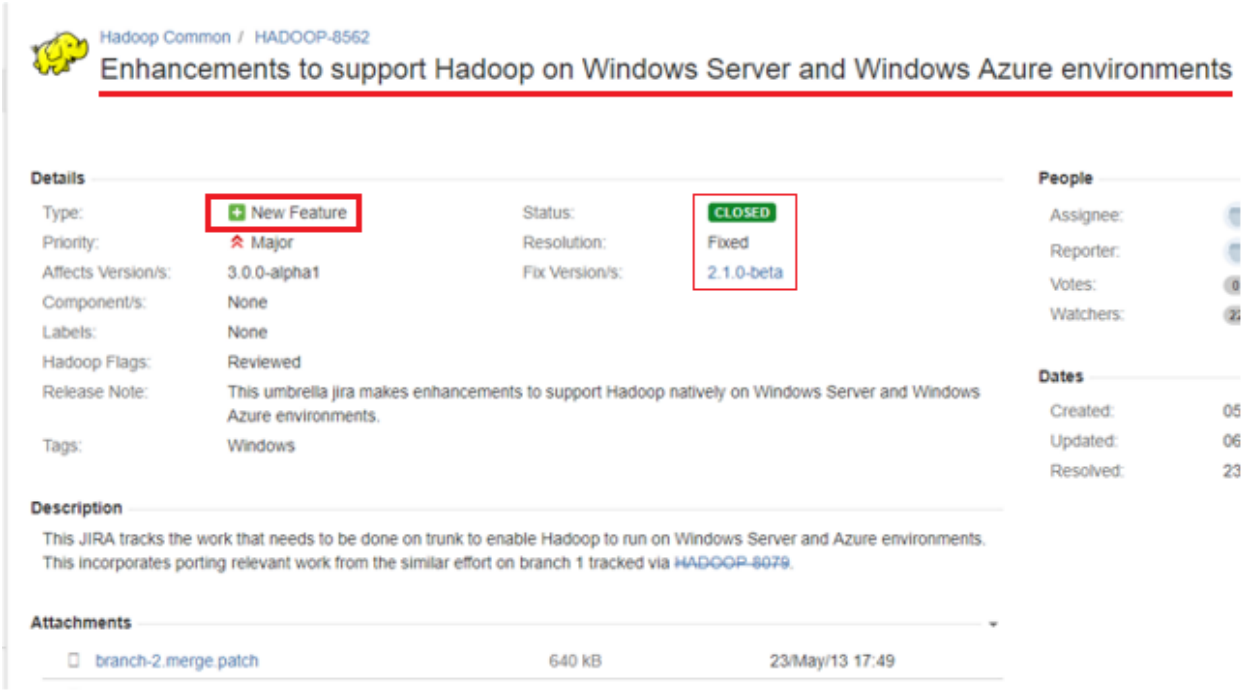


Figure 32: Enhancement to support Hadoop for Windows Server and Windows Azure at Apache.org

We'll be using the Hadoop distribution from Syncfusion, which is available from Syncfusion.com. Before installation, there are a few things we need to be aware of. Windows and Linux are very different environments, and using the same application in either environment will not be the same. If the version of Spark your customer uses is more recent than the one you're demonstrating, be aware of that in advance. If there are features of the latest version of Spark that your client depends on, it may be an issue for you. For these reasons, Hadoop distributors need to update Hadoop ecosystem components within reasonable timescales.

There are key features that Linux developers will expect to see in Hadoop for Windows. Often these issues are sorted out by looking at the feature sets of the Hadoop distribution in question. Impala, for example, is thought to be the fastest SQL-type engine for Hadoop, but it only runs on Linux. A bigger problem is when the issue is not the feature set of the Hadoop distribution, but the operating system itself. In Linux you have control groups (cgroups), which aren't present in Windows Server, nor is there an equivalent. I will discuss cgroups in Chapter 3 in the section about memory management and Hadoop performance in Windows. When you talk to Linux users about potentially using Hadoop in Windows, you should demonstrate awareness of these matters. While Hyper-V and virtual machines can be used to allocate resources in Windows, they're just not the same as cgroups in Linux.

www.dbooks.org

Network setup and installation

Before we set up a production cluster, we need to understand the network we're going to install it on. Sometimes you hear complaints that Hadoop is slow or doesn't meet expectations. Often, it's because the network it's installed on isn't the optimum network for Hadoop. A positive of Microsoft Azure is that Microsoft gives you all the computing power you need to run Hadoop. This enables companies to analyze a hundred terabytes of data or more. If you're fortunate enough to be able to build your own data network, build the fastest network you possibly can. If you have access to physical servers, use those instead of virtual servers—you'll notice the power of a production cluster on a more powerful network.

There is a price premium for these gains, but they can be negated by cost savings per terabyte. The faster a system can analyze data, the less time you spend running the cluster and its associated electricity, CPU, and cooling costs. This is partly how HDInsight works; you pay for what you use, and can provision or decommission clusters when you're not using them. You can do this yourself on-premises, but you'd have setup costs that you don't have on Azure.

Suitable network components

We require our network to run at speeds associated with a production environment. As a minimum, I would recommend a 10 Gbps switch, while the optimum would be 25-to-40 Gbps.



Figure 33: ProSAFE 16 Port 10-Gigabit Managed switch

If you're dealing with hundreds of terabytes, a good strategic investment may be a 10–100 Gbps switch; this gives a wider coverage of network speeds without having to change switches.



Figure 34: Cisco Nexus 7700 Switch - 10, 40, and 100 Gbps

Your server adapter should be of a speed commensurate to that of your network. A 40-Gbps adapter for your server is optimum; your PC's network adapter should be at least 1 Gbps.



Figure 35: QLogic 40 Gbps Ethernet Adapter

The following cables can manage up to 40 Gbps, with the most economical solution to buy bulk Cat 8 cabling and fit the RJ45 plugs.



Figure 36: BAKTOONS Cat 8, 40 Gbps RJ45 (left) and Cat 8 bulk cable 25/40 Gbps (right)

Suitable server hardware and Windows licensing

You can use your own servers that meet the requirements shown in Table 6, for example, two physical servers with five VMs (virtual machines) or five physical servers. Different configurations of the Dell C4130 shown in the next figure meet the requirements at an economical price.



Figure 37: The Dell PowerEdge C4130

Table 6: Server requirements for production Hadoop cluster

Active Namenode Server	Standby Namenode Server	Datanode 1	Datanode 2 (If needed)	Cluster Manager
CPU: 2-4 Octa-core+ 96 GB RAM Hard Drive: 2 × 1 TB	CPU: 2-4 Octa-core+ 96 GB RAM Hard Drive: 2 × 1 TB	CPU: 2-4 Octa-core+ 64 GB RAM	CPU: 2-4 Octa-core+ 64 GB RAM	CPU: 2-4 Octa-core+ 32 GB RAM Hard Drive: 2 × 1 TB

www.dbooks.org

Active Namenode Server	Standby Namenode Server	Datanode 1	Datanode 2 (If needed)	Cluster Manager
Network: 10 Gbps	Network: 10 Gbps	Hard Drive: 4 × 1 TB SAS JBOD(16 × 1 TB) Network: 10 Gbps	Hard Drive: 4 × 1 TB SAS JBOD (16x 1TB) Network: 10 Gbps	Network: 10 Gbps

If you can't access servers with the RAM listed in the preceding table, use nodes with at least 32 GB of RAM. You won't be able to handle very large amounts of data, but it will certainly work.

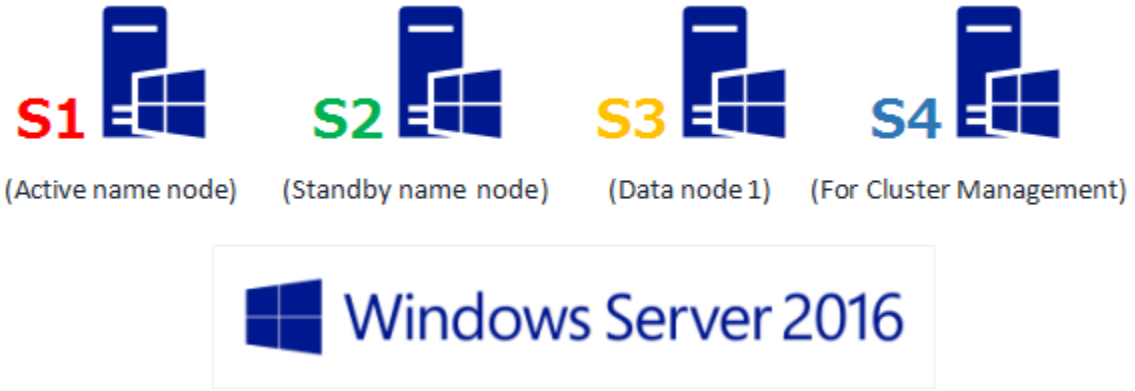


Figure 38: Server roles in our Windows cluster

We now have the server roles in our cluster defined; I'd recommend four physical servers over virtual ones. The pricing for Windows Server is listed, but if you already have Windows Server licenses, you can use those.

Table 7: Windows Server 2016 licenses

Windows Server Standard 2016	You must buy a minimum 16 core licenses per server = $883 per server • 5 servers = $4,415 • 4 servers = $3,532
Windows Server Datacenter 2016	You must buy a Minimum 16 core licenses per server = $6,155 per server • 5 servers = $30,775 • 4 servers = $24,620

A minimum of eight core licenses must be purchased per processor, and each server core has to be licensed. A 1 CPU Quad Core Server is no cheaper than a 2 CPU Quad Core, due to the minimum fee.

The reason I wouldn't recommend one physical server to host three or four virtual ones is that if it shuts down due to CPU overheating, it will shut down all running virtual servers with it. This leaves even the Hadoop standby node unavailable, and your Hadoop cluster becomes useless. You need to work out how mission critical the data and operations on your servers are going to be. Put yourself in the situation of something having gone wrong, and ask yourself what decisions you'd make. You may wish to use solid state drives (SSDs) or high-RPM hard disks. Of the two disk types, SSDs have more efficient energy use. While this is not a book about computer networking, if you hire someone to build a network for you, check that what you've specified is delivered. If you've paid for high-quality network components, find a way to check those components, and make sure inferior ones aren't used in parts of the network you can't see or that are underground.

Required Hadoop software

You will need:

- Syncfusion Big Data Agent v3.2.0.20
- Syncfusion Big Data Cluster Manager v3.2.0.20
- Syncfusion Big Data Studio v3.2.0.20

You can sign up for a free Syncfusion account to download the files from Syncfusion. For businesses with a turnover of less than £1,000,000, the Syncfusion Big Data software is free. For businesses with higher turnover, prices of around £4,000 per developer license are available. Free trials are available that are totally unrestricted.

Security and Active Directory

Syncfusion Big Data platform integrates seamlessly with Active Directory. It requires Active Directory to be installed with the DNS Server set up, and domain controllers in the AD forest.

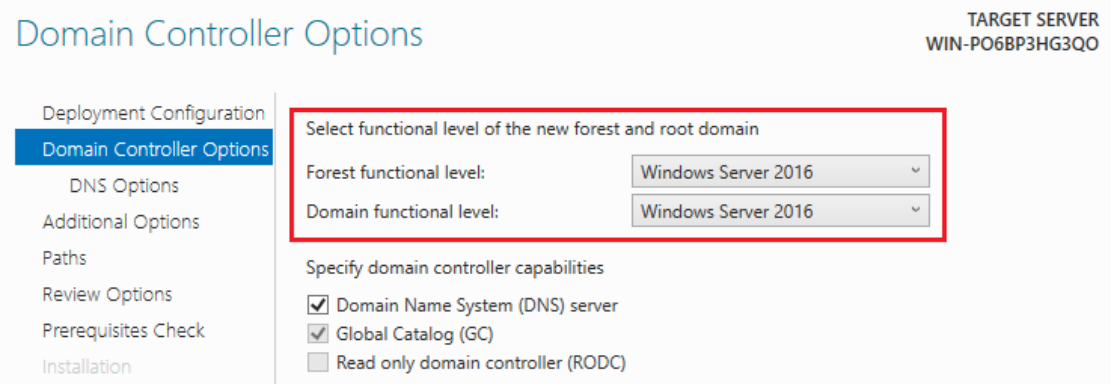


Figure 39: Choosing the functional level of the New Forest and Root Domain

An in-depth look at Active Directory is outside the scope of this book, but a competent Windows Administrator should be able to assist in setting it up. The servers in the Hadoop cluster must be part of the same domain; they should be joined to the domain via Active Directory, as shown in Figure 40. If they are not, DNS and reverse DNS validation can fail, and the Hadoop installation won't proceed. Just having computers on the same physical network is not enough.

www.dbooks.org

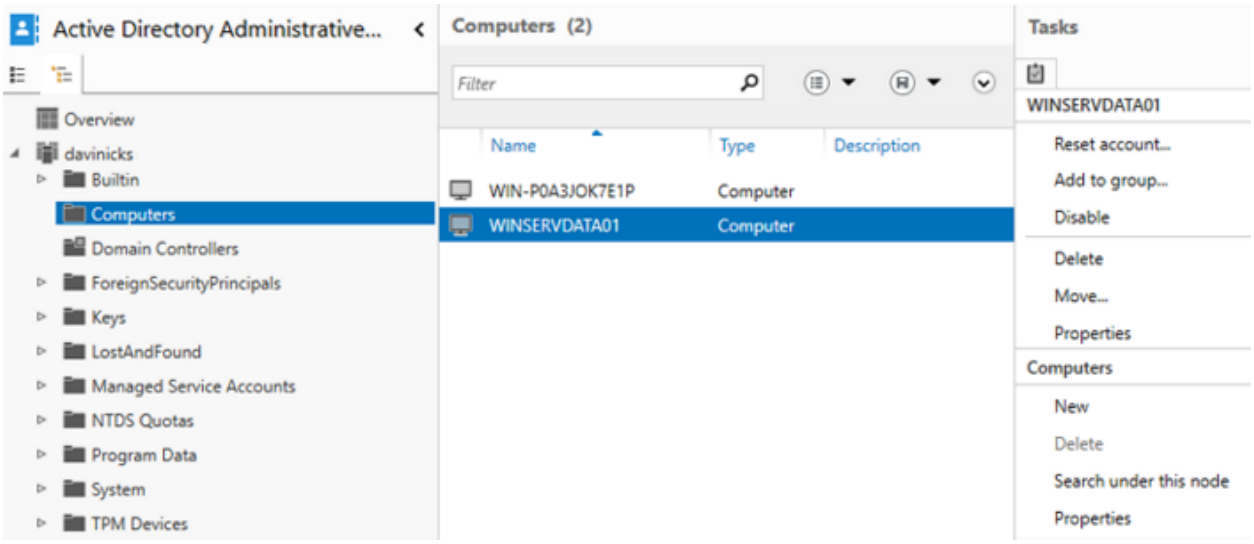


Figure 40: Adding machines to our domain using Active Directory Administration Center

Point each machine you want to join the domain to the IP address of the DNS machine. If not, your attempt to join computers to the domain may receive “DNS Server cannot be found” errors.

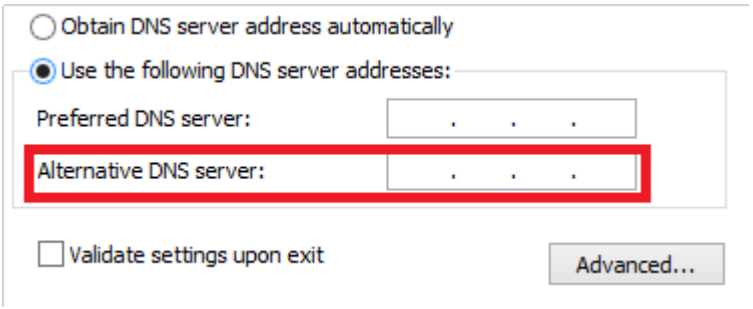


Figure 41: Point servers to the DNS server IP address

Enterprise Hadoop installation

To install Hadoop, we need to put the Syncfusion Big Data Agent on all machines we want to join the cluster. That means each server that is going to act as a cluster node requires the Big Data Agent. Please note that Port 6008 must be available on each server. This is the port the Syncfusion Big Data Agent listens on when placed on the Active, Standby, and data nodes. Do not install the Big Data Agent on the machine on which you are going to install the Cluster Manager. This is because the Cluster Manager manages the cluster, but isn't part of the cluster.

Install the Syncfusion Big Data Agent by running the downloaded Syncfusion Big Data Agent v3.2.0.20 file. Run the file as an administrator, and choose to install it to the default directory. The following installation screen appears before you see the final “Installed successfully” screen.



Figure 42: Installing Syncfusion Big Data Agent

After installing Syncfusion Big Data Agent, check that the Big Data Agent is running in Windows Server Services. Remember that it must be installed and running on all machines in the cluster.

Name	Description	Status
Syncfusion Big Data Agent	Syncfusion Big Data Agent is to manage Hadoop node and its Ecosystem.	Running

Figure 43: Syncfusion Big Data Agent running in Windows Server Services

Now run the Syncfusion Big Data Cluster Manager v3.2.0.20 file downloaded from Syncfusion. Install it as an administrator on the machine defined as the Cluster Manager. You install it in the same way you install any Windows software—this is the beauty of Syncfusion. Install it to its default location by simply following the instructions. After installation, start the Syncfusion Big Data platform from the standard Windows program menu. Once it's started, click the **Launch Manager** button under Cluster Manager, as highlighted in Figure 44.

www.dbooks.org

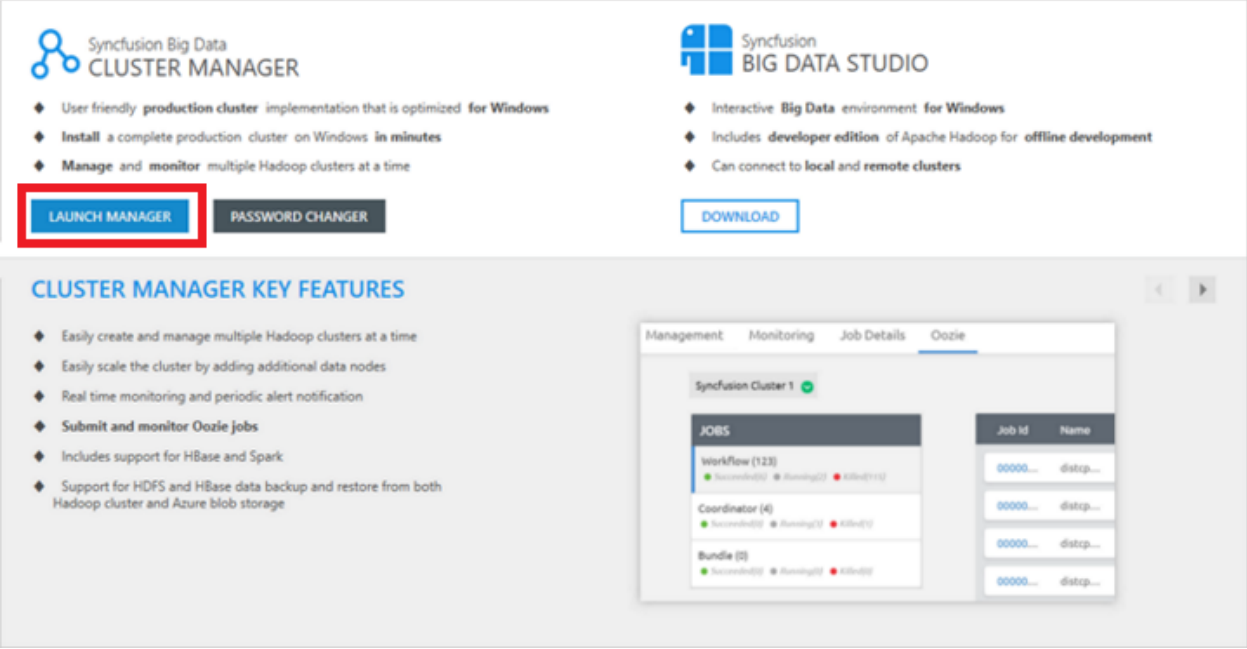


Figure 44: Syncfusion Big Data Platform screen

You will see the Syncfusion Cluster Manager interface, which opens in a web browser, as shown in the following figure.

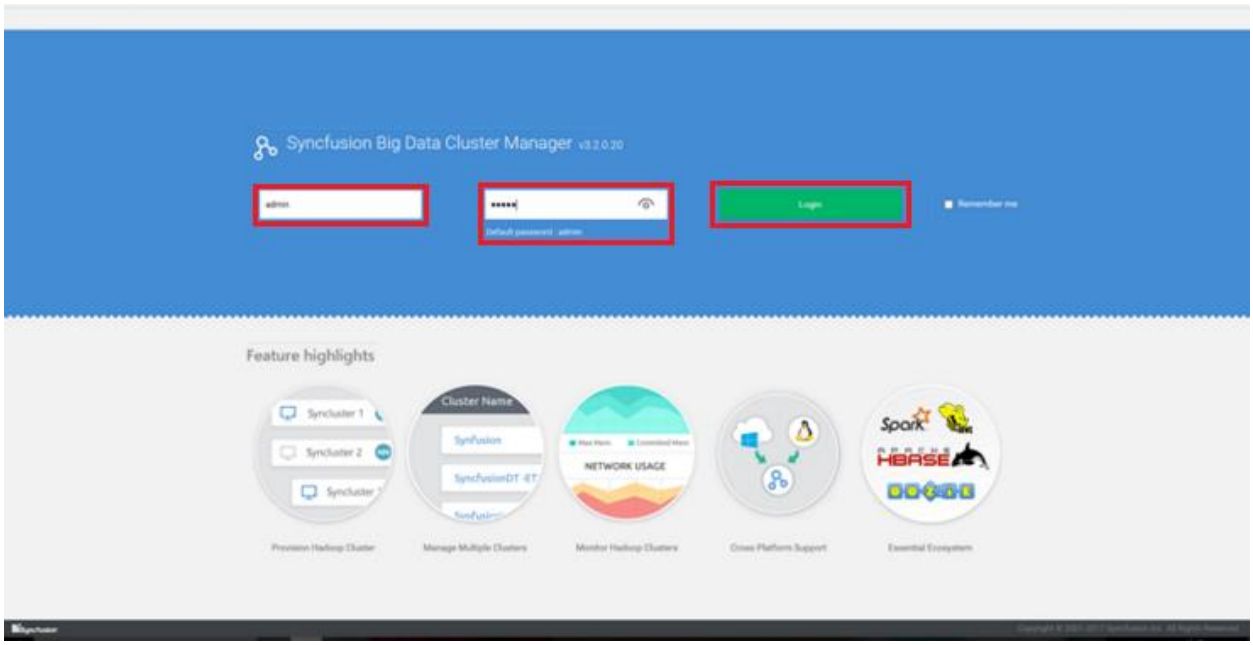


Figure 45: Browser-based Syncfusion Cluster Manager interface

Log in to the Cluster Manager with the default admin username and password, and click the green **Login** button shown in Figure 45.

Creating a multi-node Hadoop cluster in Windows

Once you're logged in, you'll see the screen in the following figure. On the right-hand side, you'll see the Create button.



Figure 46: The CREATE cluster button in Syncfusion Cluster Manager

The next figure shows a closer view of the button. Click **Create** to continue.



Figure 47: Closer view of the create cluster button in Cluster Manager

Choose **Normal Cluster** from the three options displayed, then click **Next**, as shown in the following figure.

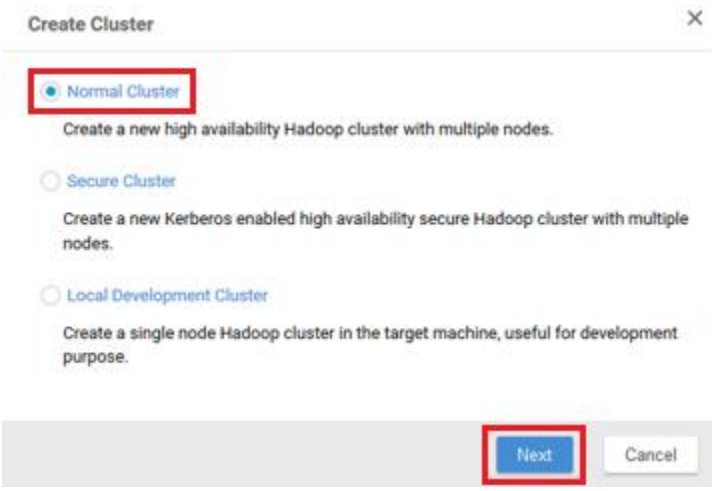


Figure 48: Choosing the type of cluster to create

On the next screen, choose **Manual Mode** and click **Next**. Provide a name for the cluster and leave the replication value at **3**. You then need to provide IP or host name information to identify and assign the following nodes.

Active Name Node:

www.dbooks.org



Figure 49: Adding cluster name and IP Address of the Active name node

Standby Name Node:



Figure 50: Add the IP Address of the Standby Name Node, leave the replication at 3

One or more Data Nodes:

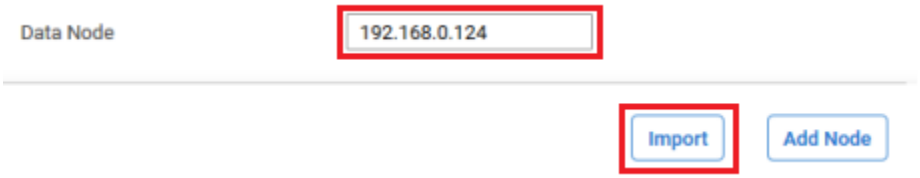


Figure 51: Add the IP Address of the data node

After adding the IP addresses of the servers, click the **Next** button on the top-right side of the screen. The Import option is for importing multiple host names or IP addresses from a single-column CSV file. To add additional data nodes (if needed), you'd click **Add Node**.

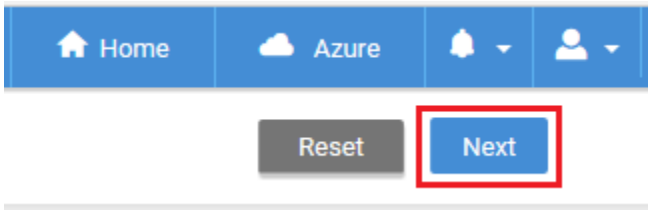


Figure 52: Location of the Next button

If after clicking **Next**, you see server clock-time errors (as shown in Figure 53), ensure that server clock times are within 20 seconds of one another. Synchronize the times of the servers in the cluster, then click **Next** again.

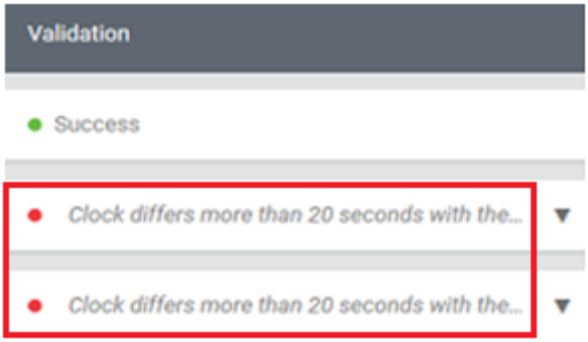


Figure 53: Server clock-time errors

The Cluster Manager resolves the proper host name and verifies that reverse DNS works. The Validation column displays **Success** and a green dot. Now, click **Next** on the top-right side of the screen.

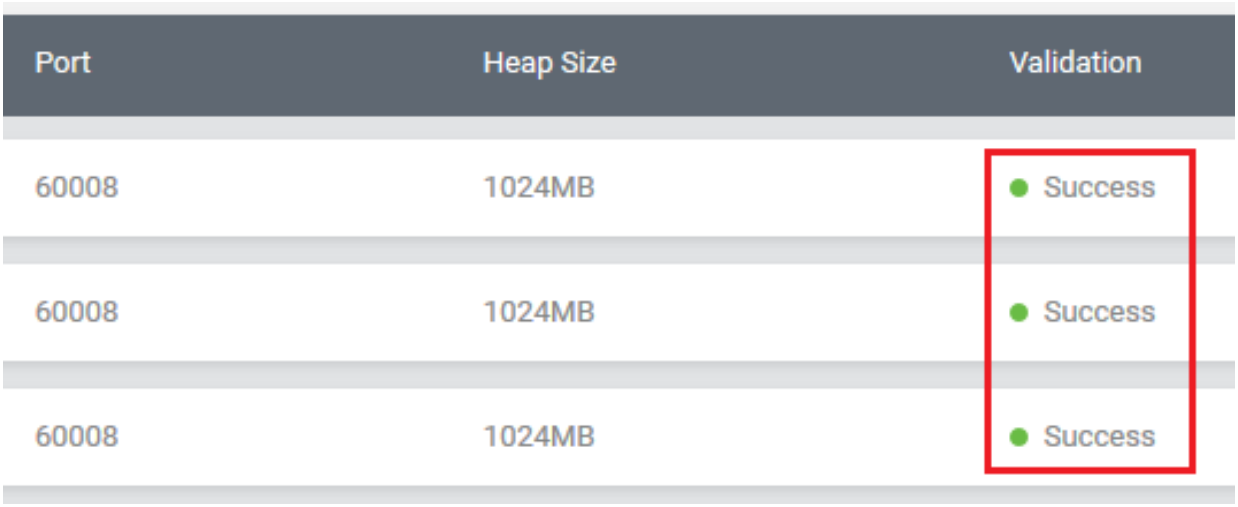


Figure 54: Reverse DNS and host name resolution success

The cluster should finish installing in 10–15 minutes, or quicker on a fast network.



Figure 55: Hadoop Cluster and ecosystem installing

www.dbooks.org

Upon completion, ensure that all the elements in the following screen are checked.

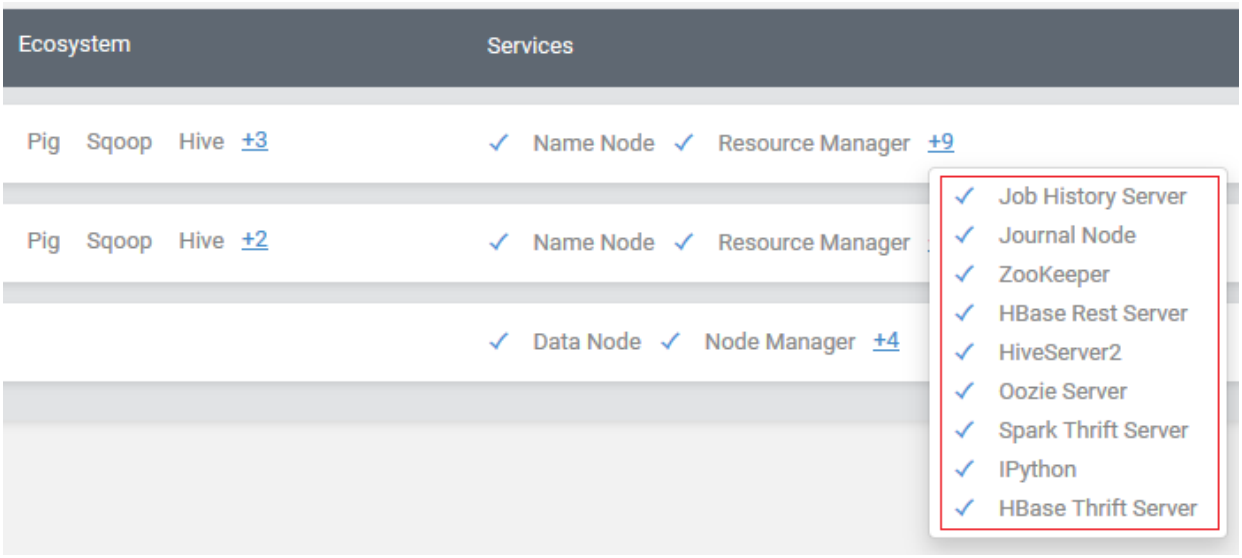


Figure 56: Hadoop and its ecosystem installed and running

If not, go to the very right-hand side of each white bar, which represents each node. You will see three gray dots, as shown in Figure 57. Click on the three dots, then click **Start Services** to select each element that is not running, to check that you can run all services. On a powerful machine on a fast network, you'll hopefully have no issues running all the services shown in Figure 56. On machines with less RAM, or where there are system or network bottlenecks, you may find you can't run all the services listed. Take care not to stop the node. If you do, be aware that when you go to start the node, there's an option to remove the node, so be careful. To prevent accidental removals, confirmation messages will appear and ask you to confirm any deletion actions.

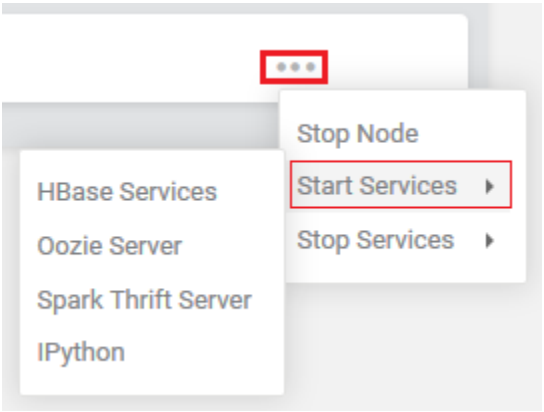


Figure 57: Starting services via the three gray dots

Cluster maintenance and management

Hadoop cluster maintenance is often done by Hadoop administrators. While respecting this, be aware that clusters can develop serious problems, and you want to prevent them before they happen. To assist us with this, we need to know the health of our clusters at all times.

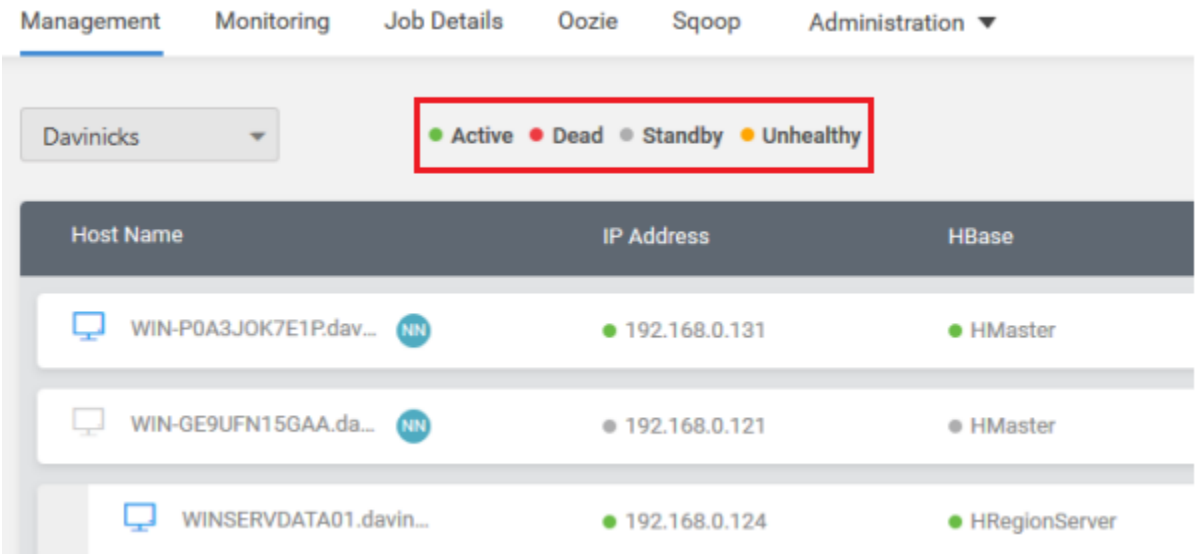


Figure 58: Status of Hadoop clusters in Syncfusion Cluster Manager

The Syncfusion Cluster Manager aids us in this by displaying the status of the nodes at all times. Figure 58 shows the four status levels that nodes can be classified as:

- **Active**: The active node is denoted by a green circle. This is why the name and data nodes have green circles by them in the IP Address and HBase columns.
- **Dead**: A dead node is denoted by a red circle, and while this is negative, it's also helpful to know before installation. You will recall the dead nodes denoted by red circles when the server clock times failed to synchronize. This allowed us to fix the problem by synchronizing the clock times so the installation could proceed. At that point, the notes turned to green.
- **Standby**: The standby node is denoted by a gray circle, and is shown with a gray circle in the IP Address and HBase columns. It is correct for the standby node to be displaying a gray circle, as it is on standby.
- **Unhealthy**: An unhealthy node is shown by an amber circle. Our cluster is showing no unhealthy nodes, but what if it was?

Whether it's a local or enterprise installation, it's imperative that you investigate why your cluster is unhealthy. You should be able to do this in any distribution of Hadoop, not just an enterprise one that does it automatically. If this isn't possible on all Windows machines running Hadoop, you can't realistically use it. To briefly test this, start the Hadoop installation we did in Chapter 1, and use the following command.

www.dbooks.org

Code Listing 13: Checking the status of the HDFS

```
hadoop fsck /
```

The output in the following image shows a healthy status in a fairly detailed output. It is worth taking a look at what those outputs mean, as they apply to any Hadoop installation.

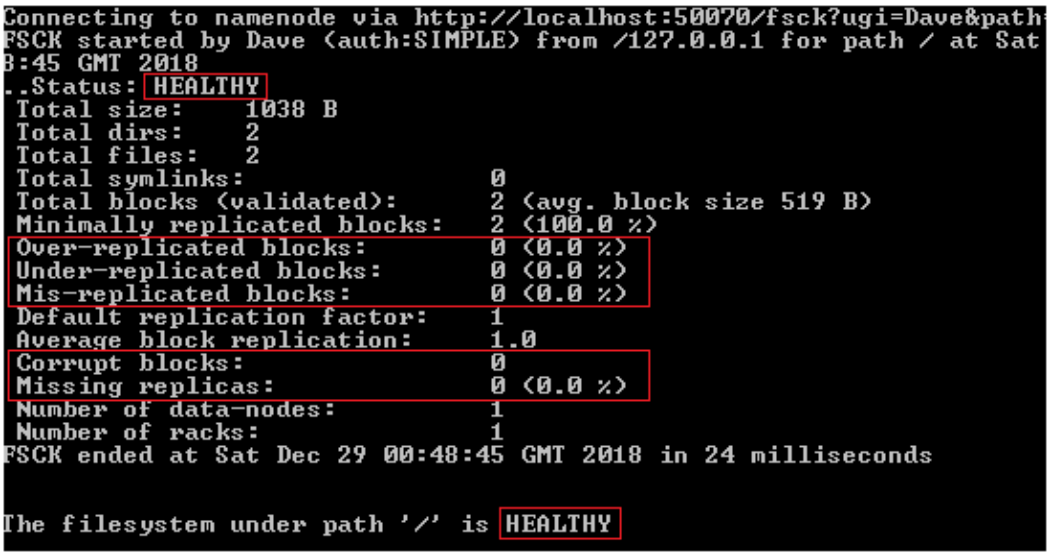


Figure 59: HDFS Healthy Status Result

- Over-replicated blocks: This refers to blocks that are replicated more than your chosen level of replication. HDFS can correct this on its own.
- Under-replicated blocks: This is basically the opposite of over-replicated blocks, and again, HDFS can correct this on its own.
- Mis-replicated blocks: These blocks involve a failure to replicate in line with your replication policy. You need to manually correct this, depending on the error you discover.
- Corrupt blocks: This is self-explanatory, and reflects corrupt blocks. This can be corrected by HDFS on its own if at least one block is not corrupt.
- Missing replicas: This is where a block has no replicas in the cluster.

If our cluster shows errors, one of the first things we can do is to seek more detail. This is achieved by using the following command to show the status of individual files.

Code Listing 14: Checking the status of files in HDFS

```
hadoop fsck / -files
```

This allows us to see the actual files that are involved in the blocks concerned so that if there were any issues, you would know which files you have to take action on. In the worst-case scenario, the ukhousetransactions.txt files we loaded into HDFS would need to be deleted using the **-delete** command. You can then replace them the same way we put them there in the first place.

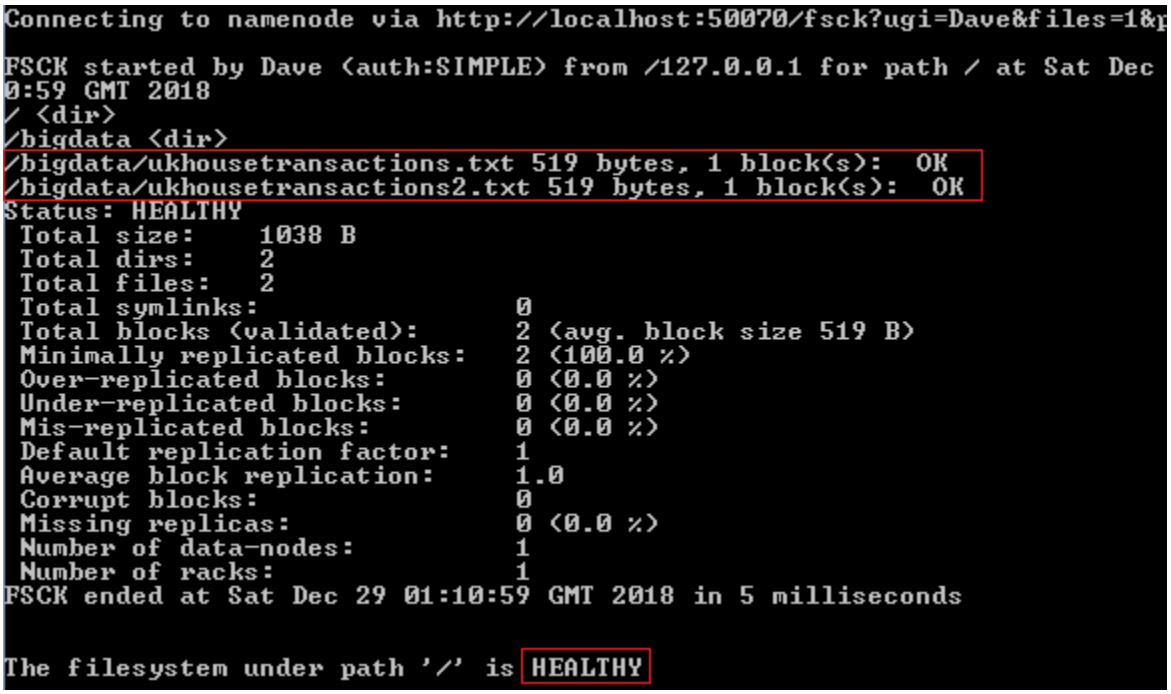


Figure 60: Results of the files status check

The **hadoop fsck / -files** command is also available in the Syncfusion Big Data Platform, and is included in the final piece of software we need to install. We will use the Syncfusion Big Data Studio v3.2.0.20 file downloaded from Syncfusion, and install it on the same machine as the Cluster Manager. You could install it on another machine on your network, but I'm doing it this way, as I want to show you something.

You install the software the same way you install any Windows software. Accept its default install location and follow the on-screen instructions until completion. Now, start the Big Data Studio from the Windows program menu, and you'll see the Syncfusion Big Data Platform screen, as shown in Figure 61. You'll notice that under Syncfusion Big Data Studio, the Launch Studio button has replaced the Download button. This is because the Cluster Manager and Big Data Studio are now recognized on the same server.

www.dbooks.org

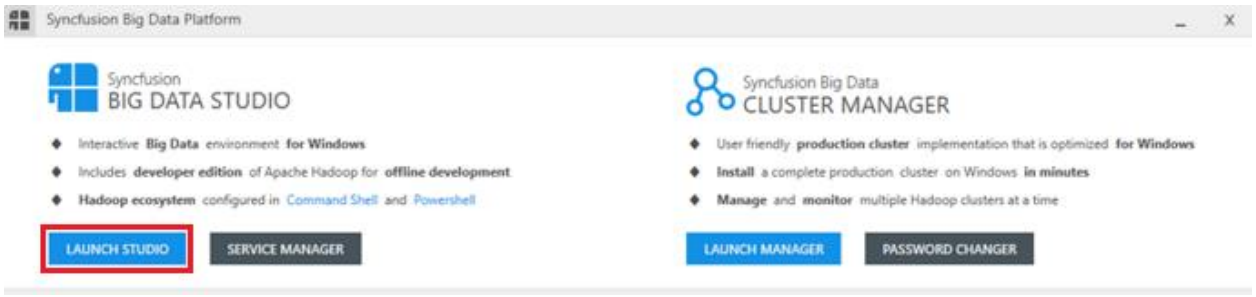


Figure 61: Cluster Manager and Big Data Studio on the same server

Click **Launch Studio**, as highlighted in Figure 61.

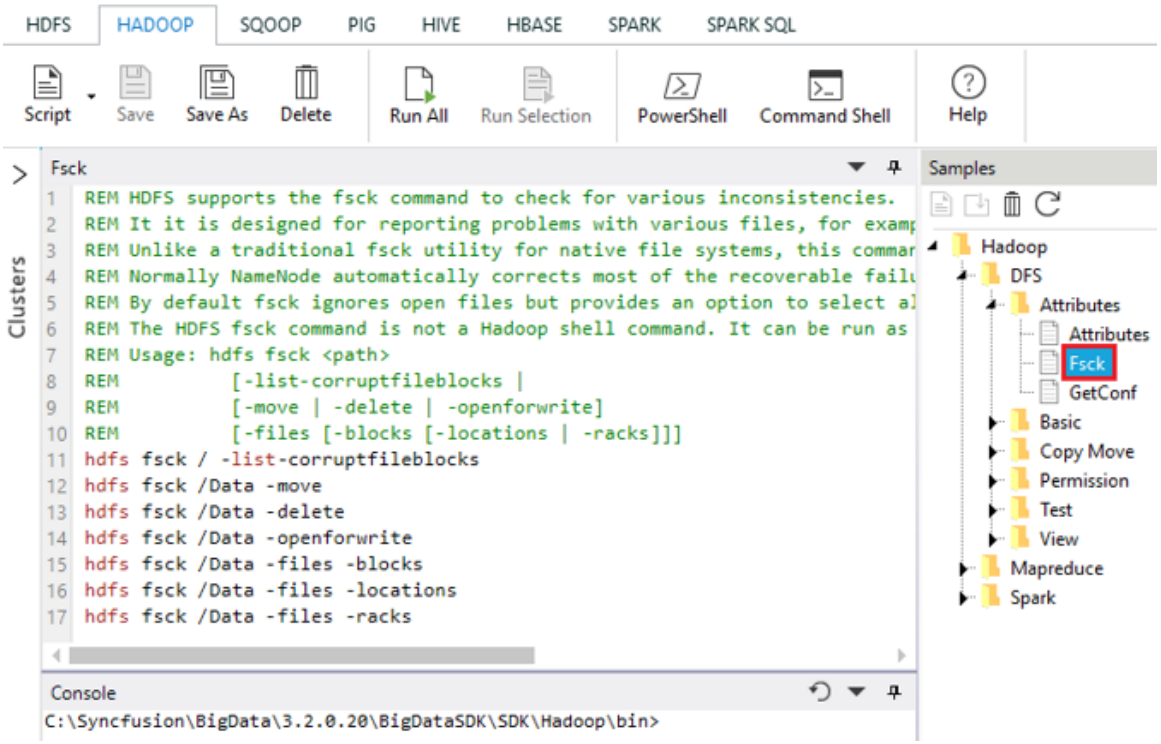


Figure 62: Fsck sample inside Syncfusion Big Data Studio

You are now inside the Syncfusion Big Data Studio where HDFS, Hive, and others are visible from the displayed tabs. If you click the **Hadoop** tab, you'll see a selection of samples. The Hadoop/DFS/Attributes folder has the **`fsck`** sample, as shown in Figure 62. If you go to the Big Data Platform main screen, you can click the **Command Shell** link shown in Figure 63, which you can use to launch a command prompt from the Big Data Studio.

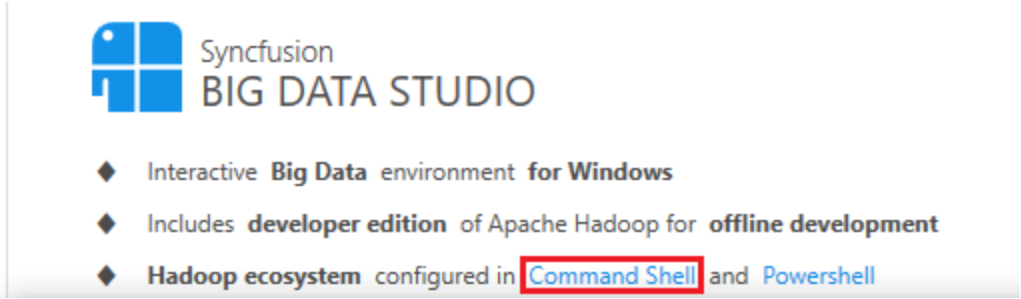


Figure 63: Launching the Command shell from the Big Data Platform main screen

From the command prompt, you can access Hadoop commands in the normal way. Syncfusion Big Data Platform is giving you flexibility to work within the more-interactive Windows environment, or to use the more traditional command-line environment.

Working with local development and live production clusters

An essential part of cluster management and maintenance is the ability to switch between production clusters and local development clusters. In Cluster Manager you can see not only the Davinicks cluster we created, but a second cluster, called Hadoop4windows, that was automatically added. Adding an existing cluster is a simple process compared to creating a new cluster, as we did earlier.

On the main screen in Cluster Manager, simply click the **Add** button on the top-right side of the screen.

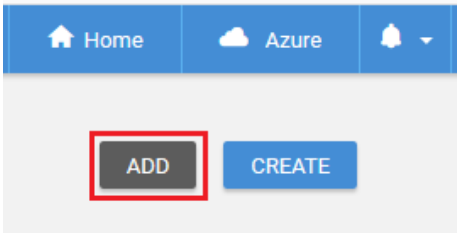


Figure 64: The Add cluster button

Next, you see a screen offering to manage an existing cluster; type in values for **Cluster Name** and **Name Node**, as shown in the following figure. Enter **localhost**, as when we installed the Syncfusion Big Data Studio, a local development cluster was automatically installed on the local machine. I named it Hadoop4windows, as I can call it any name I choose. Now click **Add**, and you'll notice a message stating "Collecting details of cluster."

www.dbooks.org

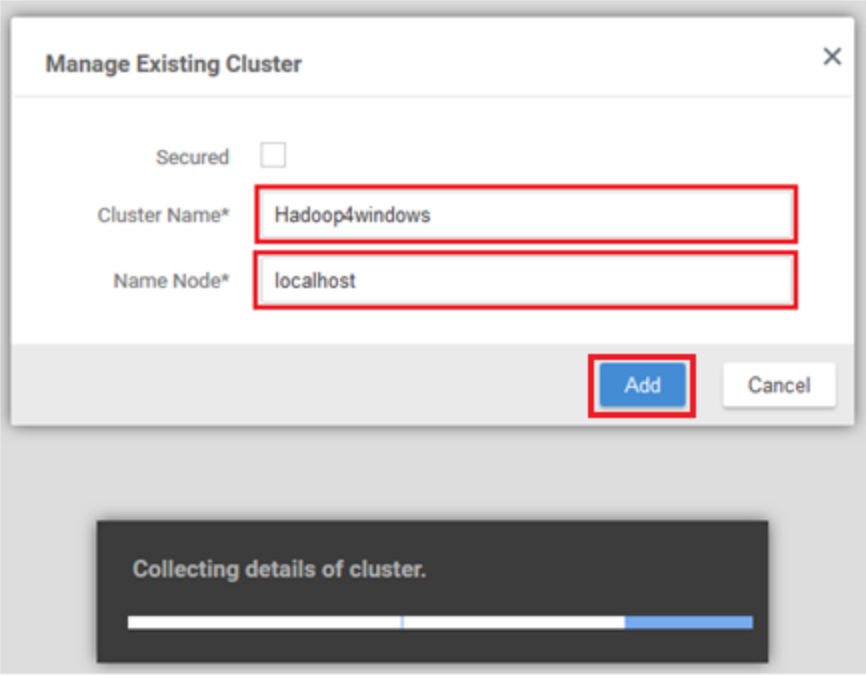


Figure 65: Adding an existing cluster in Cluster Manager

You now have two clusters, which are visible in the following figure: the multi-node cluster Davinicks, and the local development cluster Hadoop4windows. If you click a cluster under the Cluster Name column, you can access a screen with more cluster details. Click **Hadoop4windows**.

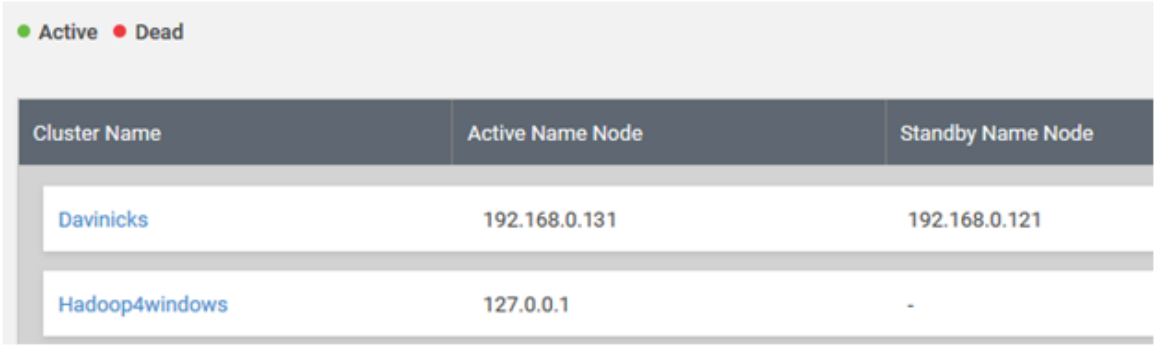


Figure 66: Multiple clusters in Cluster Manager

This takes you to a screen with menu items including Management and Monitoring, as shown in Figure 67. If you wish to switch between clusters at any time, simply click the dropdown box to select the cluster you'd like to work with.

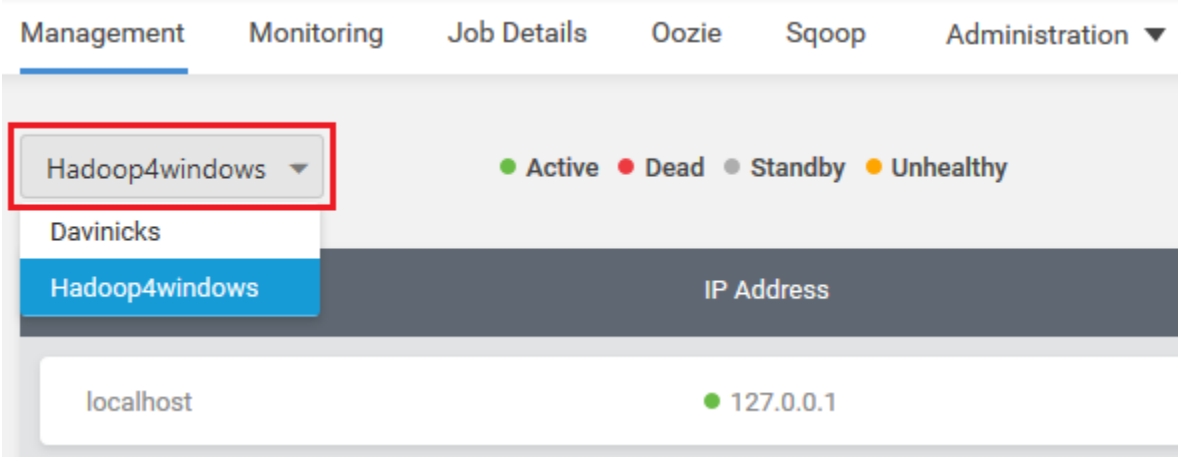


Figure 67: Switching between clusters in Cluster Manager

If you click the menu item called **Monitoring**, you can see the cluster Active Namenode and Active Datanode details. The green dots highlighted in Figure 68 denote that the nodes are active and healthy. The name of the cluster you're working with is shown on the top-left side of the screen.

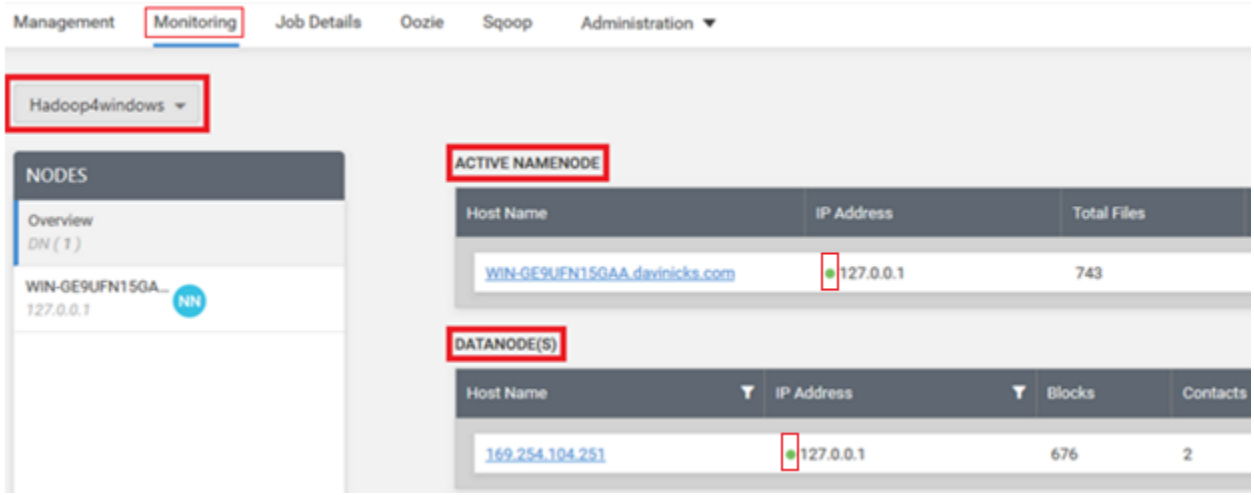


Figure 68: Monitoring your chosen cluster

We can now switch back to the production cluster called Davinicks by using the drop-down box. The Hadoop Services Monitoring screen is visible in the following figure.

www.dbooks.org



Figure 69: Hadoop Services Monitoring screen

The next figure takes a closer look at the Hadoop Services Monitoring screen for the Davinicks cluster. It shows there is rather more going on than on the local development cluster. On the left-hand side under NODES, you can see the Active Name Node highlighted in yellow, the Standby Name Node highlighted in green, and the Data Node in blue. The abbreviations NN, JN, and ZK stand for Name Node, Journal Node, and ZooKeeper, respectively. You can click on each node to see the Hadoop services for each one.

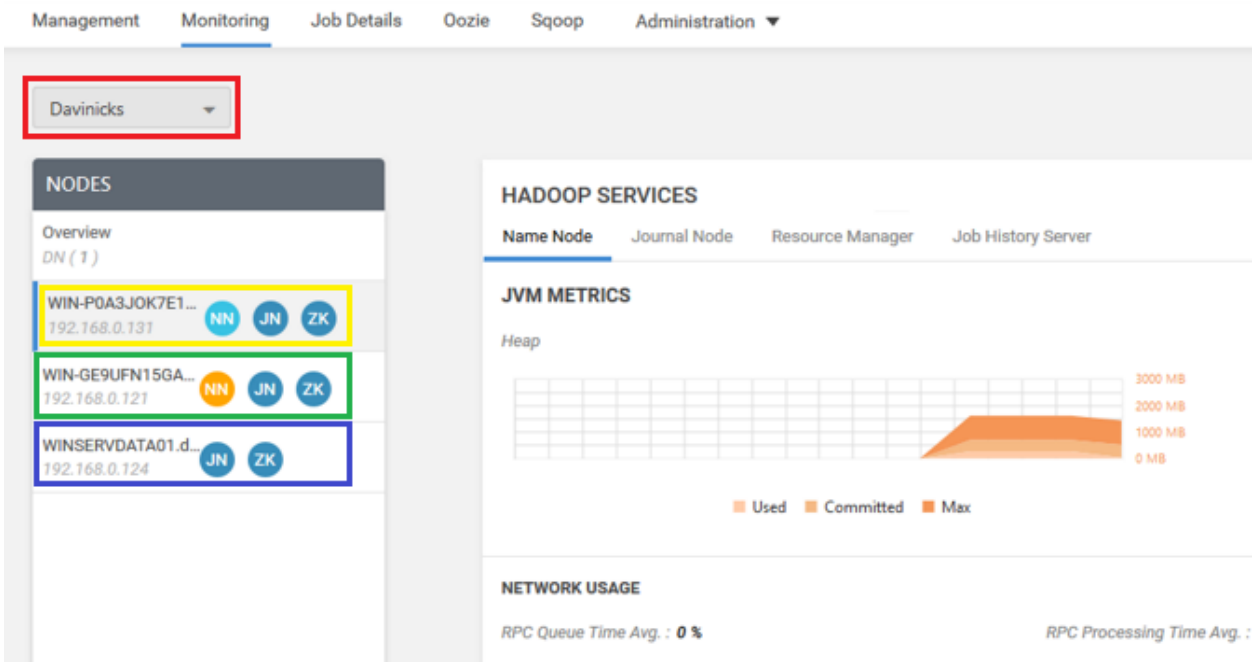


Figure 70: Zooming in on the Hadoop Services Monitoring screen

This web-style navigation allows you to see useful information on many pre-defined cluster elements. These include network usage, garbage collection, JVM Metrics, and useful information about how much RAM and disk space are available. You will also see the load on the CPU, IP addresses, and general machine configurations. It is useful that under Machine Configurations, there are three CPU information elements as highlighted: the System CPU Load, Process CPU Load, and Process CPU time.



Figure 71: Comprehensive CPU information under Machine Configurations

Experienced Windows Server administrators might say that Windows Server has many ways of accessing detailed information on similar variables, and they'd be right. That said, cluster manager is useful, as it presents data about server resources used by the cluster all in one place. You can access each Windows Server in the cluster and use its tools to compare what Cluster Manager is telling you for each node. You may also check where you feel cluster manager or Windows Server is under- or over-reporting resource use. You have to ask though, how practical is it to do that when running many clusters and nodes? It's not practical at all, which is why Cluster Manager is both essential and useful, and accurate enough to rely on. This is important, as Hadoop administrators won't use a cluster-management tool they don't trust.

Syncfusion Cluster Manager has other tools that are useful for monitoring clusters. Click the dropdown arrow next to the bell icon, as highlighted in Figure 72. You will see a circular settings icon lower down, which is also highlighted in the same figure. If you click on this icon, you will access an Alert Settings screen.

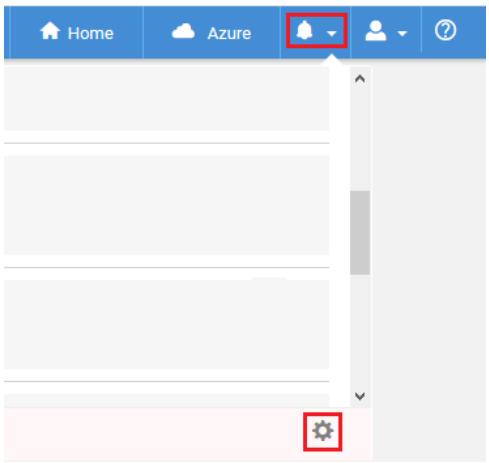


Figure 72: Accessing alert settings in Syncfusion Cluster Manager

www.dbooks.org

Check all the boxes for each cluster, as shown in Figure 73, and set the alert **Frequency** to minutes. Now, click **Save** in the top-right corner.

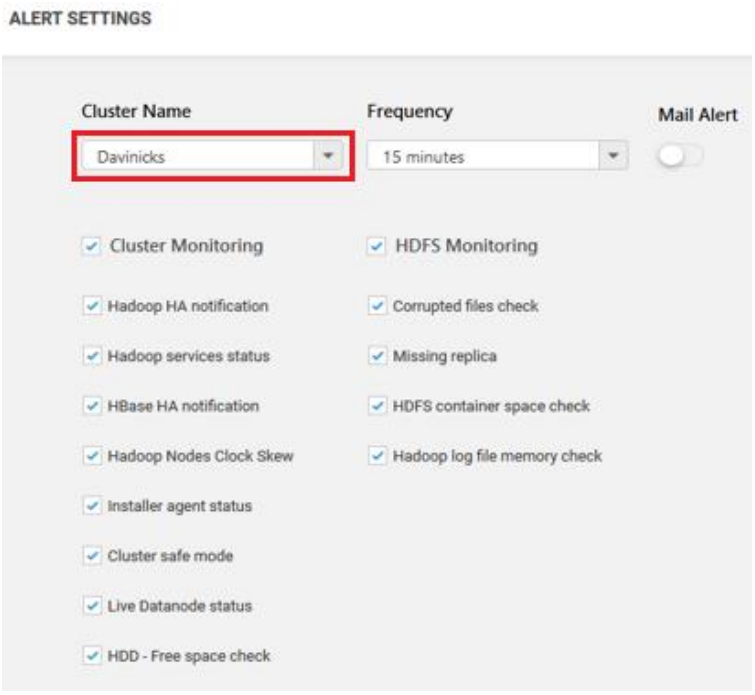


Figure 73: Selecting your clusters and choosing alert settings

I'm going to create some cluster faults that will trigger alerts, but I'm not advising you to do this in any way. After introducing the faults, the alerts box now shows the errors affecting the cluster.

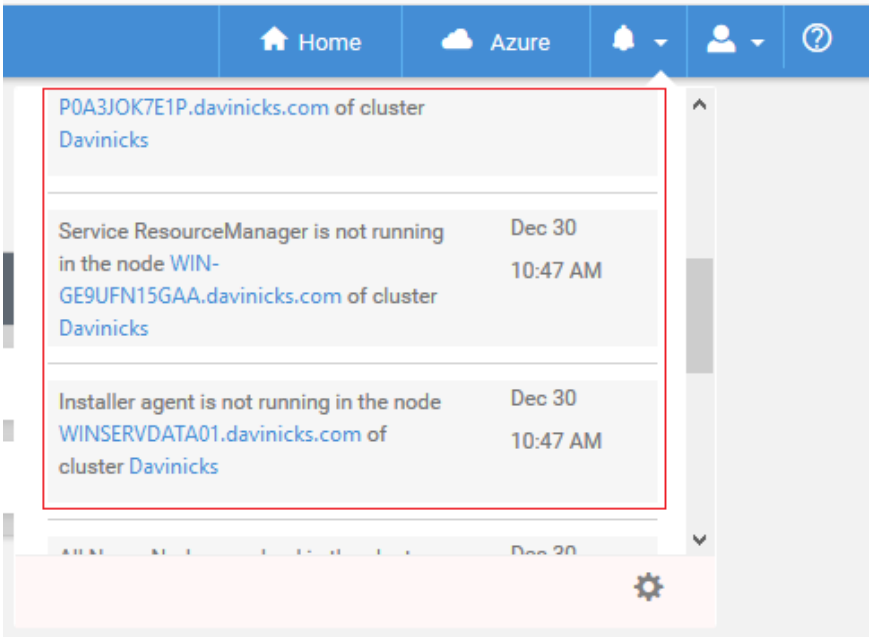


Figure 74: Alerts now showing in Cluster Manager

If you do not open or view the alert messages in Cluster Manager, you will see the number of alerts shown in white on a little red square. This is useful for letting you know there are errors as soon as you enter Cluster Manager.



Figure 75: Alerts in Syncfusion Cluster Manager

Other useful facilities for handling clusters include the ability to simply and quickly remove a cluster. On the same screen you've already used to create or add clusters, you'll see the Add and Create buttons we used. Go to the right-hand side of the cluster you wish to delete, and you'll see three gray dots, as highlighted in Figure 76. After you click on the dots, click **Remove** to remove the cluster, and answer any confirmation prompts.

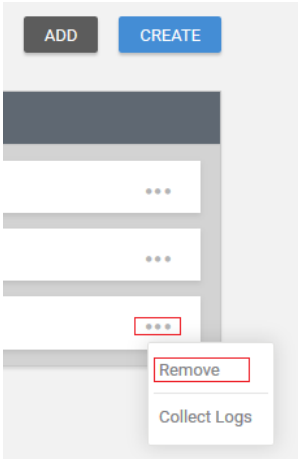


Figure 76: Removing cluster in Cluster Manager

The methodology for stopping individual nodes is to click the three gray dots on the right-hand side of each node, and then click **Stop Node**.

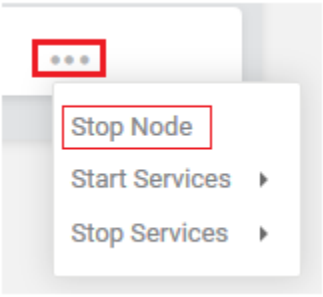


Figure 77: Stopping an individual node

At the top-right side of the screen, there are also facilities to stop or start all nodes using the Start/Stop button, as highlighted in Figure 78. In addition, there is a Manage Nodes button, which allows you to add both journal and data nodes in addition to removing dead nodes. You will remember that dead nodes are denoted by a red dot, as opposed to active nodes, which are denoted by a green dot.

www.dbooks.org

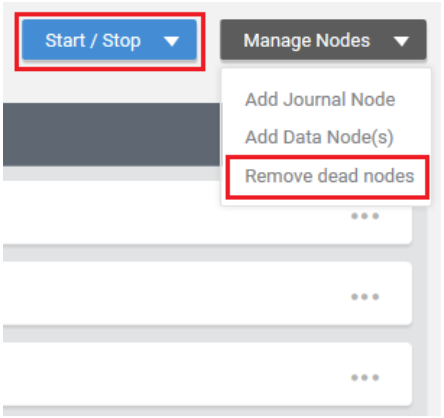


Figure 78: Remove dead nodes and Start/Stop all nodes

These facilities are useful when a node fails; they allow you to replace the failed node using the same method of node creation. This involves simply entering the IP address and node type of the node you wish to create.

To further manage and maintain clusters, we need to start putting them to work to see how they perform. To achieve this, we need to start ingesting data into Hadoop, which is covered in the next chapter.

Summary

In Chapter 2, we dealt with the network, environment, and server specifications for deploying multi-node Hadoop installations. We also covered Windows Server licensing and touched upon the setup of Active Directory. We created a Hadoop cluster using Windows Server machines and the Syncfusion distribution of Hadoop. We then ensured all components of Hadoop and the Hadoop ecosystem were installed and running without fault.

After installing Hadoop, we compared the cluster-management tools used in Apache Hadoop against those available in the Syncfusion Hadoop distribution. We established the availability of the command line, giving users choices for working within Hadoop for Windows. Cluster creation, swapping between clusters, and starting, stopping, and removing clusters was also covered. We highlighted facilities for monitoring clusters, with reference to the ability of Windows Server to monitor activity on each Hadoop node. We also set up cluster alerts and examined the methods used to start, stop, and remove cluster nodes.

Chapter 3 Programming Enterprise Hadoop in Windows

Hadoop performance and memory management within Windows Server

Consistency of Apache Hadoop builds across different Hadoop distributions is helpful, but doesn't mean that there'll be predictability of performance. In Windows, this can be both a blessing and a curse, but there are ways to ensure that it is a blessing.

Windows Server 2016 is a perfect partner for Hadoop from a technical perspective. It is capable of utilizing 256 processors, or 512 if Hyper-V is running. In addition, it can utilize 24 terabytes of RAM, or 12 on a virtual machine. In essence, it can handle any task that Hadoop may care to throw at it.

For this reason, I recommend that you use Windows Server for all the remaining exercises in this book. The only exception is if you use a local development cluster. In this case, you could use the latest version of Windows 10 Pro for Workstations, which can utilize four physical CPUs. Windows 10 Pro for Workstations and Windows 10 Enterprise could originally utilize only two physical CPUs. You would typically use the latest Windows 10 Pro for Workstations for local development, in conjunction with the multi-node cluster.

You'll need sufficient computing power to interrogate data after its ingestion into Hadoop. This doesn't just include dashboard creation tools, but also Relational Database Management Systems (RDBMS). All these tools must be running simultaneously, or you won't be able to connect to them live. This puts a load on both your network and the servers running within it. It's for these reasons that I detailed the hardware requirements earlier—Hadoop does not work in isolation in the real world.

In parts of the Hadoop ecosystem, like Hive, certain functions can be very resource intensive, such as creating joins between tables. Hive is not the only tool in which you can create joins but as it's a data warehouse I can' omit demonstrating it. Sadly, Impala is not available within Windows and its near RDBMS query speed in Hadoop is has made it indispensable for many Linux users.

We'll be using IMDB (Internet Movie Database) data files for our exercises, and there are optional exercises for those happy to download 4 GB of UK Land Registry data.

Let's examine how Hadoop in Windows will handle 4 GB of data, and the kind of performance we could expect. We'll utilize part of what is known as “Amdahl’s law” to aid us in this.

Getting 4 GB of data into Hadoop and processing it can be handled in a number of ways. It's partly dependent on the number of nodes you're using and the speed of your network. The aim is to process the data for use in dashboards or reports.

This can be expressed in the following ways:

- With four nodes, you could have four nodes processing 1 GB of data each.
- With eight nodes, you could have eight nodes processing 0.5 GB of data each.
- With 10 nodes, you could have 10 nodes processing 400 MB of data each.

www.dbooks.org

Let's say the size of the data query results after processing is 4 MB. Let's look at what is going on behind the scenes to achieve that. It's those behind-the-scenes factors that will affect performance, even if your final data query results or output are small.

We know that networks don't quite reach their stated maximum speed, but let's assume you consistently reach 40 Mbps on a 1-Gbps network.

Table 8: Hadoop options for processing 4 GB of data

Network speed in Mbps	Data to process in MB	Number of nodes	Process time per node in seconds	Size of data processed per node in MB	Total process time across all nodes in seconds	Size of final returned data in MB
40	4000	4	25	1000	**25** (100)	4
40	4000	8	12.5	500	**12.5** (100)	4
40	4000	10	10	400	**10** (100)	4
40	4000	12	8.33	333.33	**8.33** (100)	4
40	4000	16	6.25	250	**6.25** (100)	4

Table 8 shows that while your data can be processed faster, the time gains diminish the more nodes you add. The processing time drops from 25 seconds to 12.5 seconds when eight nodes are used instead of four. However, when 16 nodes are used instead of 12, data processing time only decreases by just over two seconds. Also, the data must be transported from more nodes to the node displaying the data, which adds to processing time. This leads to diminishing returns, to the point where adding more nodes becomes ineffective.

Another issue that's just as important is how Hadoop stores data in a cluster—it stores them in blocks. Often the blocks are 128 MB or 64 MB, so a 4-GB file stored in 128-MB blocks is stored across 32 blocks. With each file in Hadoop being replicated three times, you then go from 32 to 96 blocks.

A further complication is that you can only store one file per block. Therefore, if your blocks are 64 or 128 MB, then files much smaller than those block sizes are highly inefficient. This is because a 1-MB file is stored in the same 128-MB block size as a 110-MB file. It also requires metadata about each block to be stored in the RAM. Luckily, you can alter block sizes for individual files to tackle this issue. There are also other file formats you can use, such as Avro and Parquet, that can greatly compress the size of your files for use in Hadoop. The benefits can be great, with query times well over ten or twenty times as fast as queries on the uncompressed file.

Once you get a feel for how Hadoop stores data, you can begin to estimate how much disk capacity and computing power your Hadoop project will require. While the metadata for each data block uses only a tiny amount of RAM, what happens to your RAM when you have hundreds, or even thousands, of files? The inevitable happens, and that small amount of memory is multiplied by thousands. Suddenly your RAM is compromised, and you feel the impact of file uploads on performance and memory management.

I would recommend actually carrying out calculations before considering the ingestion of large numbers of files. This enables a demarcation between system resources required for Windows servers in the cluster, and system resources they're losing to running Hadoop. In Windows, the monitoring of individual Windows servers can't be ignored. They are Windows servers in their own right, in addition to being part of a cluster. Where you have more demanding requirements, you can add more RAM to avoid encroaching the base RAM Windows Server needs. You may conclude that big data systems are in fact better at handling big data than small data, which should be no surprise. The problem is that by using compression too much, you can end up creating the very thing you don't want, which is too many small files.

While this isn't something that's done too often, I think it is important to isolate what is actually happening resource- and performance-wise when Hadoop is running in Windows. This informs answers to questions such as: Should Hadoop be the only application on each server? I take a particular interest in this, as Windows does not have the control groups (cgroups) feature that is available to Linux users. Cgroups can control and prioritize network, memory, CPU, and disk I/O usage. Cgroups also control which devices can be accessed and are in most major Linux distributions, including Ubuntu. Further features include limiting processes to individual CPU cores, setting memory limits, and blocking I/O resources. It is a feature that Windows Server does not have, and that Linux developers would notice. Without these features, we need to take care to monitor resource usage for Hadoop in Windows.

Let's start by looking at the resources used by the Big Data Platform within Windows Server. You could do this by setting up a virtual machine in Windows Server and allocating as little as 12 GB, though I'd feel safer at 16 GB, as a minimum. The reason for this is a simple one: it's the same reason that Linux and Microsoft list the minimum requirements for their operating systems as low as possible. If you list the minimum requirements as too high, you'll put off some customers and diminish the distribution or sale of your product. The trick is to be realistic and state recommended minimum requirements, as we'll do now. If you launch the Syncfusion Big Data Platform on a virtual or physical server with a realistic minimum of 16 GB of RAM, you will hopefully see the Resource Monitor in Figure 79. It shows Windows Server at only 22 percent CPU usage and 35 percent memory usage; this includes resources used running Windows Server.

www.dbooks.org

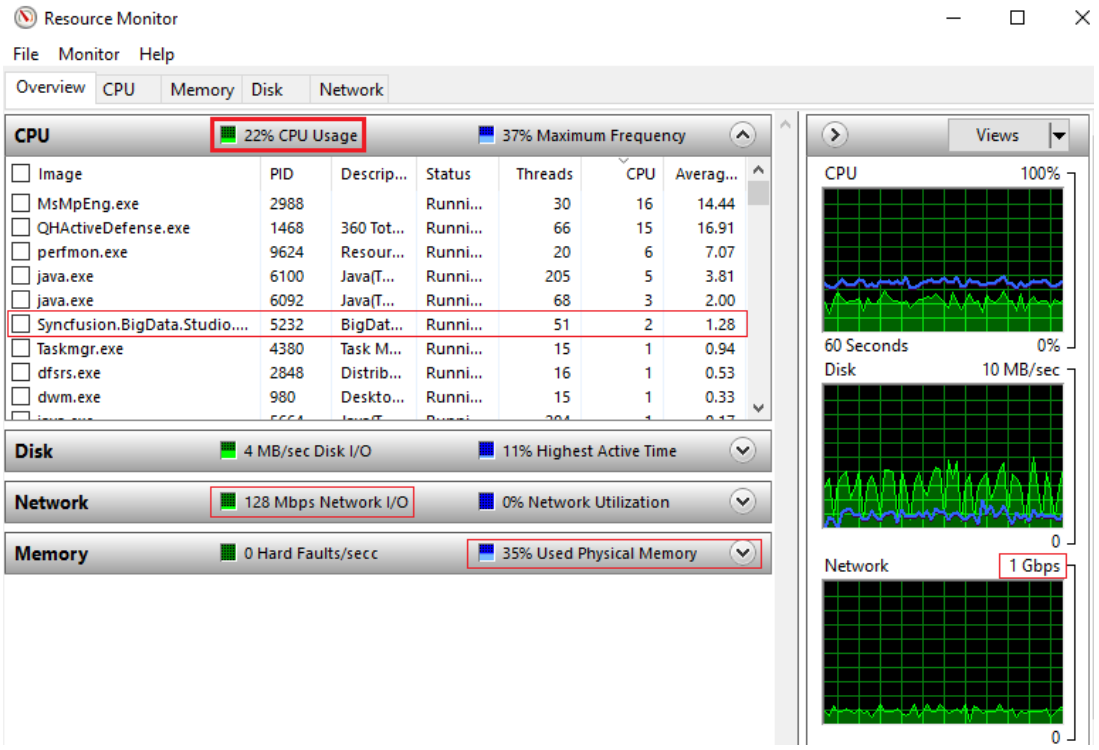


Figure 79: Syncfusion Big Data distribution running in Windows Server 2016

Note that 22 percent of CPU usage is from processes, with 13 percent CPU usage from services, so there is low resource usage.

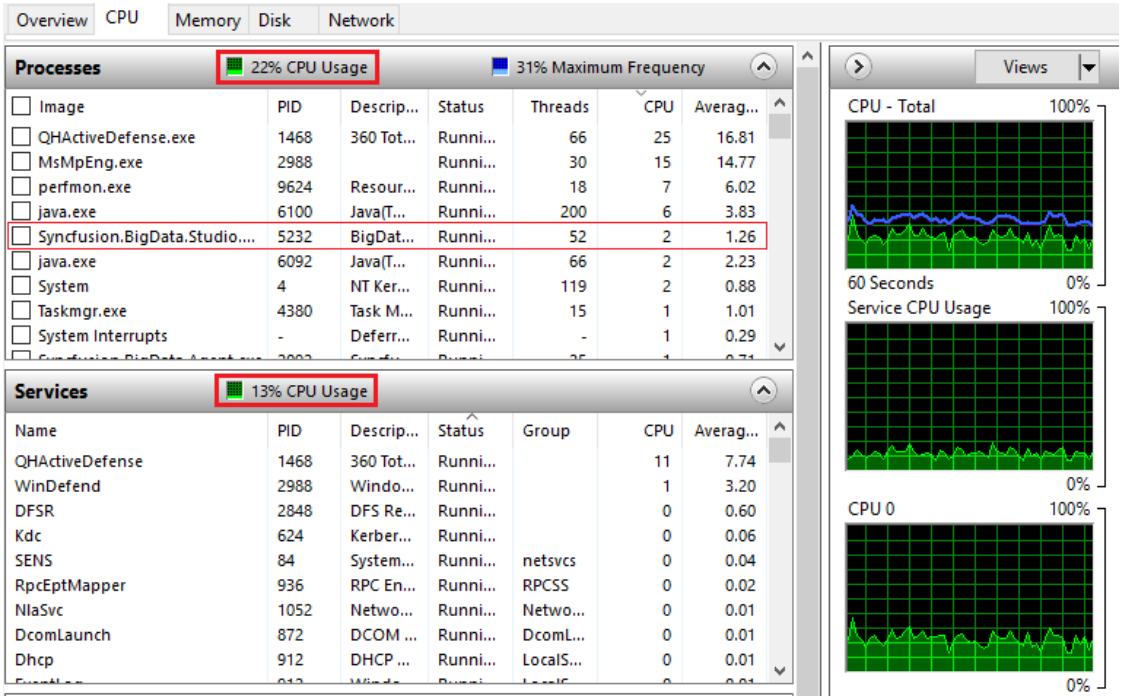


Figure 80: CPU processes and CPU services usage

At this point we have around 6 GB of memory in use by both Windows Server and the Big Data Platform, with approaching 3 GB on standby, and 7 GB free. Nearly 10 GB is available, which again, is stress-free computing. I am using i7 2.9 GHz CPUs, which are fast and robust processors.

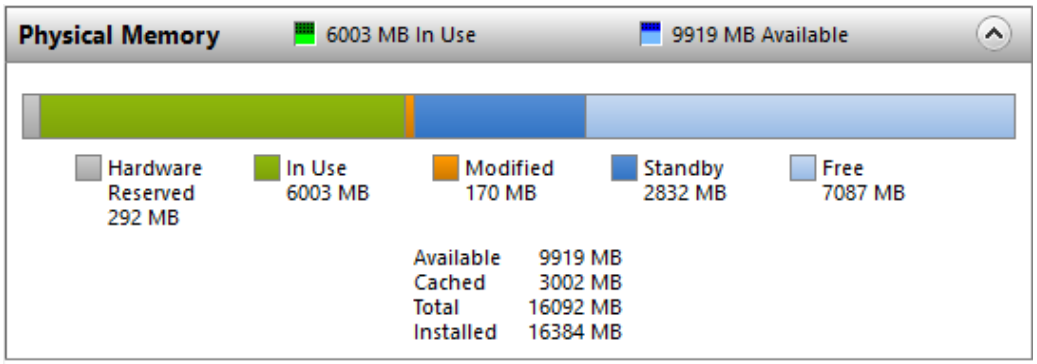


Figure 81: Big data platform memory management on Windows Server

If we run the Cluster Manager as well as the Big Data product, and upload 1 GB of data files, the CPU usage goes up to as much as 37 percent, but the RAM behaves differently. Figure 82 shows that while memory usage is about the same 35-percent usage as before, the RAM use has gone down about 250 MB from the previous figure. More interestingly, the RAM on standby has increased to almost 5 GB, from under 3 GB. This decreases the amount of free RAM to under 5 GB. While 16 GB allows you to get up and running very comfortably, we know that for heavy lifting, we need 32 GB upwards. You have to act before—and not when—free memory runs out.

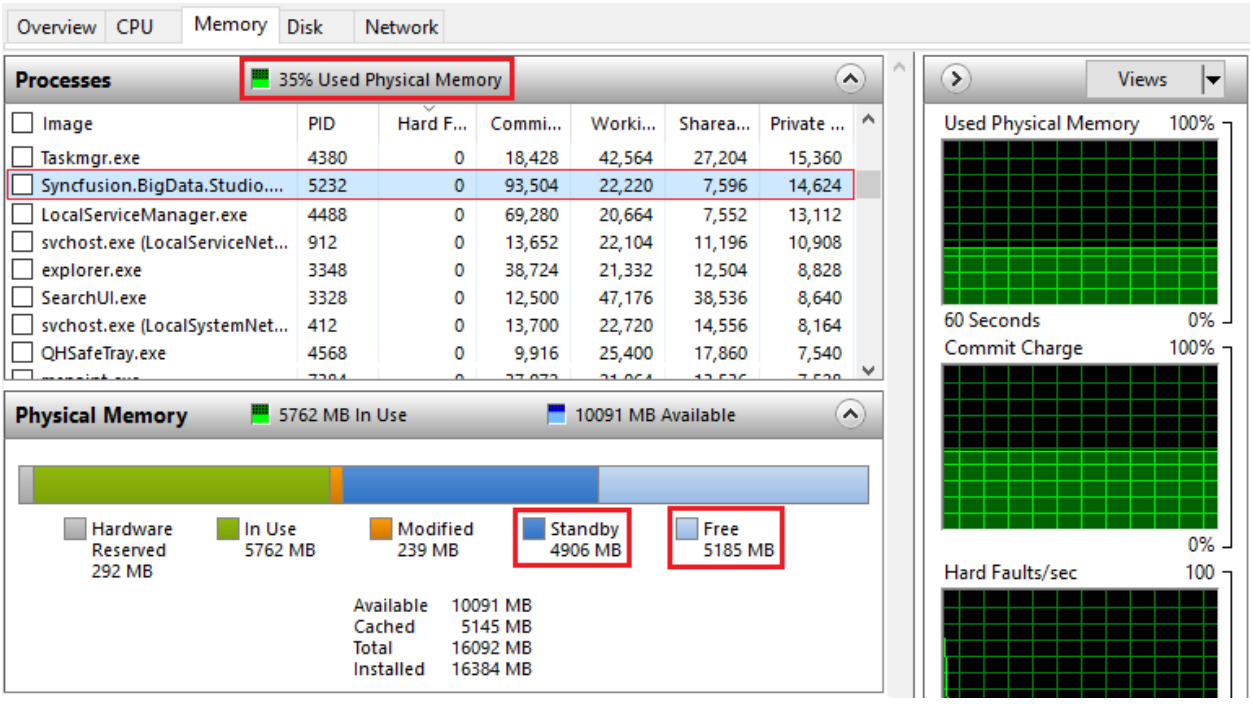


Figure 82: More RAM placed on standby leaving less free

www.dbooks.org

I mentioned the IMDB data we'd be using to demonstrate data ingestion into the Syncfusion distribution of Hadoop. I am purposely using the .tsv file format for IMBD data and the .gz compressed file format.

The IMDB data is available from here.

Subsets of IMDB data are available for access to customers for personal and non-commercial use. You can hold local copies of this data, and it is subject to terms and conditions.

Download the zipped .gz files, then unzip the .tsv files using a tool like 7-Zip. The files include:

Title.basics.tsv.gz – contains the following information for film titles:

- **`tconst`** (string): Alphanumeric unique identifier of the title
- **`titleType`** (string):The type/format of the title (for example, **`movie`**, **`short`**, **`tvseries`**, **`tvepisode`**, **`video`**, etc.)
- **`primaryTitle`** (string): The more popular title, or the title used by the filmmakers on promotional materials at the point of release
- **`originalTitle`** (string): Original title, in the original language
- **`isAdult`** (boolean): **`0`** = non-adult title; **`1`** = adult title
- **`startYear`** (YYYY): Represents the release year of a title; in the case of a TV series, it is the series start year
- **`endYear`** (YYYY): TV series end year; **`\N`** for all other title types
- **`runtimeMinutes`**: Primary runtime of the title, in minutes
- **`genres`** (string array): Includes up to three genres associated with the title

Title.akas.tsv.gz – contains the localized film title:

- **`titleId`** (string): A **`tconst`**, an alphanumeric unique identifier of the title
- **`ordering`** (integer): A number to uniquely identify rows for a given **`titleId`**
- **`title`** (string): The localized title
- **`region`** (string): The region for this version of the title
- **`language`** (string): The language of the title
- **`types`** (array): Enumerated set of attributes for this alternative title; can be one or more of the following: **`alternative`**, **`dvd`**, **`festival`**, **`tv`**, **`video`**, **`working`**, **`original`**, **`imdbDisplay`**. New values may be added in the future without warning.
- **`attributes`** (array): Additional terms to describe this alternative title, not enumerated

Title.episode.tsv.gz – contains the TV episode information:

- **`tconst`** (string): Alphanumeric identifier of episode
- **`parentTconst`** (string): Alphanumeric identifier of the parent TV series
- **`seasonNumber`** (integer): Season number the episode belongs to
- **`episodeNumber`** (integer): Episode number of the **`tconst`** in the TV series

Title.ratings.tsv.gz – contains the IMDB rating and votes information for titles:

- **`tconst`** (string): Alphanumeric unique identifier of the title
- **`averageRating`** (integer): Weighted average of all the individual user ratings
- **`numVotes`** (integer): Number of votes the title has received

We will also download UK Land Registry data, which is approaching 4 GB in file size. We require the 3.7-GB .csv file.

These include standard and additional-price-paid data transactions received at HM Land Registry from January 1, 1995 to the most current monthly data. The data is updated monthly, and the average size of this file is 3.7 GB, you can download the .csv file here. The data cannot be used for commercial purposes without the permission of HM Land Registry.

Hive data types and data manipulation language

One of the best sources of information on Hive data types and data manipulation language can be found here and here.

These two resources provide exhaustive information, whereas this section of the books lists essential information. Those of you who know SQL may find similarities between Hive data types and SQL data types; the same can be said for Hive Query Language and SQL.

Hive data types

Numeric types:

- **`TINYINT`** (1-byte signed integer, from -128 to 127)
- **`SMALLINT`** (2-byte signed integer, from -32,768 to 32,767)
- **`INT`**/**`INTEGER`** (4-byte signed integer, from -2,147,483,648 to 2,147,483,647)
- **`BIGINT`** (8-byte signed integer, from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
- **`FLOAT`** (4-byte single precision floating point number)
- **`DOUBLE`** (8-byte double precision floating point number)
- **`DOUBLE PRECISION`** (alias for DOUBLE, only available starting with Hive 2.2.0
- **`DECIMAL`**: Introduced in Hive 0.11.0 with a precision of 38 digits Hive 0.13.0 introduced user-definable precision and scale
- **`NUMERIC`** (same as DECIMAL, starting with Hive 3.0.0)

Date/time types:

- **`TIMESTAMP`** (Only available starting with Hive 0.8.0)
- **`DATE`** (Only available starting with Hive 0.12.0)
- **`INTERVAL`** (Only available starting with Hive 1.2.0)

String types:

- **`STRING`**
- **`VARCHAR`** (Only available starting with Hive 0.12.0)
- **`CHAR`** (Only available starting with Hive 0.13.0)

www.dbooks.org

Misc types:

- **BOOLEAN**
- **BINARY** (Only available starting with Hive 0.8.0)

Complex types:

- arrays: **ARRAY**<data_type>
- maps: **MAP**<primitive_type, data_type>
- structs: **STRUCT**<col_name : data_type [COMMENT col_comment], ...>
- union: **UNIONTYPE**<data_type, data_type, ...>

Hive DML (Data Manipulation Language) provides support for:

- Loading files into tables
- Inserting data into Hive tables from queries or sub-queries
- Dynamic Partition Inserts
- Writing data into the filesystem from queries
- Inserting values into tables from SQL
- Updates
- Deletes
- Merges
- Joining of tables
- Grouping or aggregating the values of columns

Hive DDL (Data Definition Language) provides support for the following:

- Create database
- Describe database
- Alter database
- Drop database
- Create table
- Truncate table
- Repair table

DDL functions also apply to the components of databases such as views, indexes, schemas, and functions. The best way of detailing or describing the terms and functions shown previously is to actually use them. To achieve this in Hadoop for Windows, we need to prepare our environment for data ingestion.

Enterprise data ingestion and data storage

To prepare our Hadoop environment for data ingestion and storage, launch the Syncfusion Big Data Studio and select the **HDFS** menu item. You'll notice in Figure 83 that there are some directories within Big Data Studio that are automatically loaded upon installation. There is also an Add Cluster button underneath the Clusters section. Underneath this button is the text **localhost**, which is the local development cluster installed automatically when you installed Big Data Studio. As we wish to connect to our multi-node cluster, let’s click the **Add Cluster** button.

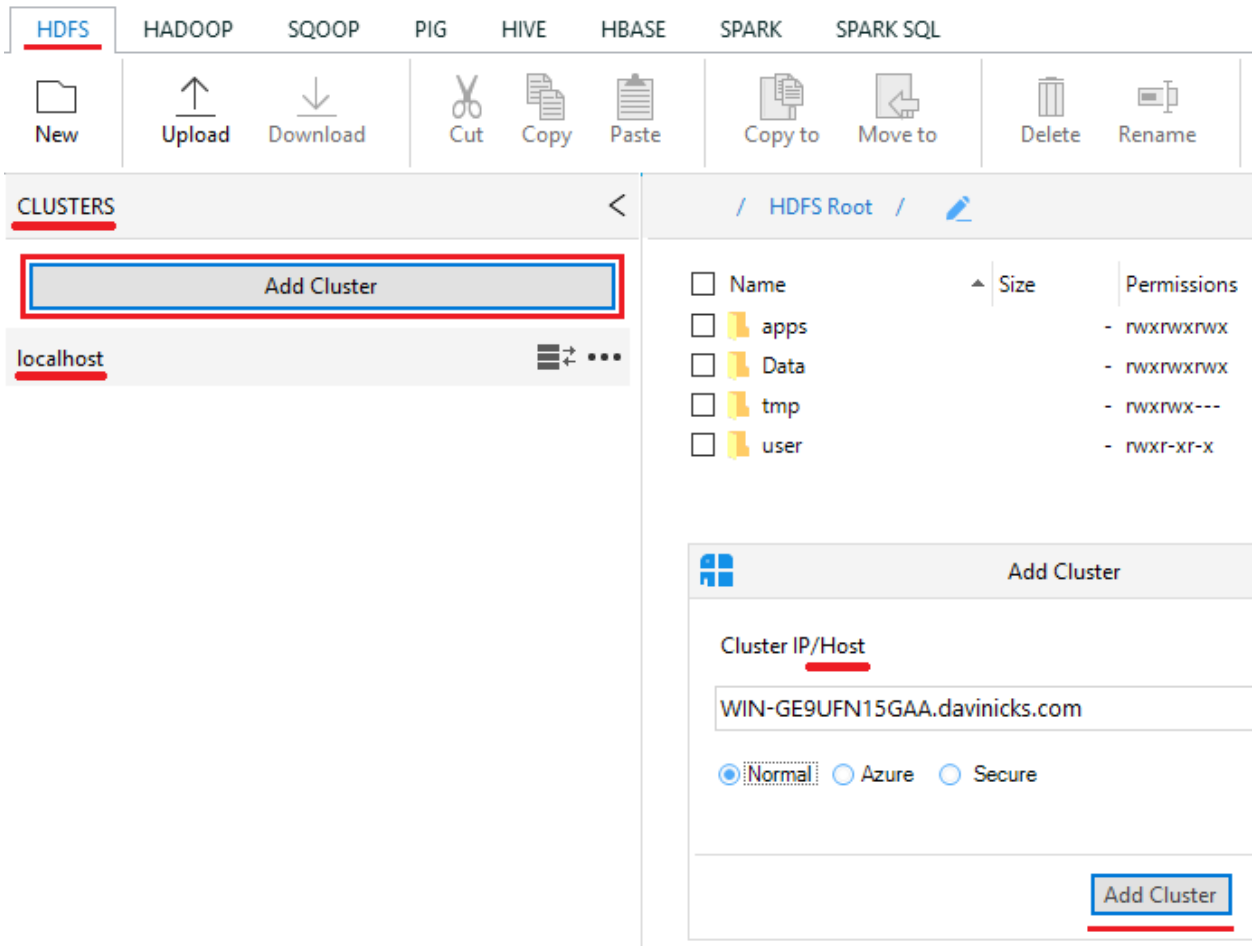


Figure 83: Add Cluster to Syncfusion Big Data Studio

We need to enter the IP address of the host name of the active name node of our cluster. Enter the host server name or IP address as requested. In Figure 83, the full computer name (including domain element) is given, and it connects to the cluster, as seen in Figure 84.

www.dbooks.org

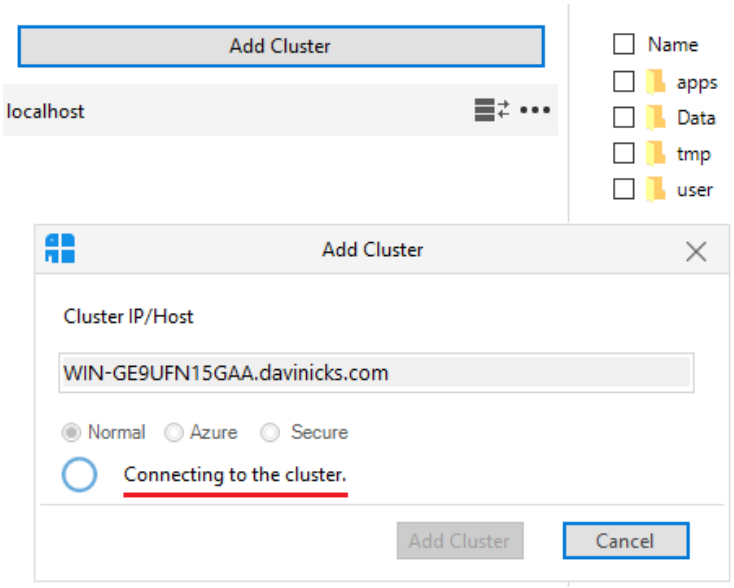


Figure 84: Connecting to Hadoop cluster

The cluster is shown underneath the localhost cluster. To start receiving data, click **New** to add a new folder, enter a name for the folder, and click **Create**.

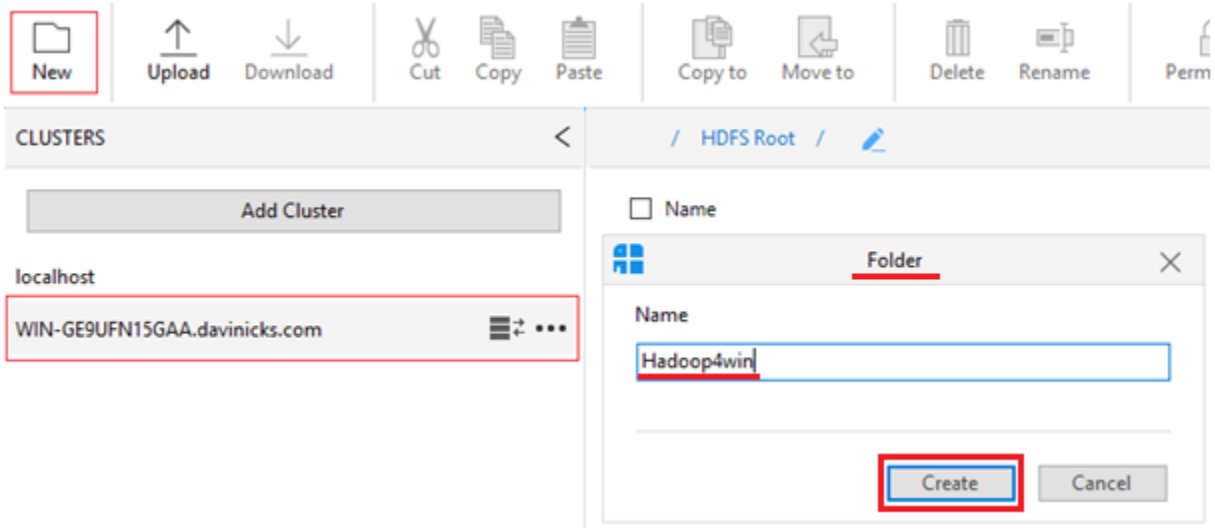


Figure 85: Creating a new folder on a cluster

You can also switch to your local cluster and create a folder there, by selecting the **localhost** cluster and clicking **Start Services**.

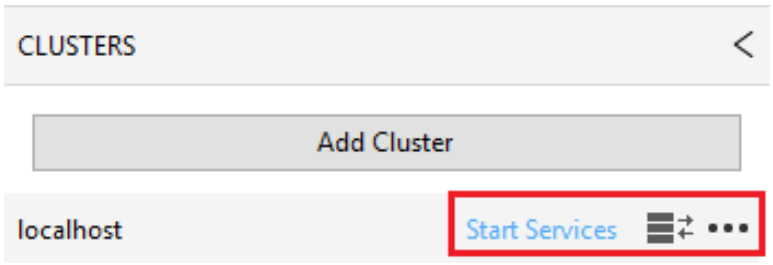


Figure 86: Starting services on the local cluster

This means you can switch between clusters by simply connecting or disconnecting.

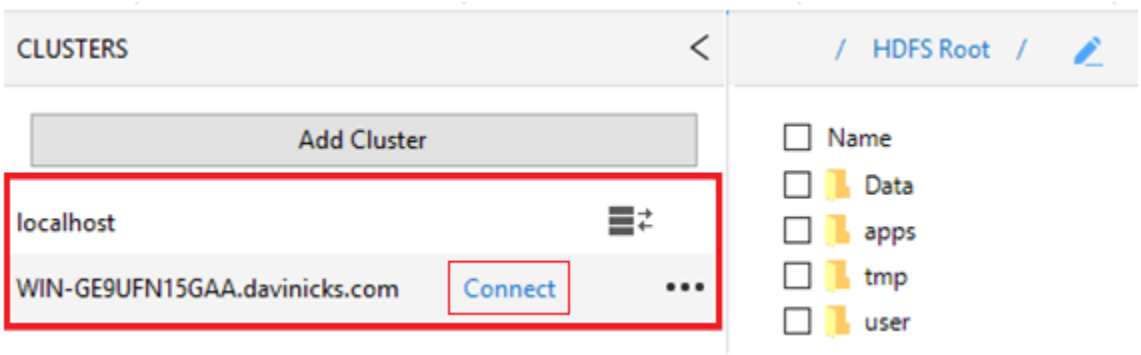


Figure 87: Option to switch between clusters

We are now going to do our first data upload to Hadoop. We are going to upload the 3.7-GB .csv file of house prices paid transactions mentioned earlier in the chapter. If you can't get ahold of the file, just follow along, as we'll be using files of a much smaller size in the next section. In Big Data Studio, choose the **HDFS** menu item. Click **New** to create a new folder, and call it **ukproperty**. It will appear to the right of the text **HDFS Root**, which is shown in Figure 88.

If you click **ukproperty**, you will access the folder. Once there, you can create another folder called **2018Update**. You will now notice that the 2018Update folder has been added to the right of ukproperty, and if you click **2018Update**, you will access that folder.

Next to the New folder button on the menu, click **Upload** and choose the radio button to select **File**. Use the button highlighted in the red rectangle in Figure 88 to select the property transactions file we are going to upload, called **pp-complete.csv**. Once you find and select the file, click **OK**, and you will see the following screen again. Now, click **Upload**.

www.dbooks.org

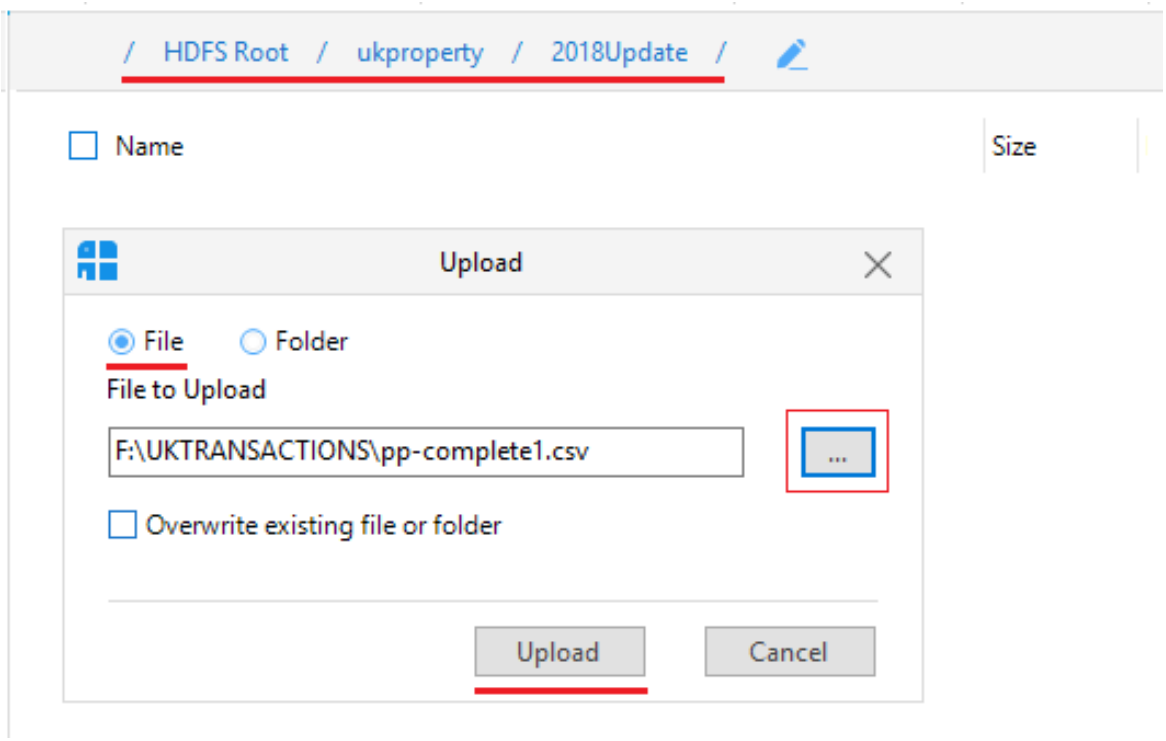


Figure 88: Selecting file for HDFS upload

On the bottom, right-hand corner of the screen, you will see a box that shows a progress bar of the file uploading. When the file has been uploaded, it shows **Completed**.

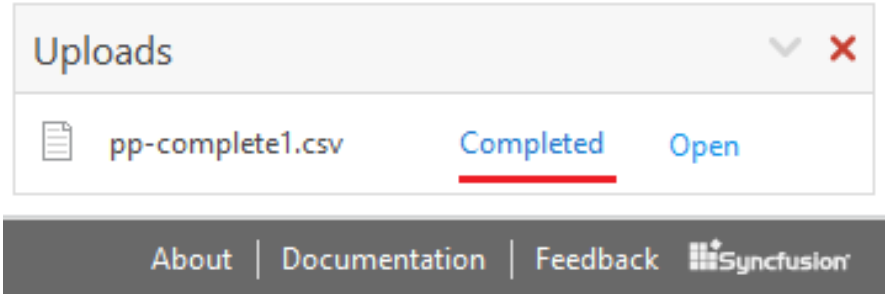
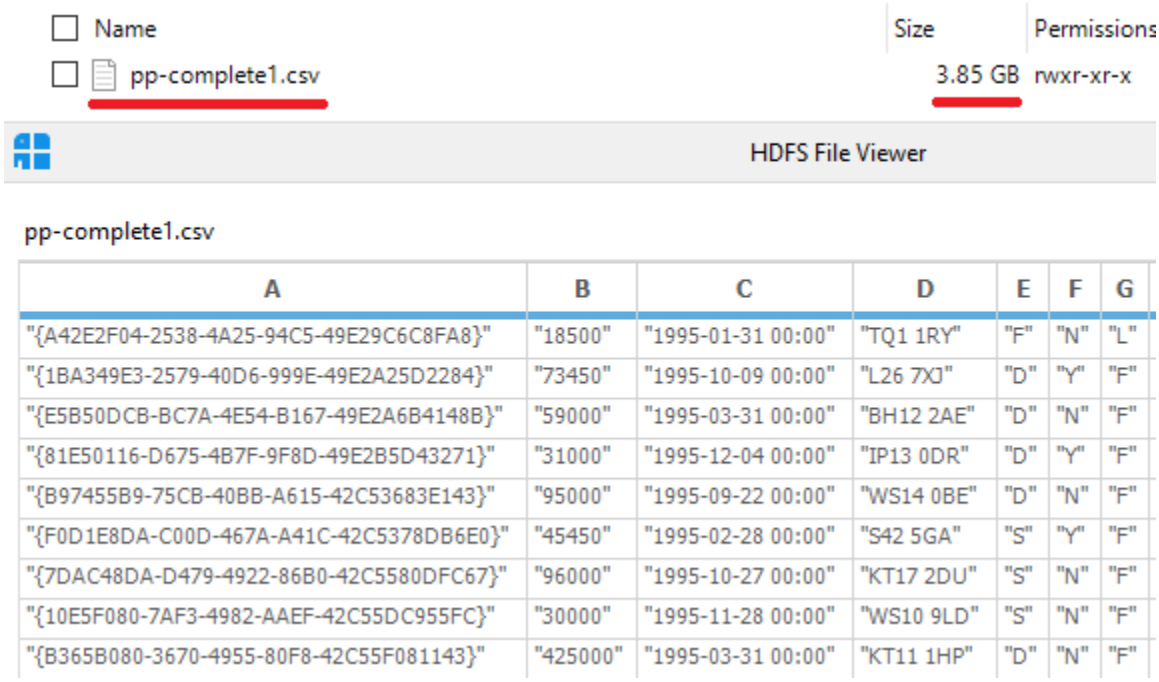


Figure 89: File upload complete

The file should not take long to upload on a fast system, but you could be waiting a few minutes on a slower one. Hadoop for Windows benefits from fast, interactive facilities that are unavailable in many Windows applications. For example, I could not even begin to load a file this size into Excel. In QlikView, you'd be using all sorts of built-in compression (QVDs) to load the file, and your system would certainly feel it. In databases, you'd be taking a while to upload it, and then have to wait for running queries to view outputs from it. In Hadoop for Windows, this file is trivial on a powerful system. Simply double-click the uploaded file **pp-complete1.csv,** or

right-click it and click **View**. The 4-GB file instantly opens in the HDFS File Viewer, which allows you to instantly go to any one of the 31,573 pages that make up the data file.



A	B	C	D	E	F	G
"{A42E2F04-2538-4A25-94C5-49E29C6C8FA8}"	"18500"	"1995-01-31 00:00"	"TQ1 1RY"	"F"	"N"	"L"
"{1BA349E3-2579-40D6-999E-49E2A25D2284}"	"73450"	"1995-10-09 00:00"	"L26 7XJ"	"D"	"Y"	"F"
"{E5B50DCB-BC7A-4E54-B167-49E2A6B4148B}"	"59000"	"1995-03-31 00:00"	"BH12 2AE"	"D"	"N"	"F"
"{81E50116-D675-4B7F-9F8D-49E2B5D43271}"	"31000"	"1995-12-04 00:00"	"IP13 0DR"	"D"	"Y"	"F"
"{B97455B9-75CB-40BB-A615-42C53683E143}"	"95000"	"1995-09-22 00:00"	"WS14 0BE"	"D"	"N"	"F"
"{F0D1E8DA-C00D-467A-A41C-42C5378DB6E0}"	"45450"	"1995-02-28 00:00"	"S42 5GA"	"S"	"Y"	"F"
"{7DAC48DA-D479-4922-86B0-42C5580DFC67}"	"96000"	"1995-10-27 00:00"	"KT17 2DU"	"S"	"N"	"F"
"{10E5F080-7AF3-4982-AAEF-42C55DC955FC}"	"30000"	"1995-11-28 00:00"	"WS10 9LD"	"S"	"N"	"F"
"{B365B080-3670-4955-80F8-42C55F081143}"	"425000"	"1995-03-31 00:00"	"KT11 1HP"	"D"	"N"	"F"

Figure 90: Instant viewing of 4-GB data fie in HDFS File Viewer

For these reasons, Hadoop for Windows is very useful for storing large files—you don't have to wait even a second to see what is contained in the large file you're viewing. Imagine quickly double-clicking on a 4-GB CSV file in Windows, and locking up your machine and whatever application tried to open it. In Hadoop for Windows you can have all your files, folders, and directories neatly ordered and instantly available to view. We will be using this 4-GB file a bit later in the book when we undertake some tasks of greater complexity.

We have ingested the data in HDFS, and we can store it and view it instantly in Windows. This is fine, but there is also a need to manipulate data once it's in HDFS. This is where the Hive data warehouse is used, along with other tools in the Hadoop ecosystem. We have to be able to use Hive in Windows to the same standard that Linux users use Hive in Linux.

Executing data-warehousing tasks using Hive Query Language over MapReduce

We're going to start by using lateral thinking, and show you something that appears to work, but doesn't really. It shows that the greatest strength of Hadoop can also be its greatest weakness, and how to avoid such pitfalls.

Create a new folder called **Hadoop4win** in the HDFS root folder, then upload the unzipped **title.akas.tsv.gz** IMDB file and name it **titleaka.tsv** in HDFS. Now, access the Hive data warehouse by selecting **Hive** from the menu in Big Data Studio.

www.dbooks.org

Code Listing 15

```
create external table IF NOT EXISTS Titleaka(titleId string,ordering
string,title string,region string,language string,types string,attributes
string,isOriginalTitle string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/Hadoop4win'
```

Enter the code from Code Listing 15 in the Hive editor window, and then click **Execute**. The Hive editor window and console window are shown in Figure 91.

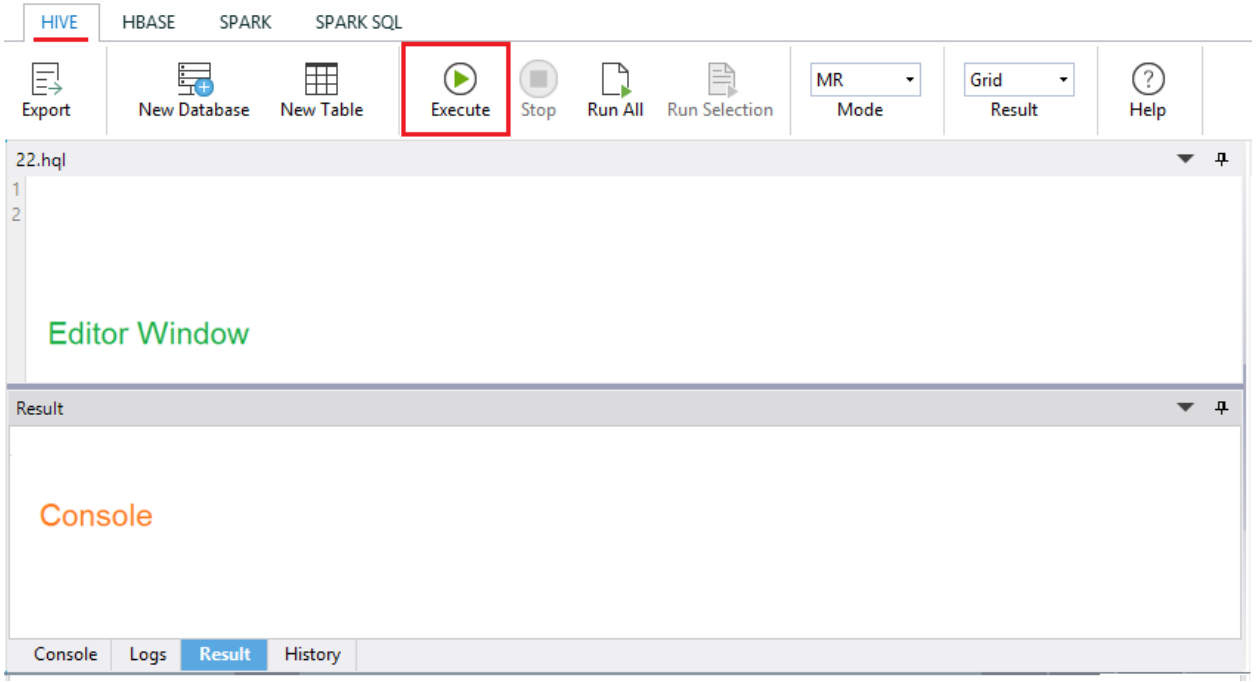


Figure 91: Hive Editor and Hive Console windows

The results are shown in the following figure.

Result

titleid	ordering	title	region	language	types	attributes
titleId	ordering	title	region	language	types	attributes
tt0000001	1	Carmencita - spanyol tÃ¡nc	HU	\N	imdbDisplay	\N
tt0000001	2	ÐšÐ°Ñ€Ð¼ÐµÐ½ÑÐ¸Ñ‚Ð°	RU	\N	\N	\N
tt0000001	3	Carmencita	US	\N	\N	\N
tt0000001	4	Carmencita	\N	\N	original	\N
tt0000002	1	Le clown et ses chiens	\N	\N	original	\N
tt0000002	2	A bohÃ³c Ã©s kutyÃ¡i	HU	\N	imdbDisplay	\N
tt0000002	3	Le clown et ses chiens	FR	\N	\N	\N

Figure 92: First results returned in Hive

We can remove the first row of data by using the code in Code Listing 16, and then clicking **Execute**.

Code Listing 16: Removing first row of data from table

```
create external table IF NOT EXISTS Titleaka(titleId string,ordering
string,title string,region string,language string,types string,attributes
string,isOriginalTitle string)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/Hadoop4win' tblproperties ("skip.header.line.count"="1")
```

The next figure shows the first row of data removed.

Result

titleid	ordering	title	region	language	types	attributes
tt0000001	1	Carmencita - spanyol tÃinc	HU	\N	imdbDisplay	\N
tt0000001	2	ĐšĐ°Ñ€Đ¼ĐµĐ½ĐĐ¸Ñ‚Đ°	RU	\N	\N	\N
tt0000001	3	Carmencita	US	\N	\N	\N
tt0000001	4	Carmencita	\N	\N	original	\N

Figure 93: First line of duplicated header text removed in Hive

While these results may look okay, try clicking on the menu item **Spark SQL**. You'll now be asked to start the Spark thrift server; click the button presented to start it. After it starts, you'll see a panel on the right-hand side of the screen; click **Databases** at the bottom of the panel, and you'll see a section at the top of the panel called DataBase. This is shown in Figure 93.

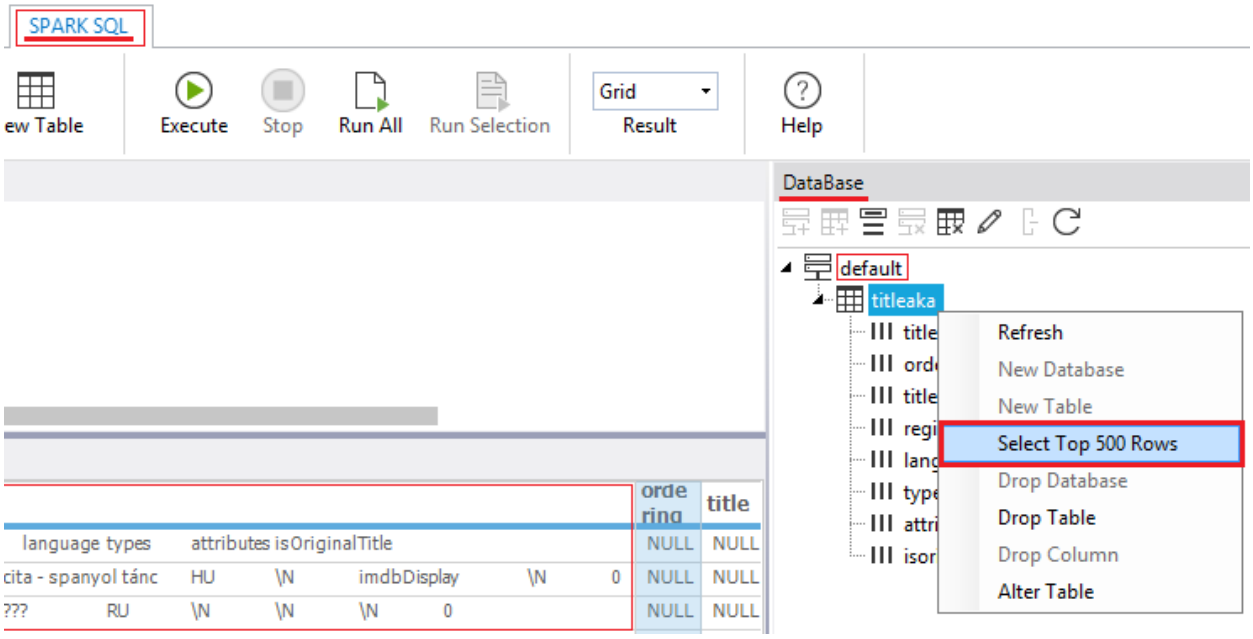


Figure 94: Spark SQL environment

www.dbooks.org

You then see a database called **default** and a database table called **titleaka**, as seen in Figure 94. It's the table we created in Hive that's now visible in Spark SQL. Right-click the **titleaka** table and click the option to **Select Top 500 Rows**. In the Spark SQL window, we can clearly see all the data was wrongly ingested into the **titleId** field; this is shown in Figure 95. The data looked correct in Hive, but it clearly isn't—imagine trying to use the table in a join with another table. As Hadoop can ingest so many different formats of data, there's a potential for errors in processing such a range of file formats.

Result

titleid										ordering	title	region
titleId	ordering	title	region	language	types	attributes	isOriginalTitle			NULL	NULL	NULL
tt0000001	1	Carmencita - spanyol tánc	HU	\N	imdbDisplay	\N	0			NULL	NULL	NULL
tt0000001	2	??????????	RU	\N	\N	\N	0			NULL	NULL	NULL
tt0000001	3	Carmencita	US	\N	\N	\N	0			NULL	NULL	NULL
tt0000001	4	Carmencita	\N	\N	original	\N	1			NULL	NULL	NULL
tt0000002	1	Le clown et ses chiens	\N	\N	original	\N	1			NULL	NULL	NULL
tt0000002	2	A bohóc és kutyái	HU	\N	imdbDisplay	\N	0			NULL	NULL	NULL

Figure 95: Data fault clearly highlighted in Spark

Hadoop in Windows includes the ability to ingest data in compressed format, as you can with Hadoop in Linux. We are going to upload the IMDB data files **title.episode.tsv.gz** and **title.ratings.tsv.gz** to HDFS in compressed form. After we upload the files to the **Hadoop4win** folder, rename the **title.ratings.tsv.gz** file to **title.rating.tsv.gz** by right-clicking it and choosing **Rename**. The files as seen in HDFS (Figure 96) are smaller, as they are compressed. It's good for storage, since we don't have to decompress the files and use more space. Even better, we can work off those compressed files as if they were in an uncompressed state.

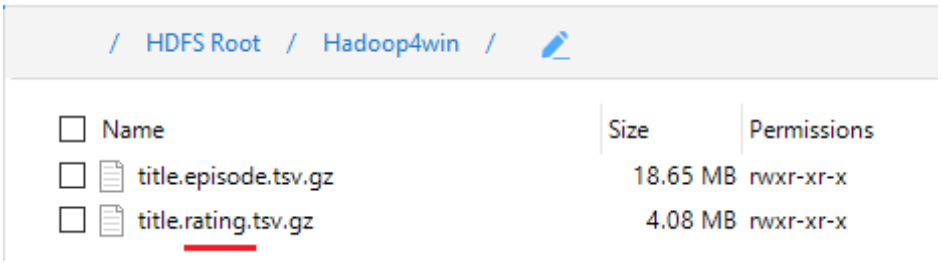


Figure 96: Uploaded compressed files in HDFS with title.ratings changed to title.rating

Enter the following code in the Hive editor window to create a table called **titleepisode** from the compressed **title.episode.tsv.gz** file.

Code Listing 17: Creating table in Hive from compressed file

```
CREATE TABLE titleepisode(tconst STRING,parentTconst STRING,seasonNumber
INT,episodeNumber INT)
row format delimited FIELDS terminated BY '\t' LINES TERMINATED by '\n'
stored AS textfile
tblproperties ("skip.header.line.count"="1");
```

```
Load data INPATH '/Hadoop4win/title.episode.tsv.gz' into table
titleepisode;
select * from titleepisode LIMIT 25;
```

Now, do the same to create a table called **titlerating** with the following code.

Code Listing 18: Code for creating second table from compressed file in Hive

```
CREATE TABLE titlerating(tconst STRING,averageRating int,numVotes int)
row format delimited FIELDS terminated BY '\t' LINES TERMINATED by '\n'
stored AS textfile
tblproperties ("skip.header.line.count"="1");
Load data INPATH '/Hadoop4win/title.rating.tsv.gz' into table titlerating;
select * from titlerating LIMIT 25;
```

The output in the next figure shows the output from the table created in the preceding code, namely **`select * from titlerating LIMIT 25.`** This selects the first 25 rows from the **titlerating** table we just created from the compressed file. This line is a good starting point for explaining the concept of MapReduce.

Result

tconst	averagerating	numvotes
tt0000001	5	1410
tt0000002	6	164
tt0000003	6	1001
tt0000004	6	99
tt0000005	6	1702
tt0000006	5	87
tt0000007	5	567
tt0000008	5	1515
tt0000009	5	68
tt0000010	6	5053

Figure 97: Rows returned from the select query

The query **`select * from titlerating LIMIT 25`** does not make use of either map or reduce. This is because the **`select *`**, which means select all, runs through and returns values from all the columns. There is no filtering in use, which is basically what the map element of MapReduce is. In addition, there's no grouping of text or value data, so there's no aggregation or summing of values. This means there is no reduction, either.

Let's put the two tables together in a join. Joins require tables to have individual columns filtered as columns to join tables on, so mapping is often a necessity. Resulting data from queries involving joins often have grouped elements, or indeed group aggregation, so reduction is also present in the query. This means that when using the Hive query language, certain queries will not quickly return your dataset. Instead, you will see map and reduce actions running before your dataset is returned. Hive is often criticized for slow-running queries involving joins, and this is not without merit. It's also why Impala has become so popular; we will look at Impala a bit later. That said, a data warehouse not capable of executing joins is no data warehouse at all.

www.dbooks.org

Let's join the tables after switching to Tez mode in Hive, which helps to speed up queries. Tez is not as fast as Impala by any means, but can make the difference between a query executing and a query failing.

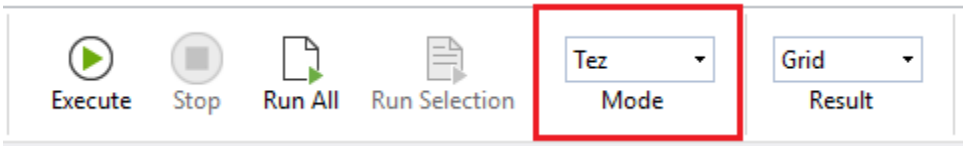


Figure 98: Switching to TEZ mode in Hive

This is achieved by clicking the **Mode** drop-down menu, as highlighted in Figure 98, and clicking **TEZ** instead of MR (MapReduce). Now, run the following code to join the two tables.

Code Listing 19: Code to join titlerating and titleepisode tables in Hive

```
SELECT te.tconst, te.seasonNumber, te.episodeNumber, tr.averageRating,
tr.numVotes
FROM titleepisode te JOIN titlerating tr ON (te.tconst = tr.tconst) LIMIT
25;
```

After running this code, you'll notice a long delay before your results are returned, as the map and reduce actions are carried out, as shown next.

```
Query ID = WIN-GE9UFN15GAA$_20190122140352_3c7908e3-1077-421d-b9c9-0d94
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1548

Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0/1    Reducer 2: 0/1
Map 1: 0/1    Map 3: 0(+1)/1      Reducer 2: 0/1
Map 1: 0(+1)/1      Map 3: 0(+1)/1      Reducer 2: 0/1
Map 1: 0(+1)/1      Map 3: 0(+1)/1      Reducer 2: 0/1
```

Figure 99: Map and Reduce actions in TEZ mode

Once the map and reduce actions are complete, the results of the joined tables are displayed.

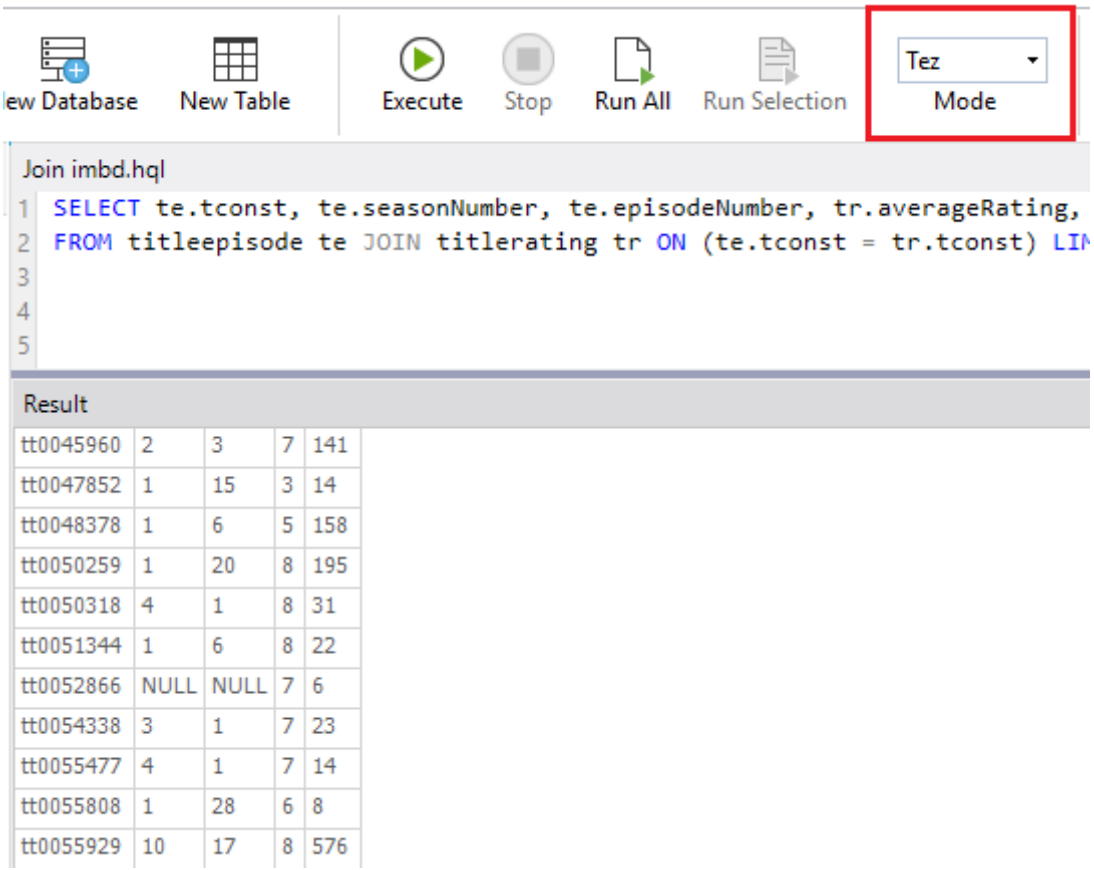


Figure 100: Query results from joining titlerating and titleepisode tables using TEZ and MapReduce

Utilizing Apache Pig and using Sqoop with external data sources

Compared to the complexities of MapReduce, Pig is seen as more straightforward and economical to use in terms of lines of code required to achieve a task. It follows that the less code there is, the easier it is to maintain. It's important to note that MapReduce is still utilized within Pig, but the more literal way of expressing code in Pig is favored by many. If there was a negative, it would be that while the Hive query language and SQL have clear similarities, Pig really is a language of its own. You may not have used a language that looks or works anything like it. The differences with Pig are not just in its form, but in its execution, as it doesn't need to be compiled.

To access Pig, simply click the **Pig** item from the menu in Big Data Studio. Create a folder within the Hadoop4win folder **called** titles, and unzip and upload the **title.basics.tsv.gz** file, and call it **titlebasic.tsv**. Now, enter the code from Code Listing 20 in the Pig editor window.

www.dbooks.org

Code Listing 20: Code for grouping data in Pig

```
--Load the titles data from its location

titlebasic = LOAD '/Hadoop4win/titles' using PigStorage('\t') as
(tconst,titleType,primaryTitle,originalTitle,isAdult,startYear,endYear,
runtimeMinutes,genres);

--Group the data by startYear
group1  = GROUP titlebasic BY startYear;
--Generate the Group alone
group2 = FOREACH group1 generate group;

--Display the Group
Dump group2;
```

After you enter the code in the editor, you will notice the code created in the console. An important line in that code is the URL to track the job, as highlighted next.

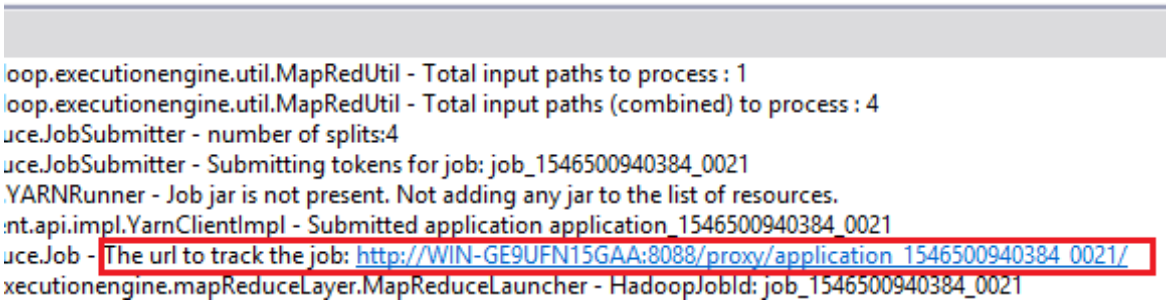


Figure 101: URL to track job in Pig

You simply click on the link shown, and you'll see an image similar to the one in Figure 102. The figure shows the same job as in the URL, which is **1546500940384_0021**. It shows that the job succeeded and the time, date, and elapsed time. It also shows the map, shuffle, merge, and reduce time.

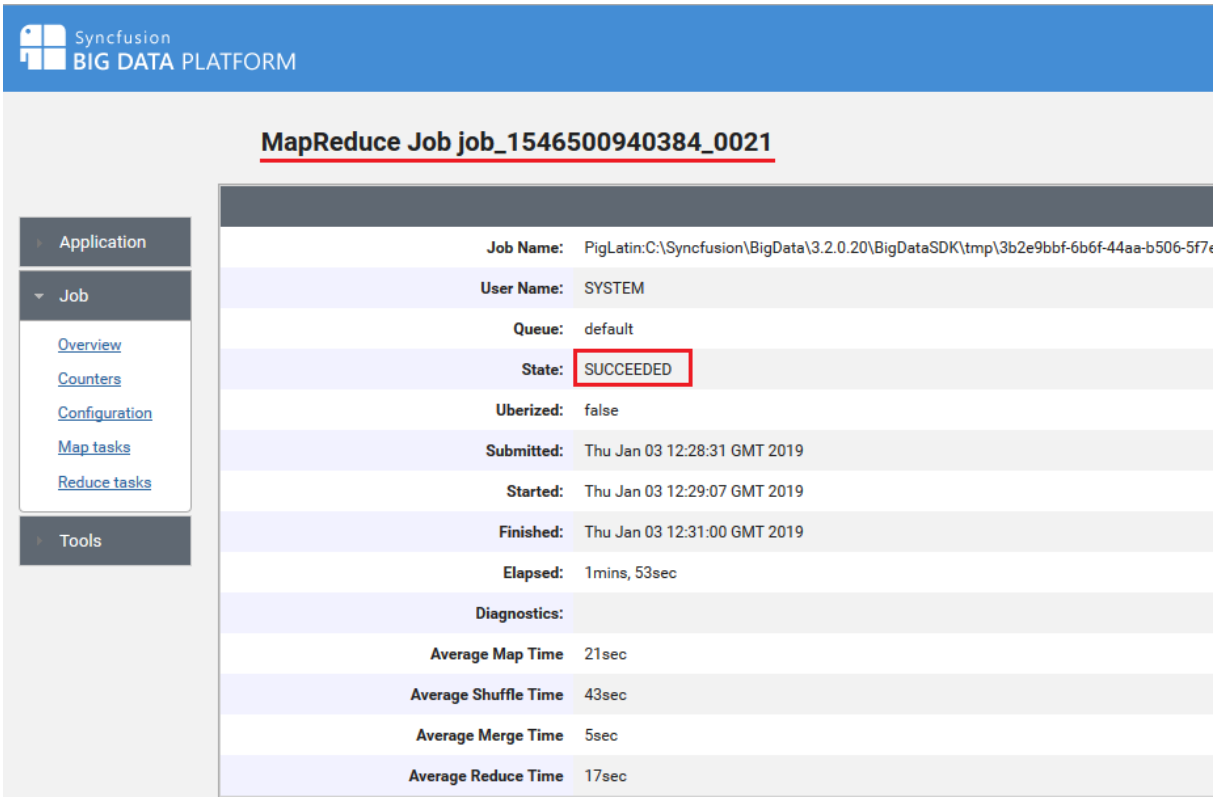


Figure 102: Tracking Pig MapReduce job

The requested result is also shown in the results window, which groups the individual start years of the films in the database.

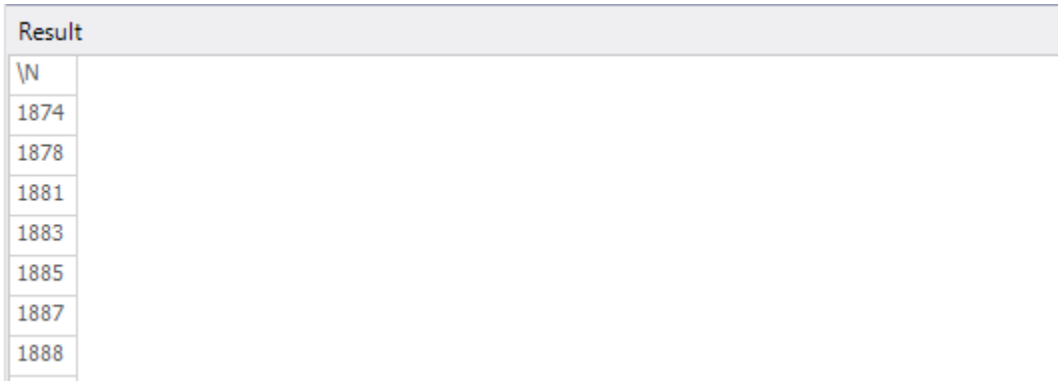


Figure 103: Returned results of Pig query showing the start years of films in the IMDB.

Some people have preferences for Hive, and others for Pig; it can depend on what exactly you're doing. Other people prefer to export their data out of Hadoop to work with relational database systems. This can be because they're much faster when working with joins between tables than Hadoop, for example. There's no real difference when using Pig in Windows to using Pig with Hadoop in Linux. It's therefore a good time to move onto Sqoop, which can import and export data to and from Hadoop. This is a key feature of Hadoop, and is available in Hadoop for Windows.

www.dbooks.org

Sqoop

Because of current advances in integrating Hadoop in Windows, this section nearly failed to make the book. It would though be wrong to exclude it though, since it is used by so many people in Hadoop to import and export data. To access Sqoop, simply click **Sqoop** on the Big Data Studio menu. You'll notice in the following figure that the JDBC connectors are not installed. If you are able to go online, click the checkbox to select the **Microsoft SQL Server JDBC Connector**, and the highlighted Install button will become enabled. Click **Install** if you choose to install the driver in this fashion, or click the link highlighted in Figure 104 to get the instructions to install the connector jars manually.

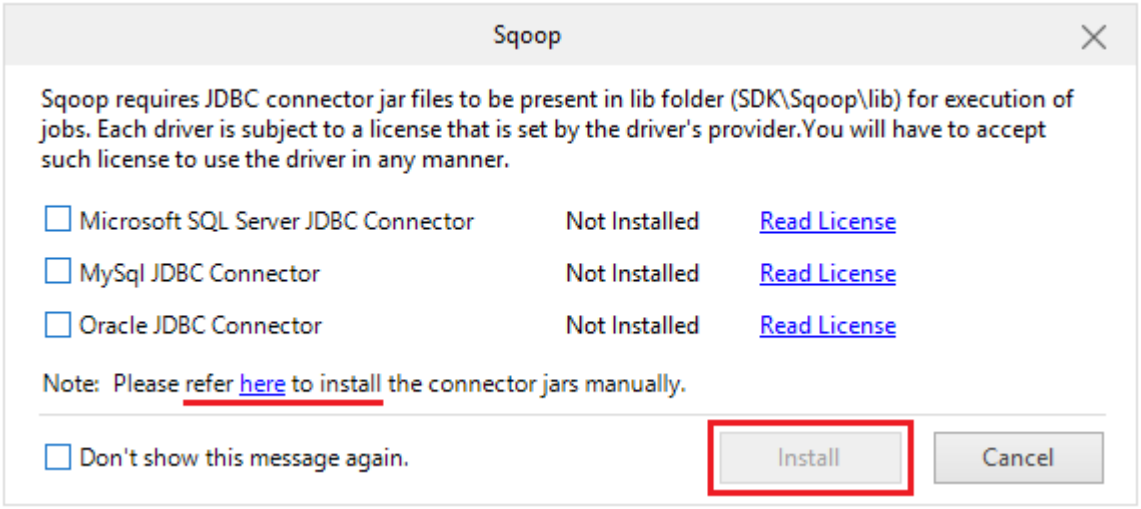


Figure 104: JDBC connector is not installed

As I am working with SQL Server v-next CTP 2.0 2019 Technology preview, I needed to download a connector that would work with it. So if you download sqljdbc_6.0.8112.200_enu, sqljdbc_7.0.0.0_enu or any other version it depends on the version of SQL Server you are using. You need to visit the Microsoft.com website to accurately determine this. One you extract the files you take the jar file which in my case is sqljdbc42 and place it in the Sqoop\lib folder. So in a multi-node cluster environment you place that file on each node where Sqoop is installed. The directory you place it in should be:

<*InstalledDirectory*>\Syncfusion\HadoopNode\<*Version*>\BigDataSDK\SDK\Sqoop\lib

If you are working on a local development cluster, the directory you should place it in is:

<*InstalledDirectory*>\Syncfusion\BigData\<*Version*>\BigDataSDK\SDK\Sqoop\lib

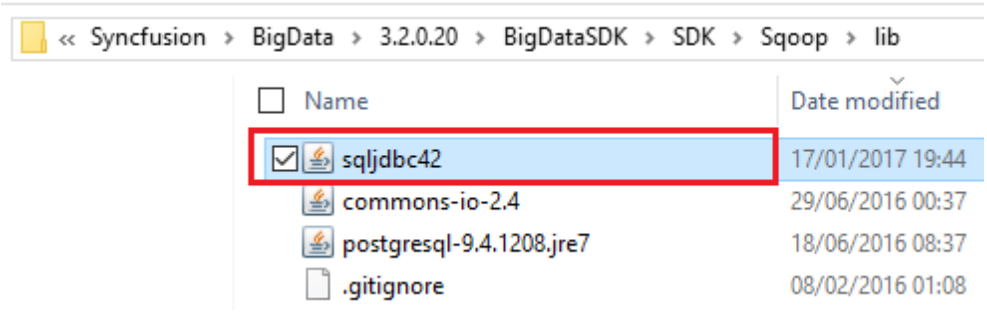


Figure 105: Connector Jar in the Sqoop\lib folder.

After you put the file in the relevant folder, close the Big Data Studio, and then open it again. The JDBC connector for SQL Server is now installed, as shown in Figure 106.

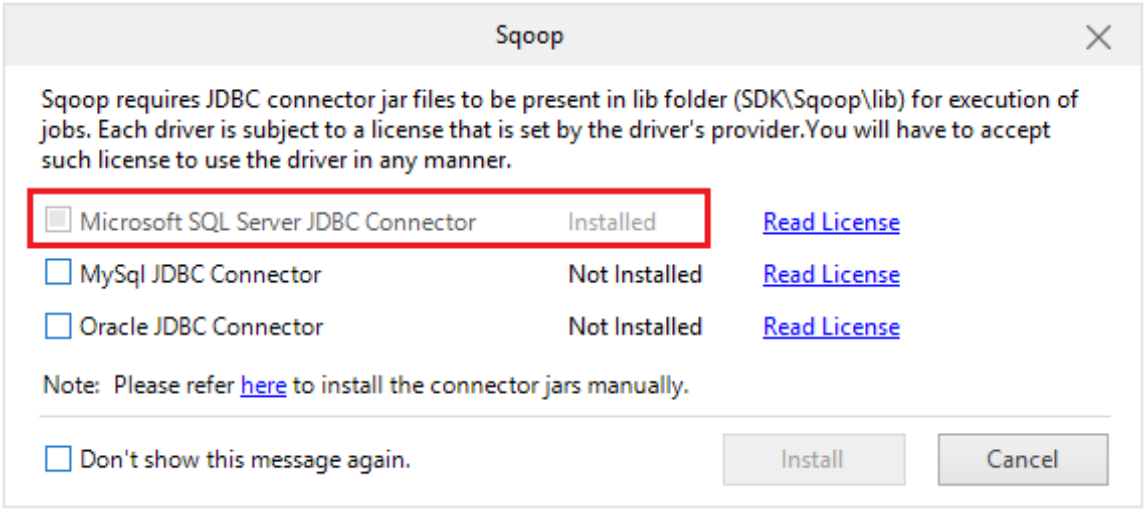


Figure 106: SQL Server JDBC Connector is now installed.

Click **Add Connection**, as shown in Figure 107, and insert the details for your own SQL Server system. Choose the title of the connector, the connector for SQL Server, and the connection string for your server. I named the title for my connector **SqoopMov02**. Then put in the username and password for your SQL Server account; I'm using the SQL Server systems administrator account I set up in SQL Server. Next, click **Save**.

www.dbooks.org

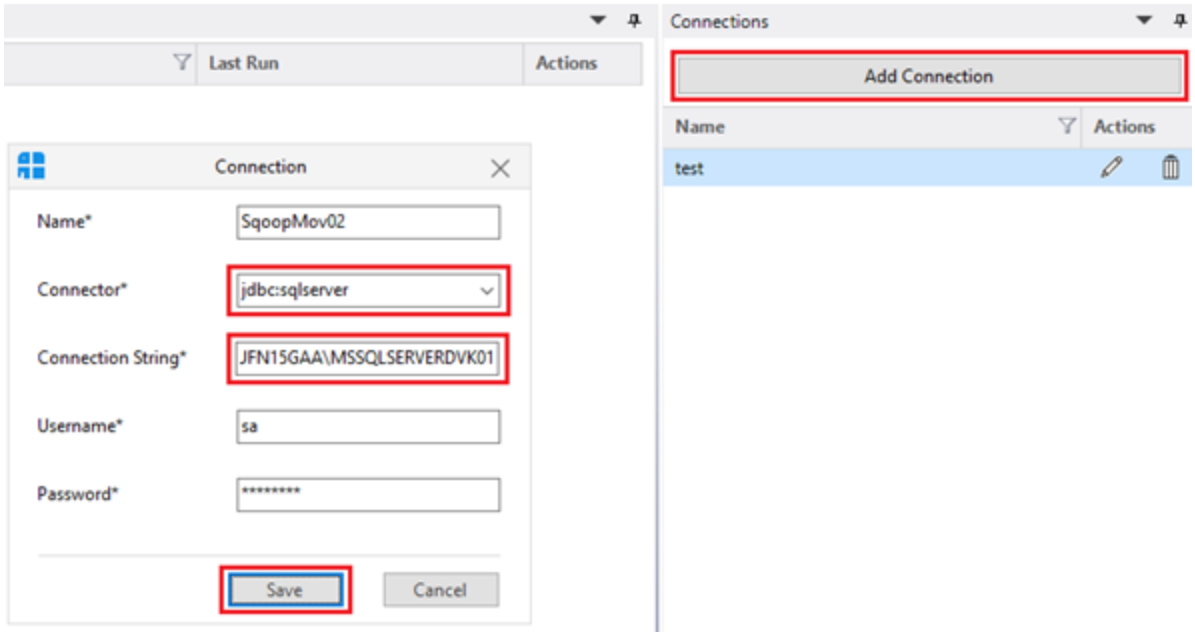


Figure 107: Creating a connection to SQL Server in Sqoop.

After saving the connection click **Add Job** and name the job **CurrencyImport**. Choose the connection **SqoopMov02** from the drop-down box, as shown in Figure 108.

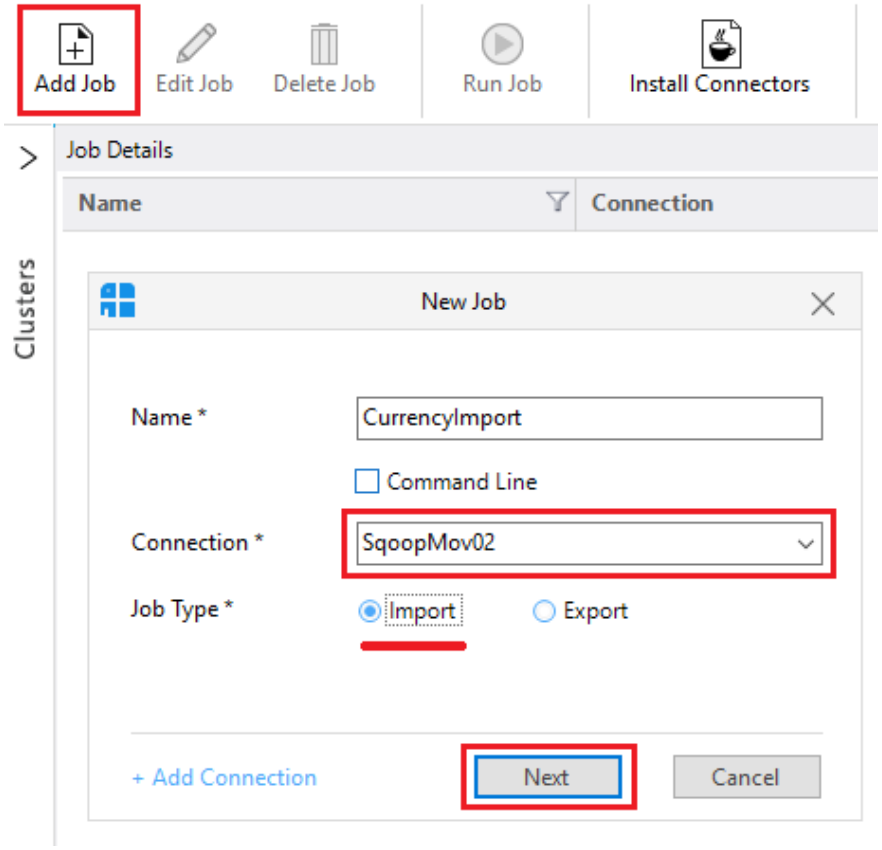


Figure 108: Adding new job in Sqoop

Since we're doing an import, click the **Import** radio button, then click **Next**. Now we will enter the details of the database and the table in SQL Server that we wish to import data from, as shown next. The database in SQL Server is called **INTERN28**, and the table is called **Currencies**. Click **Next** to continue.

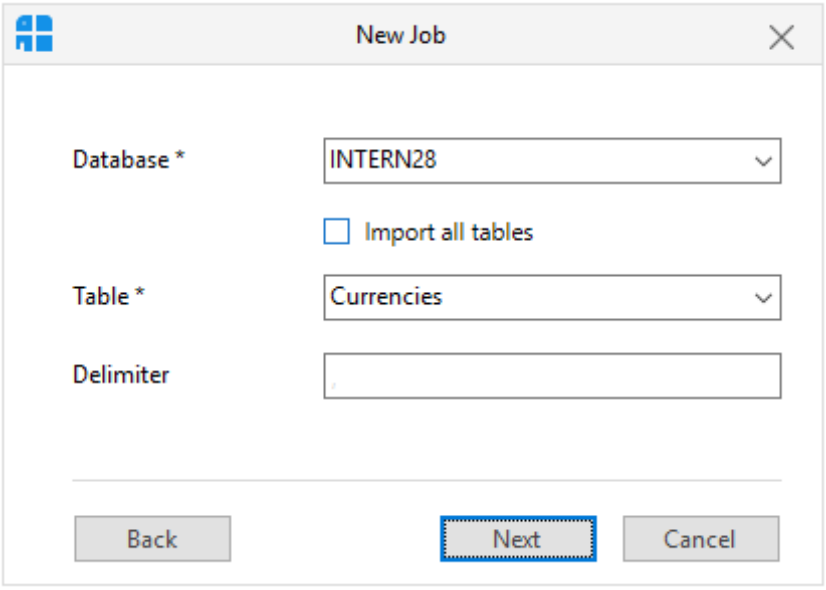


Figure 109: Identifying database and table for data import

We now need enter the HDFS directory that we want the data imported into; in my case it's **/Hadoop4win/data**. Now, click the **Save & Run** button.

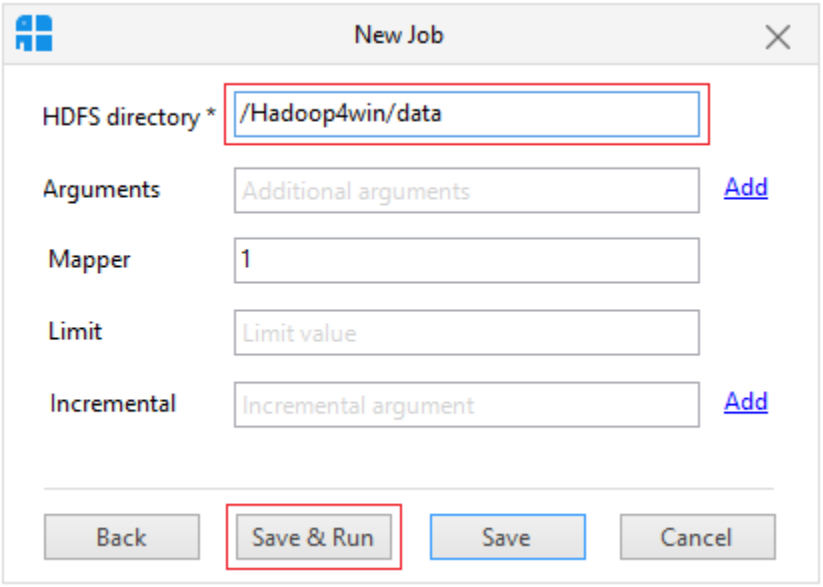


Figure 110: Save and run import job in Sqoop

You now see that the job is accepted, as reflected in the following figure.

www.dbooks.org

Connection	Type	Status
SqoopMov02	Import	ACCEPTED

Figure 111: Accepted Sqoop job

The status of the job then changes from **Accepted** to **Succeeded**.

Connection	Type	Status
SqoopMov02	Import	SUCCEEDED

Figure 112: Successfully completed Sqoop job

We now need to go the directory where we wanted the data imported to called /Hadoop4win/data. There you see the _**Success** notification, and underneath that, you see the table that has been imported, referenced **part-m-00000**. Double-click on it, and you see data we imported via the HDFS File Viewer. Notice that the file arrived first at 22:56:49, and then the success notification arrived a second later, at 22:56:50.

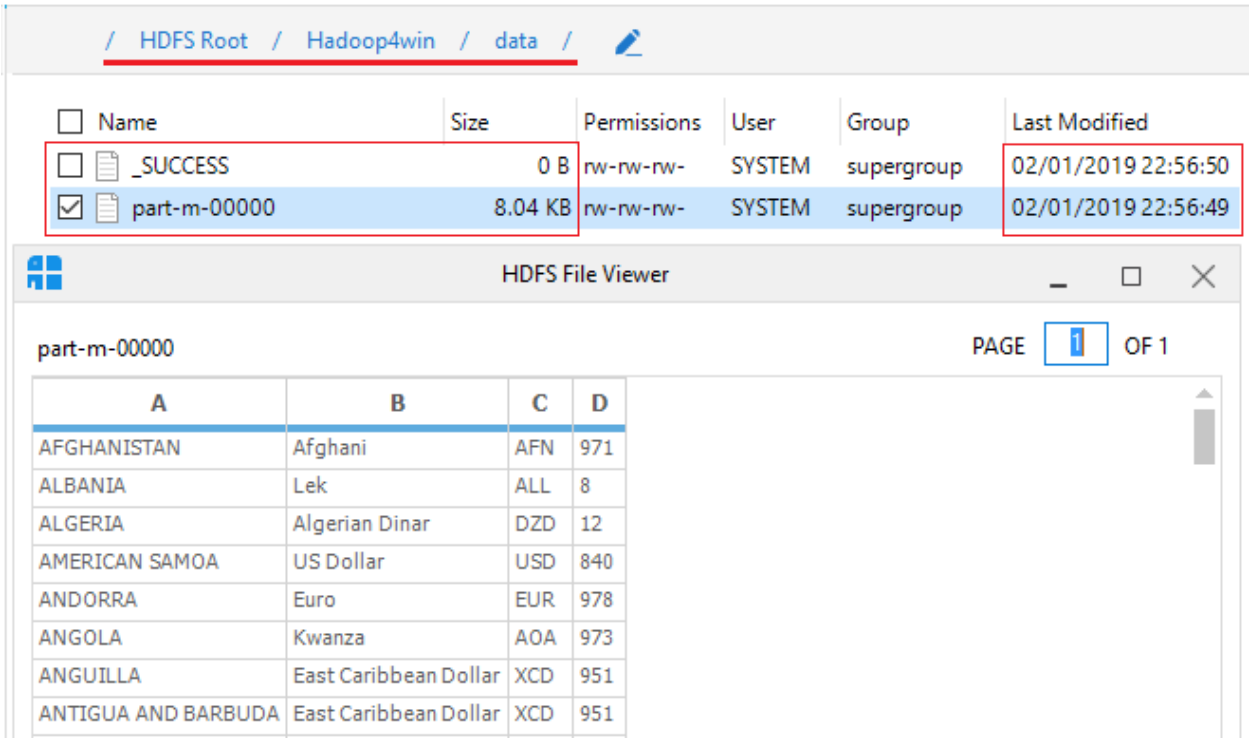


Figure 113: Sqoop data arrived successfully

Now let's do a Sqoop export; we'll use a file that's installed when you install the Syncfusion Big Data Studio. It's the Customers.csv file in the Customers folder, as shown in Figure 114.

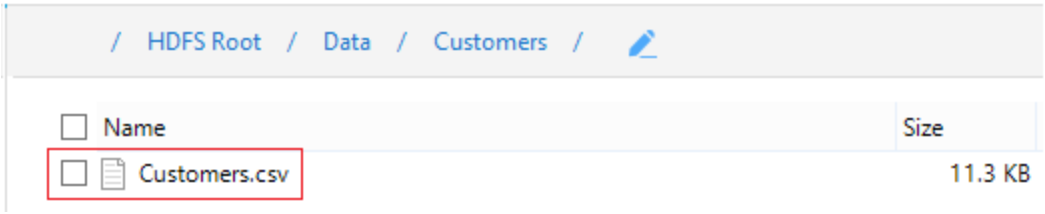


Figure 114: Customers csv file that is installed with the software

We'll add a job as we did with the data import, but this time we choose the **Export** option. We'll use the same **SqoopMov02** connection as we used for the import, as shown in the following figure. Now, click **Next**.

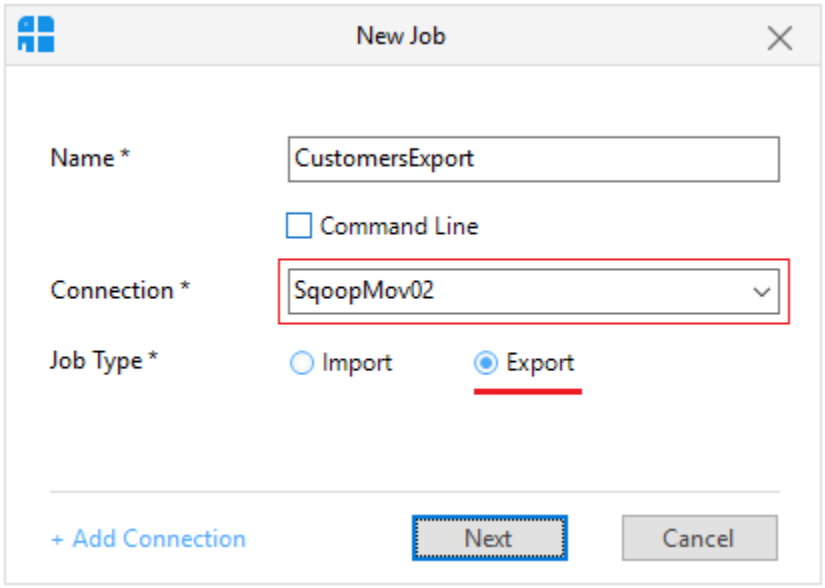


Figure 115: Starting an export job in Sqoop

We'll use the button highlighted in Figure 116 to choose the folder we wish to export data from; here, it's **Data/Customers/.** Click **Next**.

www.dbooks.org

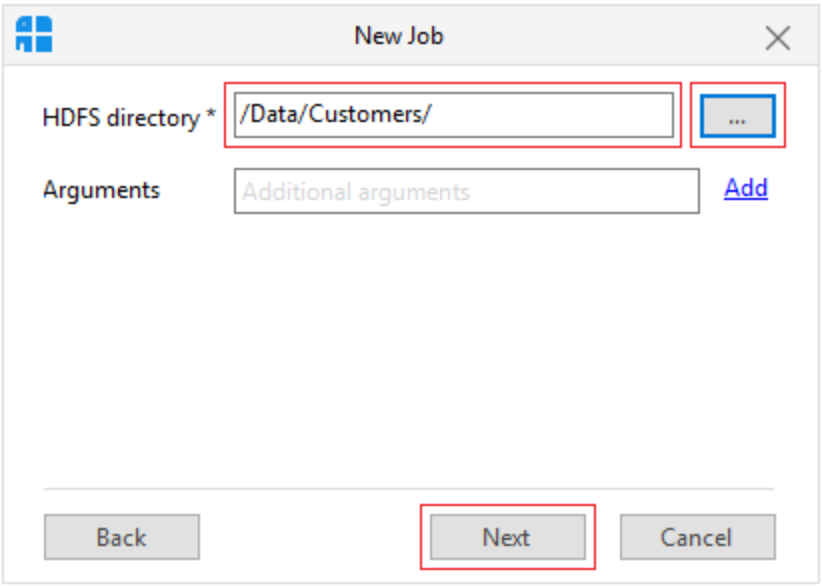


Figure 116: Specifying folder to export data from

We now enter the name of the database we want to export the data to, and the name of the table we want to export the data to, and click **Save & Run**.

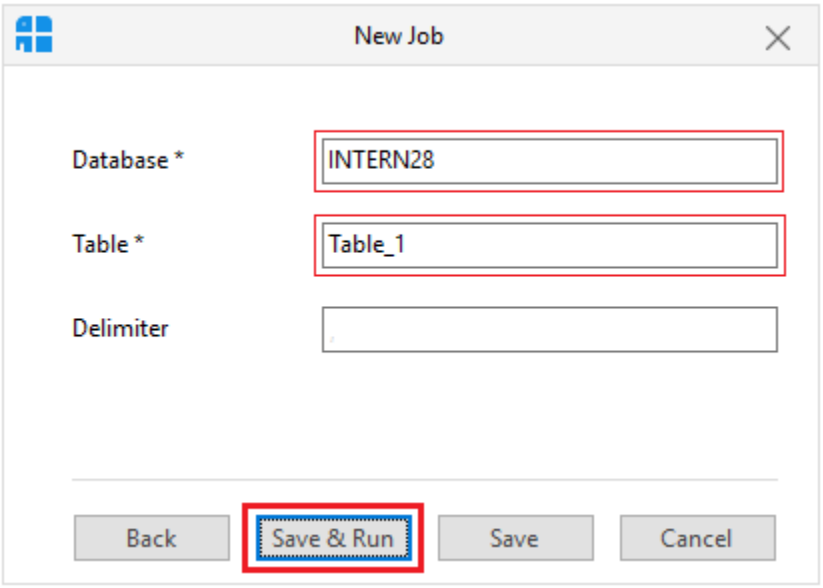


Figure 117: Save and run an export job in Sqoop

We now see the Succeeded status of the CustomerExport job shown under the previous successful CurrencyImport job. It shows that both jobs used the same Connection, as seen in Figure 118.

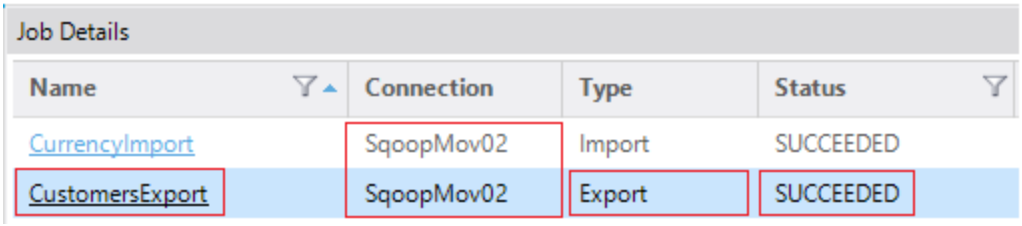


Figure 118: Sqoop export job succeeded

You should now also see the data exported from Sqoop in SQL Server, as displayed in the following figure. The customers.csv file data has arrived in the INTERN28 database table called Table_1.

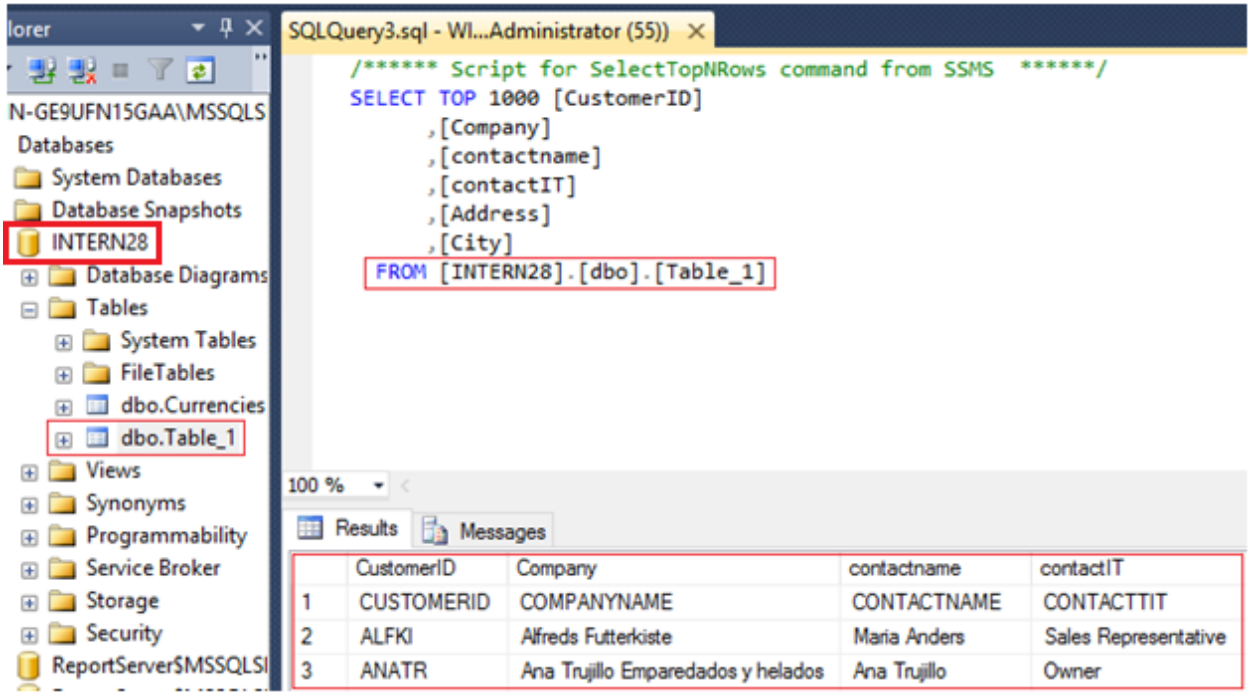


Figure 119: Exported table from Syncfusion Hadoop distribution arrived in SQL Server

While Sqoop can import and export data, I can't say it's the smoothest or most efficient tool I've ever seen. This is nothing to do with Windows or Linux; I've just never felt it's an impressive tool. Fortunately, Microsoft is making real progress at integrating Hadoop with Windows, and this is providing alternatives. We will look at this in more depth in Chapter 4.

Summary

We started by presenting the capabilities of Windows Server as a perfect partner for Hadoop. Its ability to utilize 24 terabytes of RAM and 256 processors allow it to scale to any task that Hadoop can throw at it. We then used Amdhal's Law to examine how Hadoop transports data across a network, and the mechanisms Hadoop uses to store and access data on disk.

www.dbooks.org

We examined the system resources required by Hadoop in Windows Server, before looking at Hive data types and the Hive data manipulation language. We prepared Hadoop to ingest data as we began to upload files, create tables, and manipulate data in the Hadoop ecosystem. We manipulated compressed data files and executed table joins in Hive before carrying out jobs in Pig and setting up connections in Sqoop. This included loading external SQL Server drivers and focusing on the role of MapReduce in querying data in Hive and Pig.

We concluded by creating both import and export jobs between SQL Server and Sqoop after setting up Sqoop connections for data transfer. I think we can safely say that Hadoop and its ecosystem run perfectly on Windows Server—we are not missing out on anything in Windows that is available in Linux.

When we go one step further—to connect to and report from Hadoop—we see the advantages of the Windows environment over Linux. If you had asked me about this even a short while ago, I would not have agreed. Luckily, the release of the SQL Server v-next CTP 2.0 Technology Preview has changed all that.

Chapter 4 Hadoop Integration and Business Intelligence (BI) Tools in Windows

Hadoop Integration in Windows and SQL Server 2019 CTP 2.0

SQL Server 2019 v-next CTP 2.0 has the ability to read the native Hadoop Distributed File System (HDFS) in Microsoft Windows. Earlier aspirations of integrating Hadoop in Windows are now robust operational functionality. The SQL Server PolyBase Java Connector provides fully integrated querying of the HDFS, using T-SQL. Further enhancements include data virtualization, whereby the exporting of data from Hadoop to SQL Server is eliminated. Instead, you read the data live from the HDFS using SQL Server in Windows. In SQL Server 2019, fully integrated querying of Teradata and other relational and non-relational systems are also available.

PolyBase, which is at the heart of the preceding developments, supports Hortonworks Data Platform (HDP) and Cloudera Distributed Hadoop (CDH). I have tested the PolyBase functionality with the Syncfusion Hadoop distribution and can confirm it works with the Syncfusion platform in Windows. Please bear in mind that SQL Server 2019 v-next is a technology preview, so if you choose to follow along in this section, you do so at your own risk.

What do these developments mean for ETL and tools like Sqoop? Why spend time importing data if you don't need to? Microsoft has gone further with Hadoop integration in Windows and introduced big data clusters. Steps to install and configure them may seem complex at first, but you can quickly get used to them.

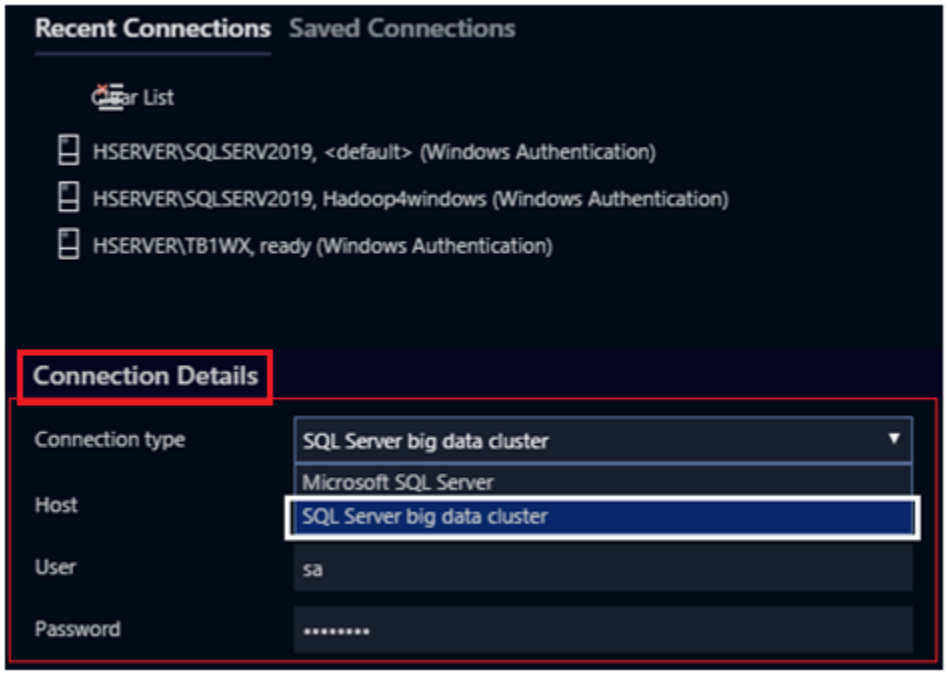


Figure 120: Accessing SQL Server Big Data Clusters using Azure Data Studio

www.dbooks.org

Integrating Hadoop into Windows has implications for Windows BI users connecting to Hadoop data in Linux. Can the Hadoop integration used in SQL Server be taken advantage of by third-party BI tools in Windows? If the answer is “yes,” it provides an alternative for Windows users connecting Windows BI tools to Hadoop in Linux. For those who think SQL Server is not significant to Hadoop or Linux users, I'll address this directly.

The most successful SQL Server is SQL Server 2017 for Linux—it had over 7,000,000 downloads between October 2017 and September 2018.

Let's have a look at SQL Server 2019 v-next CTP 2; please bear in mind that it's a technology preview. You can download the .ISO file from www.microcrosft.com. When installing, make sure you choose the highlighted items shown in Figure 121. Ensure that the **PolyBase Query Service for External Data** and the **Integration Services** options are selected to install.

Please remember that, unlike older versions of SQL Server, SSMS (SQL Server Management Studio) is not installed during the installation—you have to install it separately yourself. SSMS v17.5 is available free from Microsoft, but includes no database engine.

We'll be connecting SQL Server 2019 with the Syncfusion Hadoop distribution we used previously. You don't need to install the machine learning options selected in the following figure; that was simply my preference. If you do choose to install them, you have to download them separately from a link provided during installation. Like SSMS, they are not included in the SQL Server 2019 installer, and the combined .cab files are quite large. After installation is complete, create a database in SQL Server called **Hadoop4windows**.

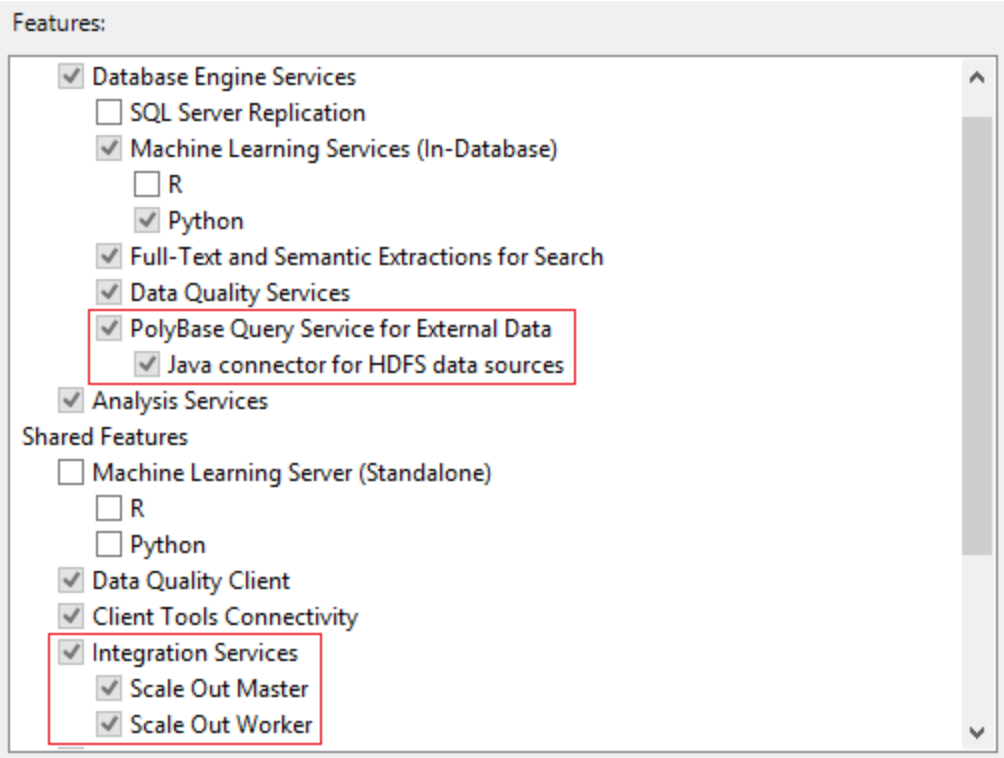


Figure 121: Installing SQL Server v-next 2019 CTP 2.0

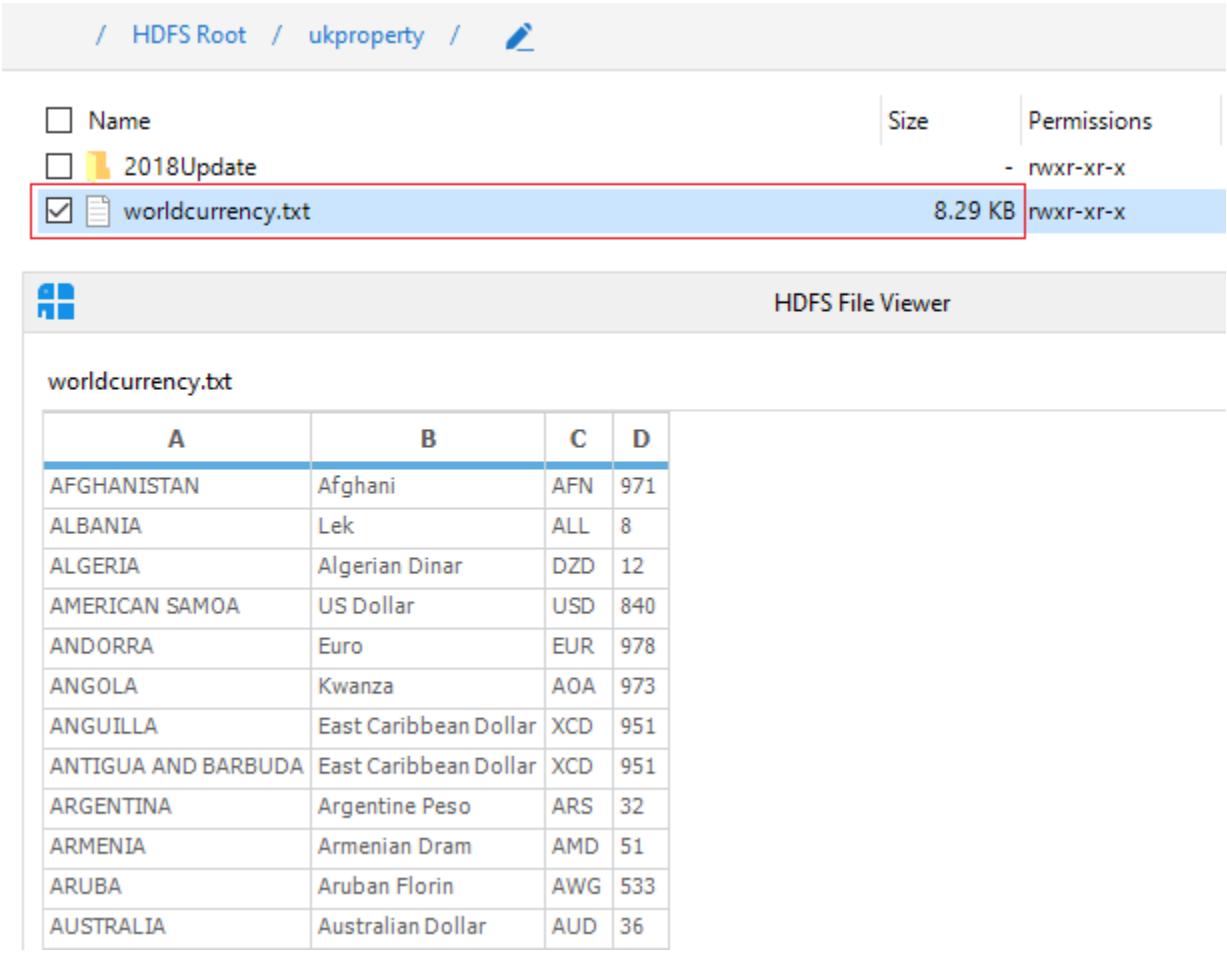
Even if PolyBase is installed, you have to enable it for Syncfusion with the following code.

Code Listing 21: Enabling PolyBase for Syncfusion Hadoop distribution in SQL Server 2019

```
-- To enable the PolyBase feature
exec sp_configure @configname = 'PolyBase enabled', @configvalue = 1;
RECONFIGURE WITH OVERRIDE;

-- To set Hadoop connectivity value to 7 for Syncfusion compatibility
sp_configure @configname = 'hadoop connectivity', @configvalue = 7;
GO
RECONFIGURE;
-- You MUST RESTART SQL SERVER after entering the preceding code.
```

Imagine you have a Hadoop currency file that's updated as different jurisdictions trade in more and more currencies. You don't have to import it anymore; you can run it live in SQL Server. To achieve this, we'll upload **worldcurrency.txt** to a folder we create in Hadoop called **ukproperty**.



A	B	C	D
AFGHANISTAN	Afghani	AFN	971
ALBANIA	Lek	ALL	8
ALGERIA	Algerian Dinar	DZD	12
AMERICAN SAMOA	US Dollar	USD	840
ANDORRA	Euro	EUR	978
ANGOLA	Kwanza	AOA	973
ANGUILLA	East Caribbean Dollar	XCD	951
ANTIGUA AND BARBUDA	East Caribbean Dollar	XCD	951
ARGENTINA	Argentine Peso	ARS	32
ARMENIA	Armenian Dram	AMD	51
ARUBA	Aruban Florin	AWG	533
AUSTRALIA	Australian Dollar	AUD	36

Figure 122: World currency text file in Hadoop

To access Hadoop data sources, we need to write some code in the query window in SQL Server. Connect to the ***Hadoop4windows*** database we created, and enter the code in Code Listing 22. This sets up external data sources, file formats, and tables in SQL Server.

www.dbooks.org

Code Listing 22: Code to set up live connection to HDFS from SQL Server 2019

```
-- CREATE EXTERNAL DATA SOURCE

USE Hadoop4windows
GO

CREATE EXTERNAL DATA SOURCE hadoop_4_windows WITH
(
    TYPE = HADOOP,
    LOCATION ='hdfs://127.0.0.1:9000'
)
GO

-- CREATE EXTERNAL FILE FORMAT

USE Hadoop4windows
GO

CREATE EXTERNAL FILE FORMAT TextFileFormat WITH (
          FORMAT_TYPE =  DELIMITEDTEXT,
          FORMAT_OPTIONS

           (

           FIELD_TERMINATOR =',',
           USE_TYPE_DEFAULT = TRUE));

-- CREATE EXTERNAL TABLE

USE Hadoop4windows
GO

CREATE EXTERNAL TABLE [dbo].[worldcurrency] (
          [country] nvarchar(200) NOT NULL,
          [currency] nvarchar(100) NOT NULL,
          [alphabeticcode] nvarchar (100) NOT NULL,
          [numericcode] nvarchar(100) NOT NULL)


    WITH (LOCATION='/ukproperty/',
          DATA_SOURCE = hadoop_4_windows,
          FILE_FORMAT = TextFileFormat
          );
```

Once your code has run, you should see the highlighted changes in SQL Server Object Explorer. Refresh or restart your server if you don't see them initially.

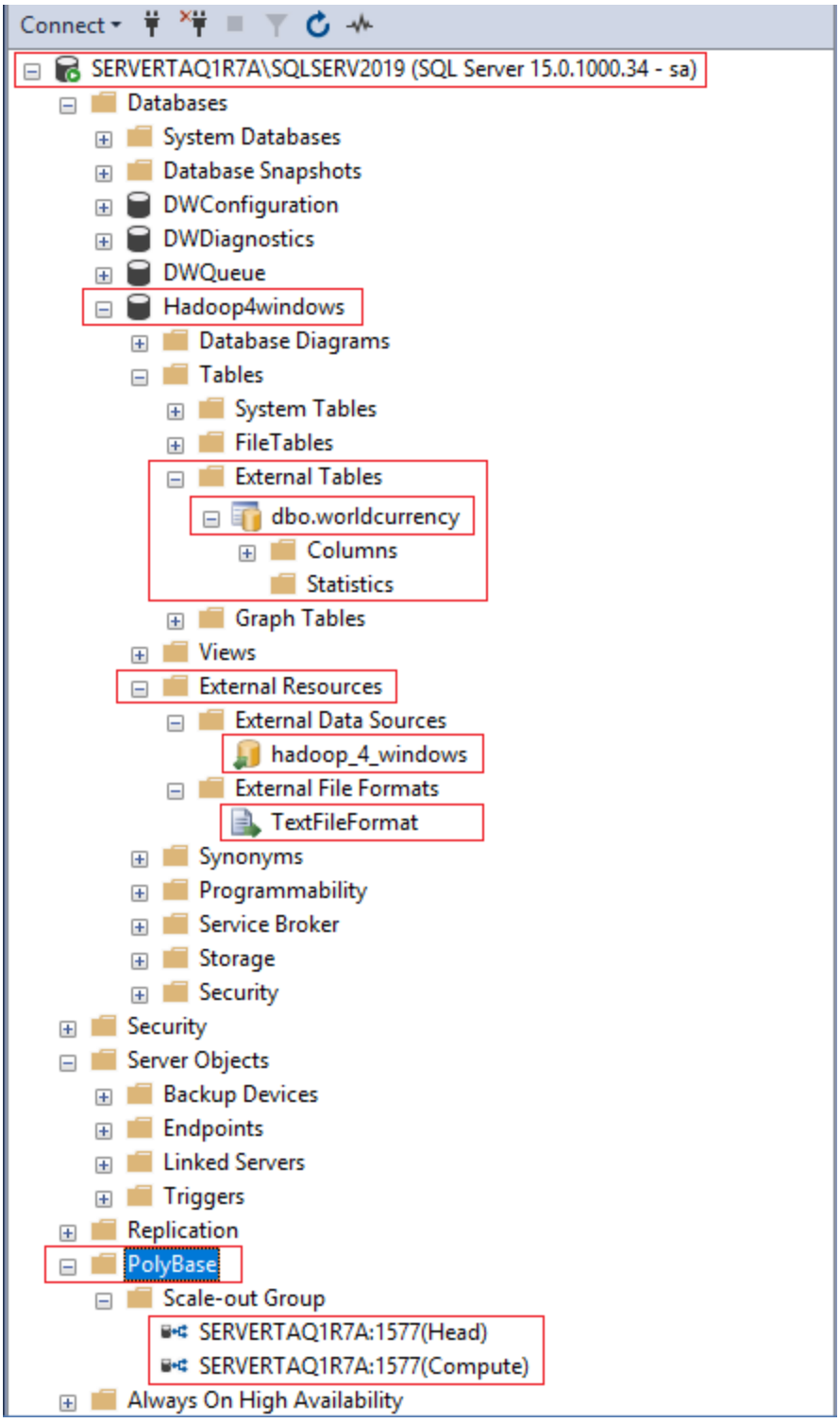


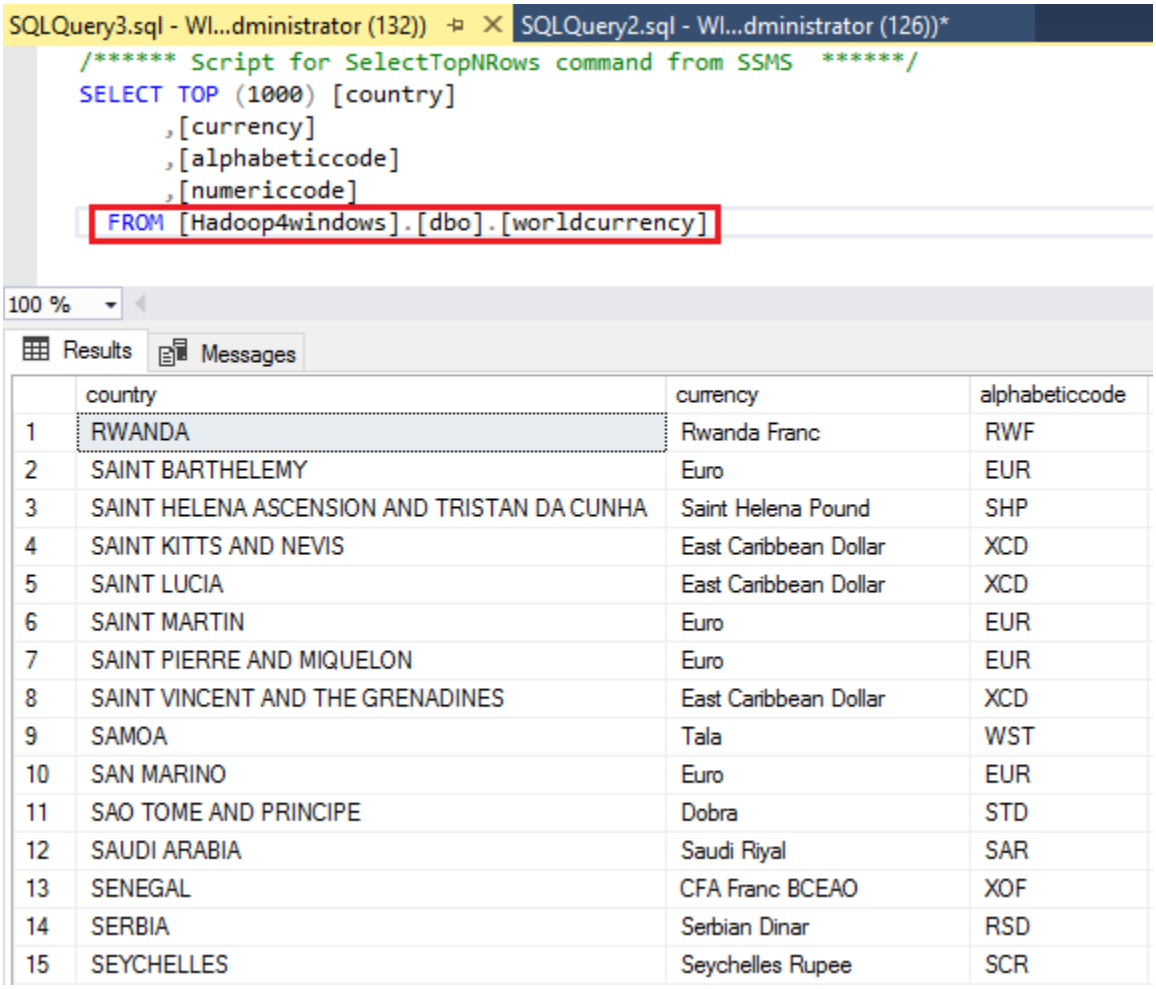
Figure 123: PolyBase and external resources enabled in SQL Server 2019

Note that you can see the **dbo.worldcurrency** table containing data from the **worldcurrency.txt** file in Hadoop under the External Tables section shown in Figure 123. It's a live connection, and you can query it live as you would any other table in SQL Server 2019. Right-click the table in SQL Server and select the built-in **Select Top 1000 Rows** query to see

www.dbooks.org

the output in the next figure. As SQL Server 2019, CTP 2.0 can read the items inside HDFS; you don't even have to create a table in Hadoop. With this way of working, you don't have to export updates to SQL Server; you just read the data live. Of course, you can still keep storing historic copies in Hadoop.

Use it like any SQL Server table; it's as fast as other tables when involved in joins, as opposed to the inherently slow joins in Hadoop.



	country	currency	alphabeticcode
1	RWANDA	Rwanda Franc	RWF
2	SAINT BARTHELEMY	Euro	EUR
3	SAINT HELENA ASCENSION AND TRISTAN DA CUNHA	Saint Helena Pound	SHP
4	SAINT KITTS AND NEVIS	East Caribbean Dollar	XCD
5	SAINT LUCIA	East Caribbean Dollar	XCD
6	SAINT MARTIN	Euro	EUR
7	SAINT PIERRE AND MIQUELON	Euro	EUR
8	SAINT VINCENT AND THE GRENADINES	East Caribbean Dollar	XCD
9	SAMOA	Tala	WST
10	SAN MARINO	Euro	EUR
11	SAO TOME AND PRINCIPE	Dobra	STD
12	SAUDI ARABIA	Saudi Riyal	SAR
13	SENEGAL	CFA Franc BCEAO	XOF
14	SERBIA	Serbian Dinar	RSD
15	SEYCHELLES	Seychelles Rupee	SCR

Figure 124: Running live query against file in HDFS from SQL Server 2019 v-next CTP 2.0

You can also see the PolyBase group of objects in the Object Explorer in SQL Server, if you right-click the **Scale-out Group** and click the option **Configure PolyBase Scale-out Group**, you'll see what's shown in Figure 125. This shows the PolyBase scale-out cluster instance head node, which is displaying **Scale-out cluster server ready**. You need to be running SQL Server Enterprise to designate a head node. You then set up compute nodes as desired to create a cluster.

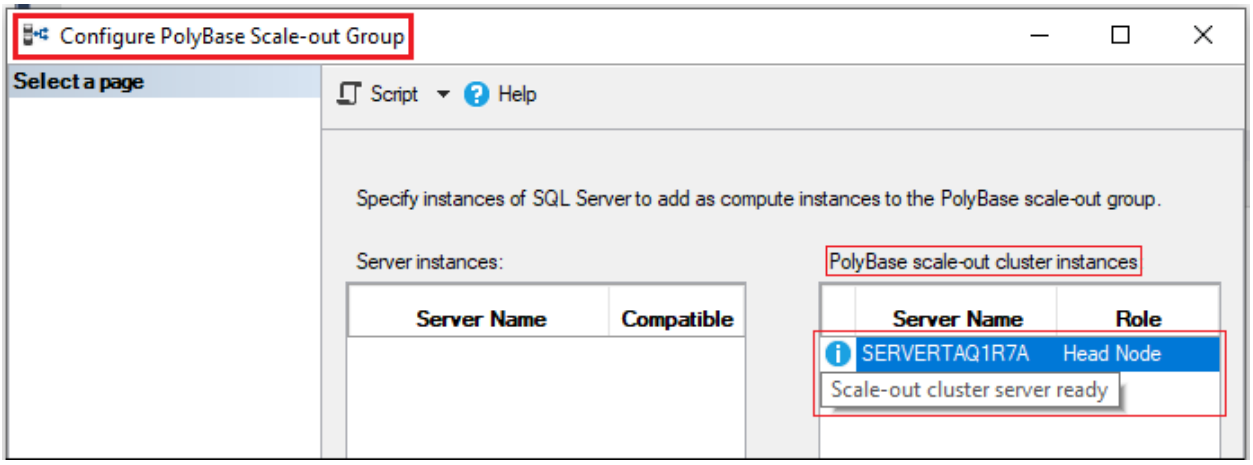


Figure 125: PolyBase Scale-out group cluster instance

If you wanted to add clustering, you'd require multiple servers and a domain name server. You would then install instances of SQL Server on each server node. The next task is to enable PolyBase on each server and designate head nodes and worker nodes. If you click the highlighted **+** icon in the **Configure PolyBase Scale-out Group** window, you can invite another SQL Server 2019 instance into the scale-out group, as shown next.

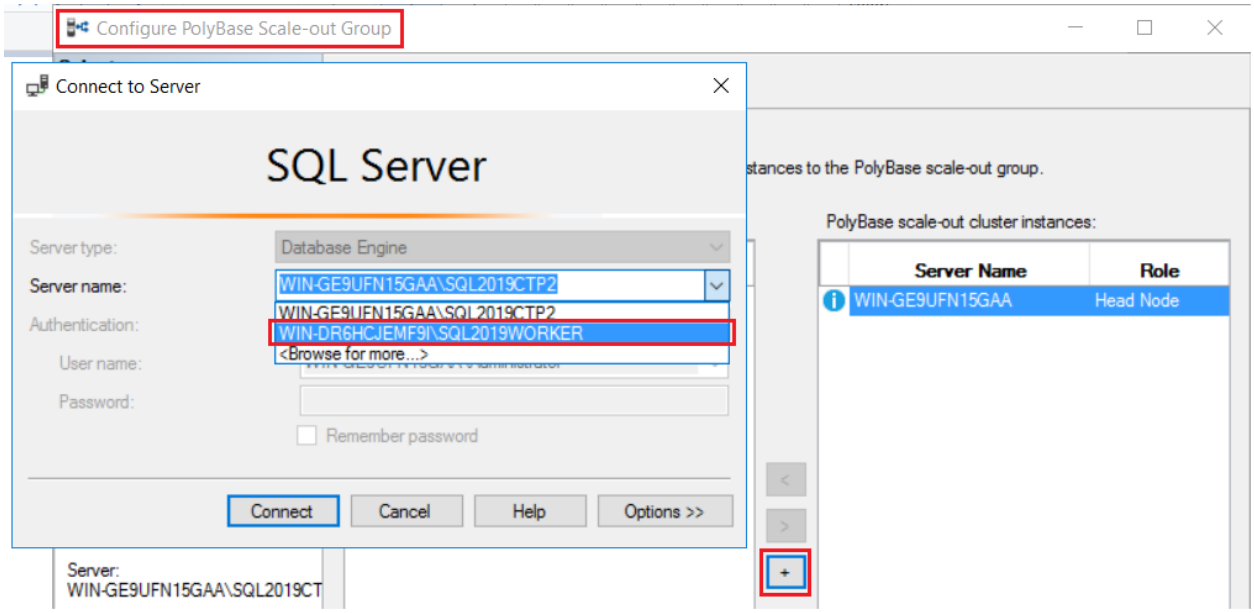


Figure 126:Adding a worker node to a cluster scale-out group head node

www.dbooks.org

For this to work, you have to designate an account you created in Active Directory as the account to run the PolyBase engine and data movement services. This is done during installation and is shown in Figure 127. You also need to install SQL Server 2019 as a compute node, and not a head node, as this will open the firewall for connectivity.

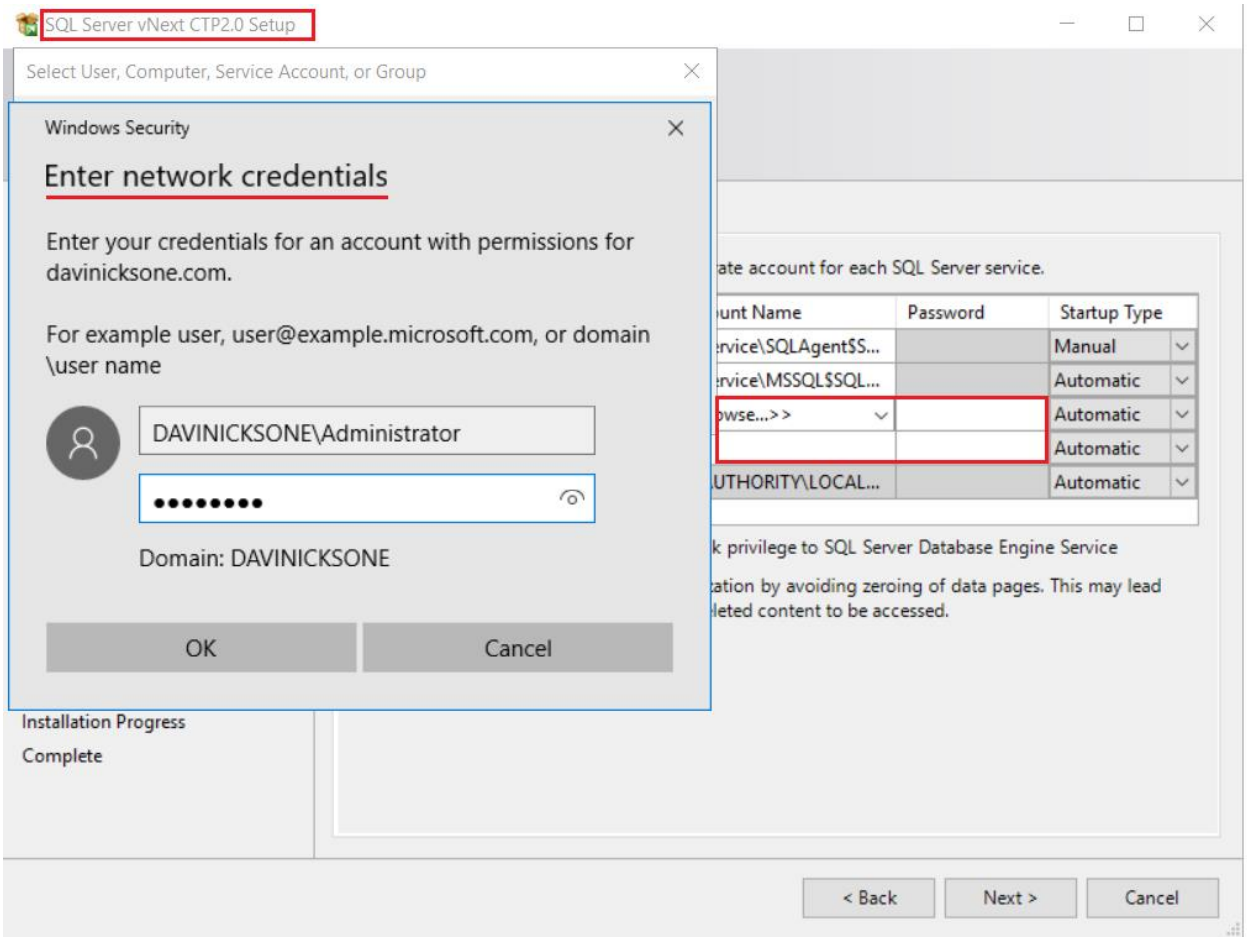


Figure 127: Designating accounts for PolyBase services in SQL Server 2019 installation

The installation will not continue unless you provide your network credentials, which you set up in Active Directory. You must use this account when installing SQL Server on each node in the cluster. The next figure shows an acceptable account setup for the PolyBase engine and data movement services.

Service	Account Name	Password	Startup Type
SQL Server Agent	NT Service\SQLAgent$S...		Manual
SQL Server Database Engine	NT Service\MSSQL$SQL...		Automatic
SQL Server Integration Services 15.0	NT Service\MsDtsServer...		Automatic
SQL Server Integration Services Sc...	NT Service\SSISScaleOut...		Automatic
SQL Server Integration Services Sc...	NT Service\SSISScaleOut...		Automatic
SQL Server PolyBase Engine	DAVINICKSONE\Admini...	••••••••	Automatic
SQL Server PolyBase Data Movem...	DAVINICKSONE\Admini...	••••••••	Automatic
SQL Full-text Filter Daemon Launc...	NT Service\MSSQLFDLa...		Manual
SQL Server Browser	NT AUTHORITY\LOCAL ...		Automatic

Figure 128: Set up of accounts for PolyBase services in SQL Server 2019

The previous steps are important because the service instance I've just added to the scale-out group is now showing as compatible, as seen in Figure 129.

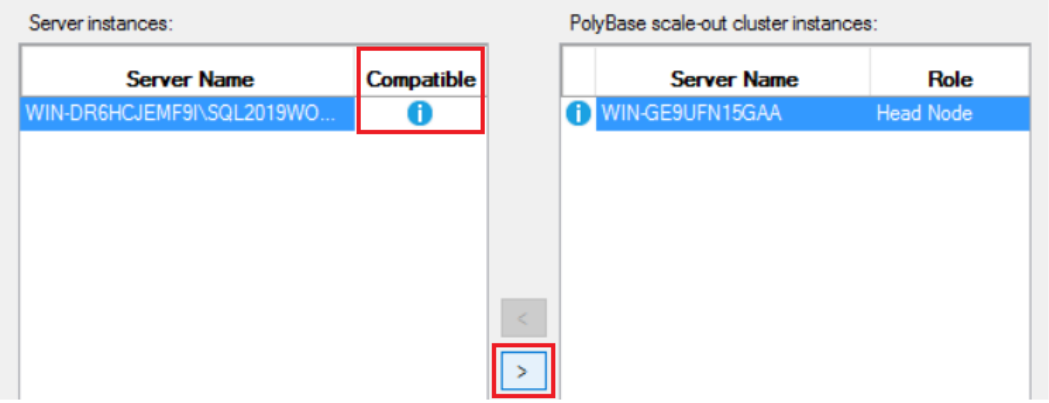


Figure 129: An imported server instance compatible with PolyBase scale-out groups

Had we not completed the steps outlined during installation, we would see a red icon denoting incompatibility, and the process would end here. You'd then have to reinstall the node correctly.

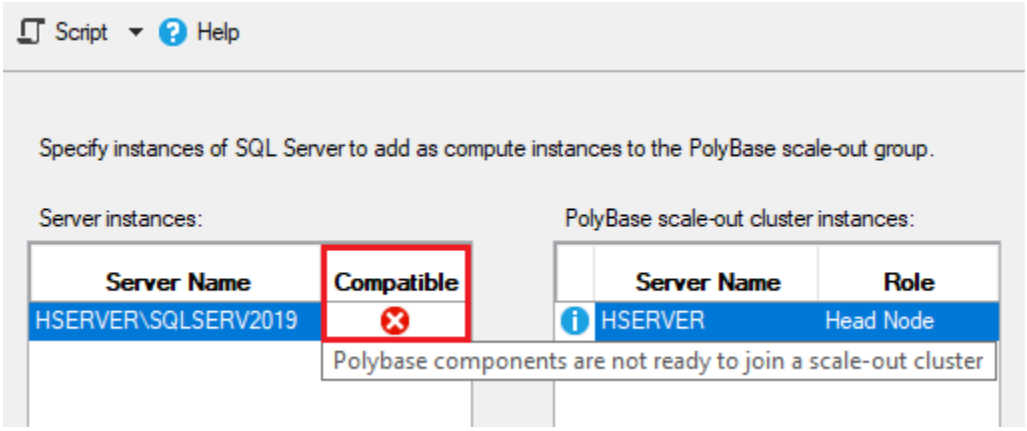


Figure 130: An example of an incompatible compute node instance

www.dbooks.org

We can click the arrow to add the server instance to the scale-out cluster. As you can see in Figure 131, it's been added as a compute (worker) node, hence the different symbol.

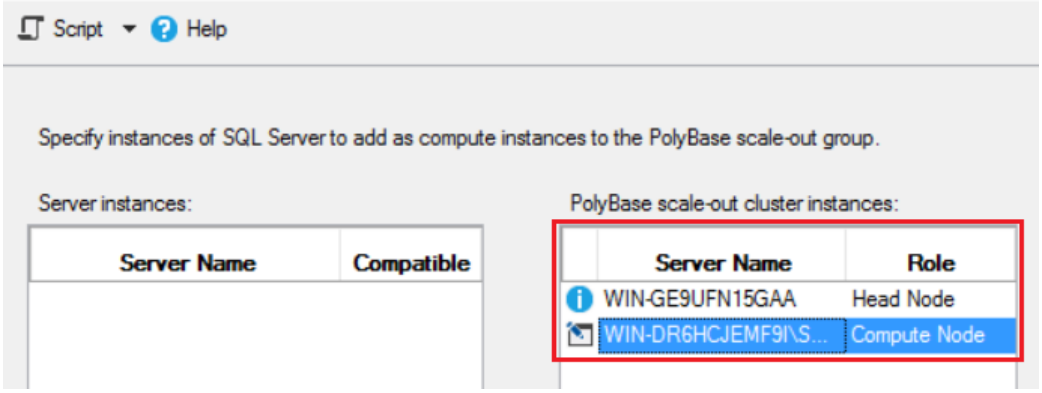


Figure 131: Server instance joined to PolyBase scale-out cluster

This book is being written as these innovations from Microsoft are in technology preview. There is an emerging commitment from Microsoft to support Hadoop within Windows. It is almost as if they are compensating for the Hadoop developer community that Linux has, and Windows doesn't. The very portable and lightweight Azure Data Studio (80-MB download), for example, can access Hadoop via the connection we just made in SQL Server 2019. We will look at Azure Data Studio in a bit more detail a little later. As it can connect to Hadoop, it becomes BI for Hadoop, albeit with a beautifully tiny footprint. It's also available for Linux and Mac, in keeping with Visual Studio Code, which is a small-footprint version of Visual Studio released by Microsoft.

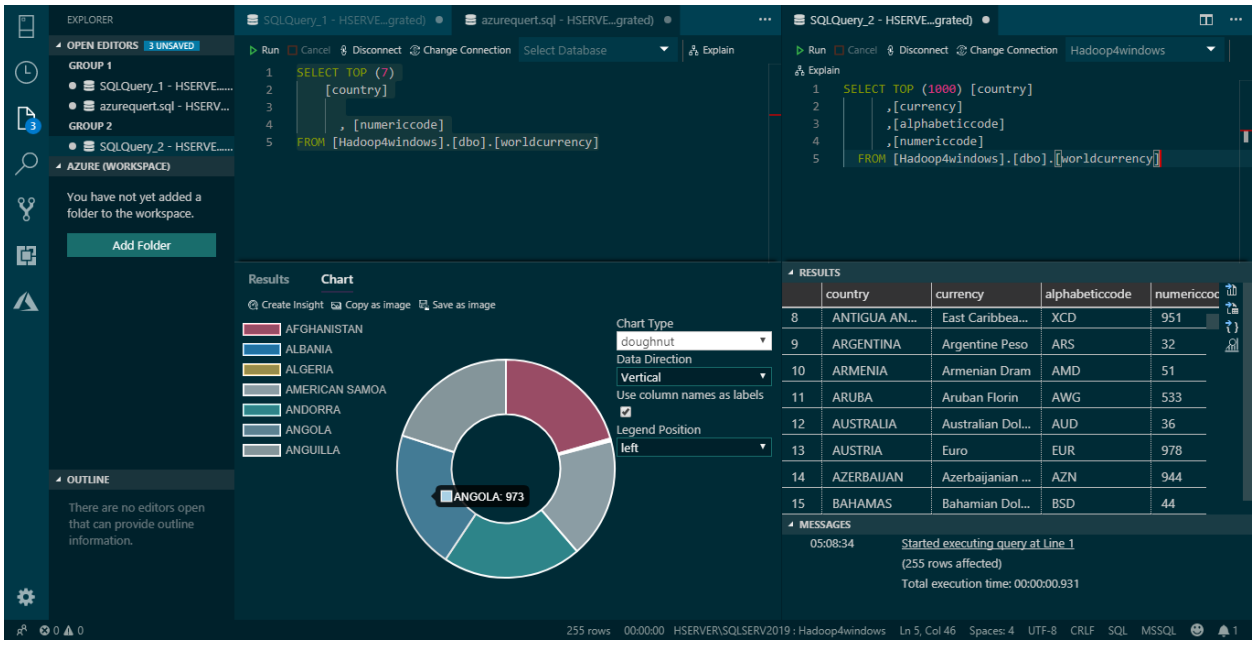


Figure 132: The allure of Microsoft Azure, Azure Data Studio connecting to HDFS via SQL Server 2019

There could be those who take another view, namely that setting up SQL Server with PolyBase scale-out groups involves more setup time than Hadoop itself. In Windows there's also an upfront server cost, so is this setup appearing twice as expensive and complex? Remember,

SQL Server is not free, just as Windows is not free. Here's a statement from a well-known Windows-based organization, summarizing why they don't use Hadoop for Windows.

"While possible, Hadoop for Windows would bring a lot of complexity."

It takes longer to set up Microsoft big data clusters than Syncfusion Hadoop clusters or certain Linux Hadoop clusters. To be objective, we must remember that it's only a technology preview at this stage. Also, the tiny footprint and simplicity of Azure Data Studio is a route that Microsoft is also pursuing. My hope, and I think the optimal solution, is for the two approaches to meet in the middle.

The choice of BI tools for Hadoop for Windows

In the previous section, I asked if Hadoop features integrated in SQL Server could benefit Windows BI tools. Was Microsoft's introduction of big data clusters an admission that traditional BI doesn't work well with big data? To answer these questions, we'll look at seven of the best namely:

- QlikView
- Tableau
- Power BI
- Azure Data Studio
- Arcadia
- Elasticsearch & Kibana
- Syncfusion Dashboard Designer

QlikView: qlikview.com

QlikView at one time or another been the best selling and most popular BI tool. It has a Server and Desktop edition, and as far as big data is concerned, can connect to Hive from Windows via ODBC. There is also a Spark ODBC driver, and the software can run on a group of servers handling larger datasets.

It is recommended that QlikView be the only application running on an individual server, due to its heavy resource usage of RAM in particular. This is partly because it is an in-memory tool that relies heavily on data compression, native QVDs, and data aggregation. Direct Discovery within QlikView allows you to connect live to an external data source, but you can only work with a reduced feature set when using Direct Discovery.

Tableau: tableau.com

Tableau has managed to take sizeable chunks of the BI market, and is arguably the second-biggest player behind QlikView in terms of purpose-built BI tools. Tableau comes in desktop and server editions but, unlike QlikView, is available for Linux. This is a more recent development, providing advantages like lower operating-system costs. Tableau connects to Hive via ODBC drivers, and is an in-memory application. It provides live connections to external data sources and has a wide range of data connectors, including Spark SQL. Tableau has excellent data grouping and classification tools that are part of a strong in-built feature set.

Power BI: powerbi.microsoft.com

www.dbooks.org

As we've already seen, Power BI can directly connect to the HDFS in Windows. It therefore gives deeper access to Hadoop than Hive data warehouse. It has benefitted from Hadoop integration across the Microsoft BI stack and wider Windows environment. A few years ago, I didn't use Power BI at all. Now it's indispensable, due to its native big data connectivity and tight Windows integration.

A new breed of BI for big data

Azure Data Studio: docs.microsoft.com

Azure Data Studio seems like a sleek and lightweight design on the surface. Dig a little deeper, and there's a lot of power that can be tightly integrated with all Microsoft big data and database innovations. Azure Data Studio does this in part by using extensions that can be added to the application. You cannot access Microsoft big data clusters without extensions, for example. Importantly, Microsoft sees big data as a whole concept, not just Hadoop. In addition, Azure Data Studio is cross-platform: it works on the Windows, Linux, and Mac platforms. Azure Data Studio has something of a companion product in Visual Studio Code. Visual Studio Code is a lightweight take on Visual Studio, and is similar in appearance to Azure Data Studio.

Visual Studio Code has given the Windows command prompt a contemporary feel and new lease on life. It can be enhanced by using extensions in the same way Azure Data Studio does. Visual Studio Code, shown in Figure 133, looks almost identical to Azure Data Studio. Extensions for both pieces of software can be installed from the internet or downloaded for offline installation. You click on the three red dots, as shown in the following figure, to get the extensions menu. From here, click **Install from VSIX**, which is the file extension for the file extensions.

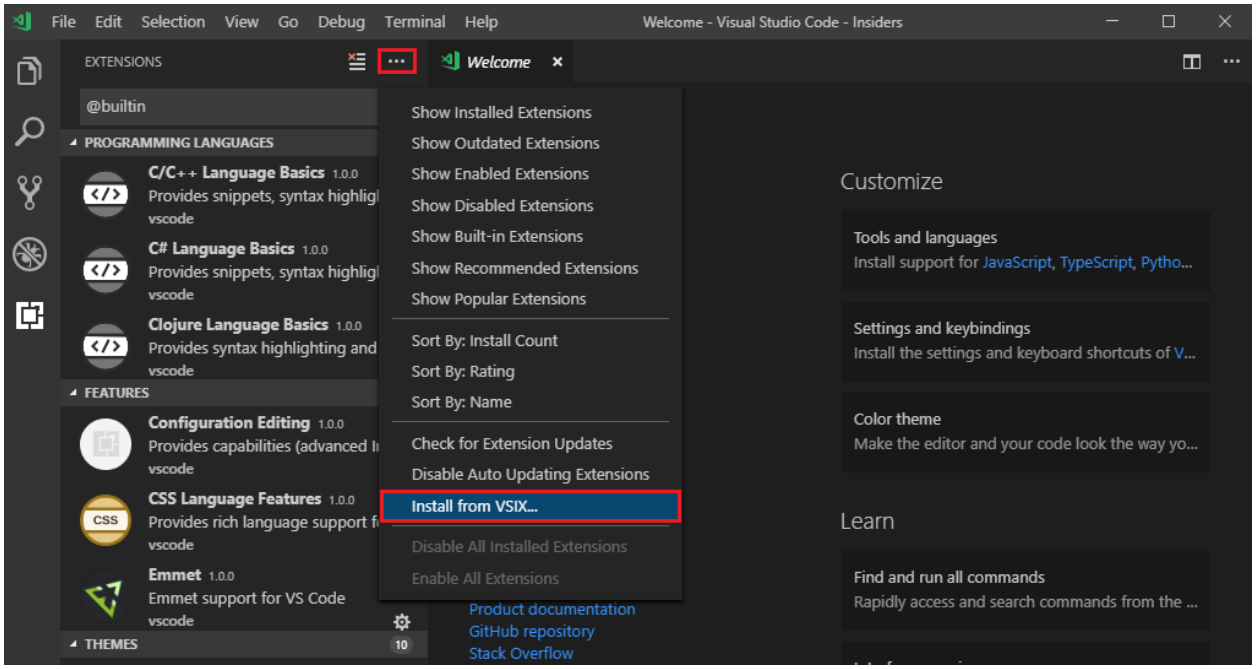


Figure 133: Microsoft Visual Studio Code

Arcadia: arcadiadata.com

Arcadia needs no help when connecting to Hadoop—there's no need to load ODBC drivers, as Arcadia connects to Hadoop natively. Arcadia also connects directly to the fast Impala query engine and runs on Windows and Mac. It's a BI tool built for big data, embracing data granularity, controlling data volumes, and accelerating processes. Join creation is automated, and connections to RDBMS and big data sources are supported. For some, it may be the only BI tool they need, because it meets requirements of scale of any BI task. It can also directly access Hadoop to run queries at very impressive speeds. Using Arcadia is like being in Hadoop itself. Add to that every visualization tool you'll ever need, and you've got Arcadia.

Elasticsearch and Kibana: elastic.co

While not originally designed to work with Hadoop, Elasticsearch can now use the HDFS as a snapshot repository. Support for the HDFS is provided via an HDFS snapshot/restore plugin. Elasticsearch and Kibana can, of course, search data in Hadoop and run just as smoothly on Windows as in Linux. Sure, the texts for these products are written around Linux, but this belies their integration with Windows. Elasticsearch 6.6.0 can be installed as a Windows service, and Kibana 6.6.0 has new features that allow the easy import of data files. The real strength of these tools is the ingestion and display of real-time and near real-time data. Their footprint is comparatively small, and installation is simple and speedy. You can also very tightly manage the resources allocated to nodes and clusters within your installation.

Cloudera and Hortonworks in Linux: hortonworks.com, cloudera.com

It's important for Windows products to compare well against their Linux counterparts. Sadly, the absence of Impala on the Windows platform is currently a problem. In a project unrelated to this publication, I have been in contact with the Impala developers on this subject.

Outside of Microsoft big data clusters, what tools are there for fast queries involving joins? What if you don't want to use SQL Server—what other choices are out there? All too often, the other choices involve using Linux products, even if it means learning a new system.

Cloudera CDH is a Hadoop release you'd struggle to create in Windows, as key functionality isn't available. If Syncfusion could put Impala in their Big Data Platform they would; currently, however, it isn't possible. I haven't mentioned it until now, but the exercises I've been doing in Windows are exercises I've duplicated in Hadoop on Linux. We'll see some of this work later on in the chapter. I did this because it's important to identify features that may benefit Hadoop in Windows. You'll recall that I mentioned cgroups and updating ecosystem elements like Hive, as new versions become available.

Cloudera and Hortonworks have features you'd be pleased to see in Windows solutions. Hadoop for Windows should take more advantage of the interactive nature of Windows. This is because the best Linux Hadoop distributions are more interactive than they used to be. The older, rather "wooden" Linux Hadoop is being replaced by sleeker, more modern designs. Microsoft has observed this and given Azure Data Studio a contemporary feel that you can change at will. You'll see some of this as we go further into this chapter.

www.dbooks.org

Connecting BI tools to Hadoop in Windows

QlikView

A key benchmark for Hadoop in Windows is the ability to connect live with Hadoop. QlikView connects live to data sources with Direct Discovery, but you lose certain functionality. Direct Discovery also requires different code scripting. Some other BI tools, like Tableau, achieve it by just clicking a button.

The features unavailable with Direct Discovery include:

- Advanced calculations
- Calculated dimensions
- Comparative Analysis (Alternate State) on QlikView objects using Direct Discovery flds
- Direct Discovery fields are not supported on Global Search
- Binary load from a QlikView application with Direct Discovery table
- Section access and data reduction
- Loop and Reduce
- Table naming in script does not apply to the Direct table
- The use of * after `DIRECT SELECT` on a load script (`DIRECT SELECT *`)

To see if we can access the HDFS live without restrictions, we'll connect to SQL Server via **Edit Script**. Click **Connect** and enter the login details for SQL Server, as shown in Figure 134.

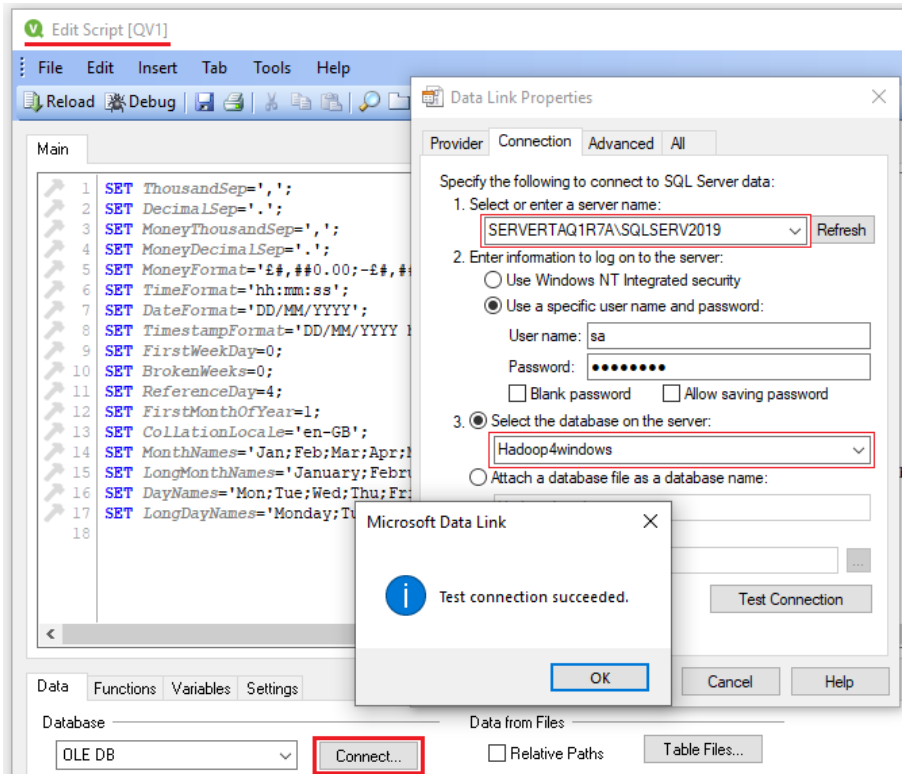


Figure 134: Connecting to SQL Server 2019 from QlikView

Click **Test Connection** before proceeding. You should see the **Test connection succeeded** message displayed, which tells you you've successfully made connection with SQL Server. The connection to SQL Server 2019 has allowed direct access to Hadoop via the external table that we created called **worldcurrency**. The Preview facility displays the data, as shown in the following figure.

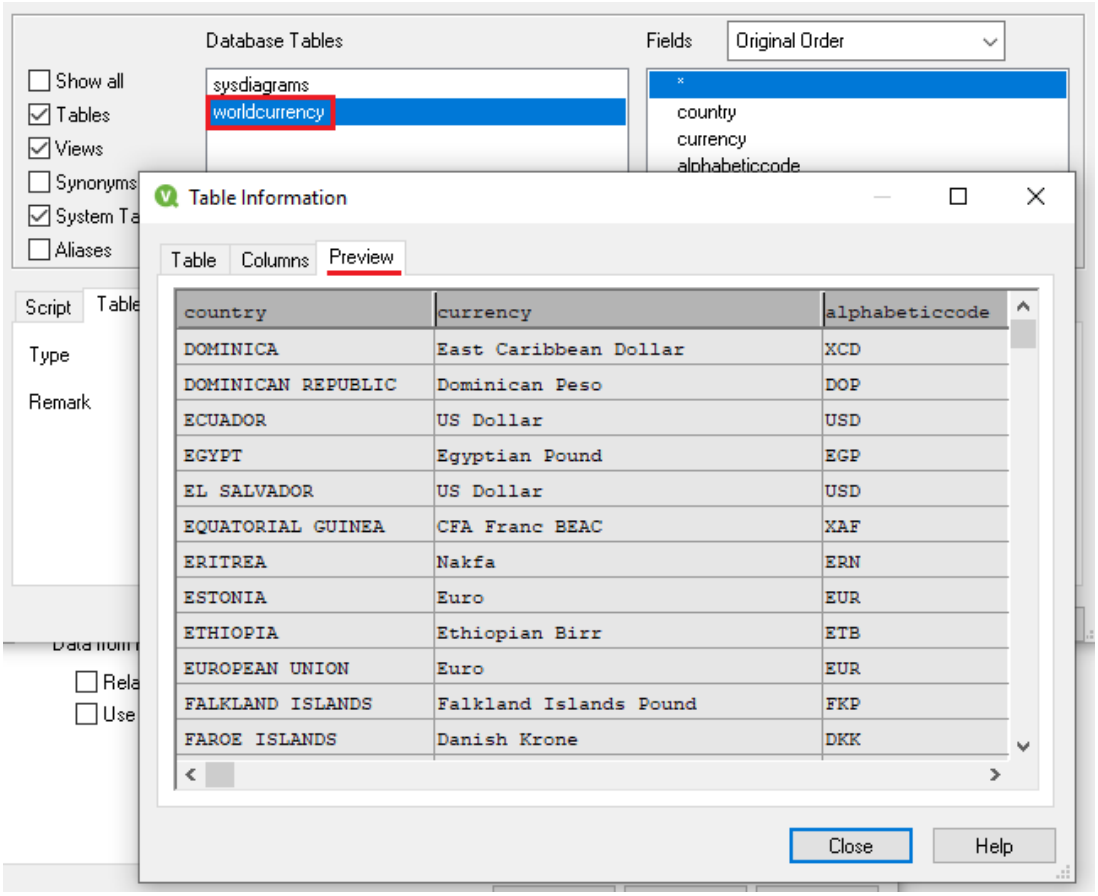


Figure 135: Hadoop file being read live in QlikView via SQL Server 2019

The load script is then executed to load the data into the application, as shown next.

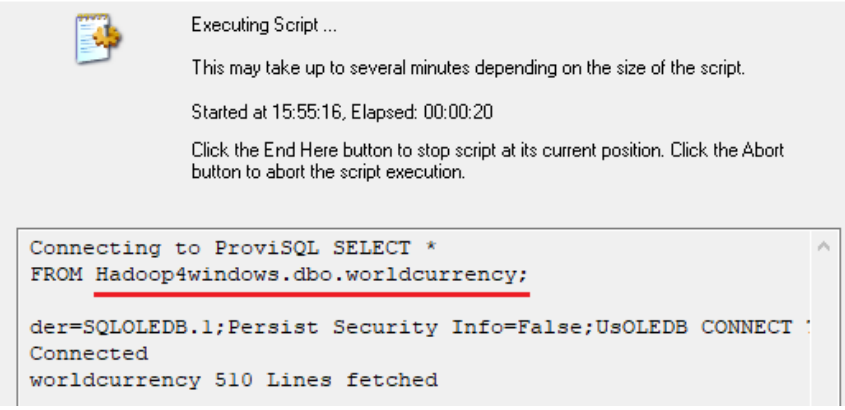
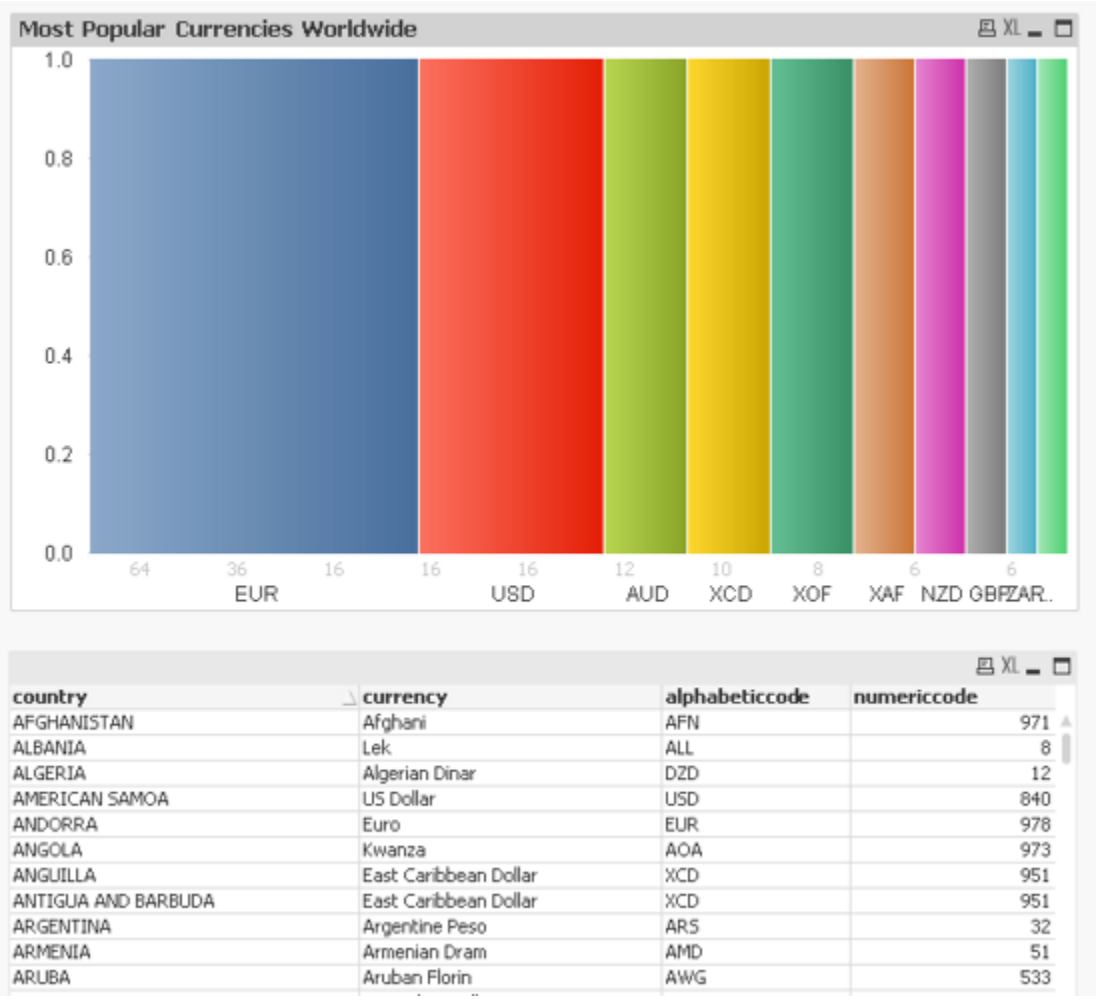


Figure 136: Script is run within QlikView to load Hadoop data into the application

www.dbooks.org

Though we were able to connect to Hadoop from SQL Server, the action of loading the script ensures we can't ingest live data without the limited Direct Discovery. The way it works means you essentially work from extracts; you would need to reload data to pick up any changes. While we can create dashboards, as shown in Figure 137, we don't have the live connection to Hadoop we desire or could have. Like so many BI tools not built for big data, we're importing data due to the limits of the BI tool. Direct Discovery compensates for the fact that QlikView wasn't built for big data analysis. Sadly, the unavailability of key functionality is counterproductive. The restrictions of Direct Discovery previously listed are also not the only restrictions. You cannot use pivot charts and mini charts, for example, and there are performance-tuning issues that must be dealt with at the data source.

We could load the data into QVD files for fast load times and high data compression. You'd still be loading extracts though, albeit much faster and with functionality unavailable in Direct Discovery. Organizations I've worked at with multiple QlikView servers never considered using it for live connections.



country	currency	alphabeticcode	numericcode
AFGHANISTAN	Afghani	AFN	971
ALBANIA	Lek	ALL	8
ALGERIA	Algerian Dinar	DZD	12
AMERICAN SAMOA	US Dollar	USD	840
ANDORRA	Euro	EUR	978
ANGOLA	Kwanza	AOA	973
ANGUILLA	East Caribbean Dollar	XCD	951
ANTIGUA AND BARBUDA	East Caribbean Dollar	XCD	951
ARGENTINA	Argentine Peso	ARS	32
ARMENIA	Armenian Dram	AMD	51
ARUBA	Aruban Florin	AWG	533

Figure 137: Maybe it's live, maybe it's not: QlikView visualization from loaded Hadoop data extract

Tableau

Let's see if Tableau fares any better than QlikView when attempting to connect live to the HDFS. Connecting to SQL Server from Tableau is a straightforward task. You can't help but notice the large number of other connectors available.

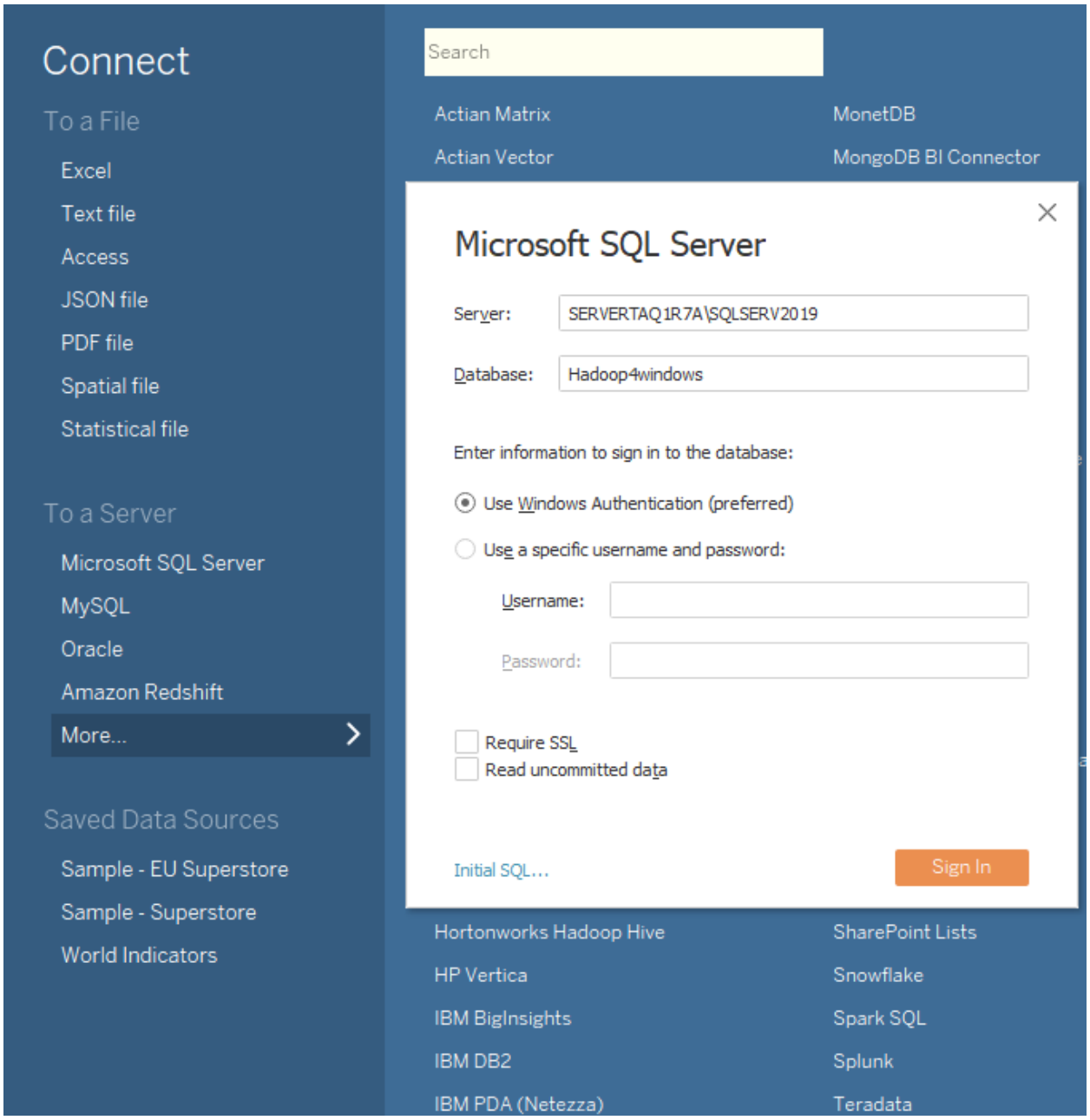


Figure 138: Connecting to SQL Server 2019 within Tableau

After connecting to SQL Server from the preceding screen, you should see what's shown in Figure 139. It is very different from what we saw in QlikView at the same stage.

The first thing we notice is the "Live" connection, indicated at the top-right side of the screen. Tableau has also identified the live **worldcurrency** table from the HDFS file data. Tableau

www.dbooks.org

achieves this by sending dynamic SQL to the source system rather than importing it. Advantages to this approach include less storage space, as you avoid duplicating source data in the BI system.

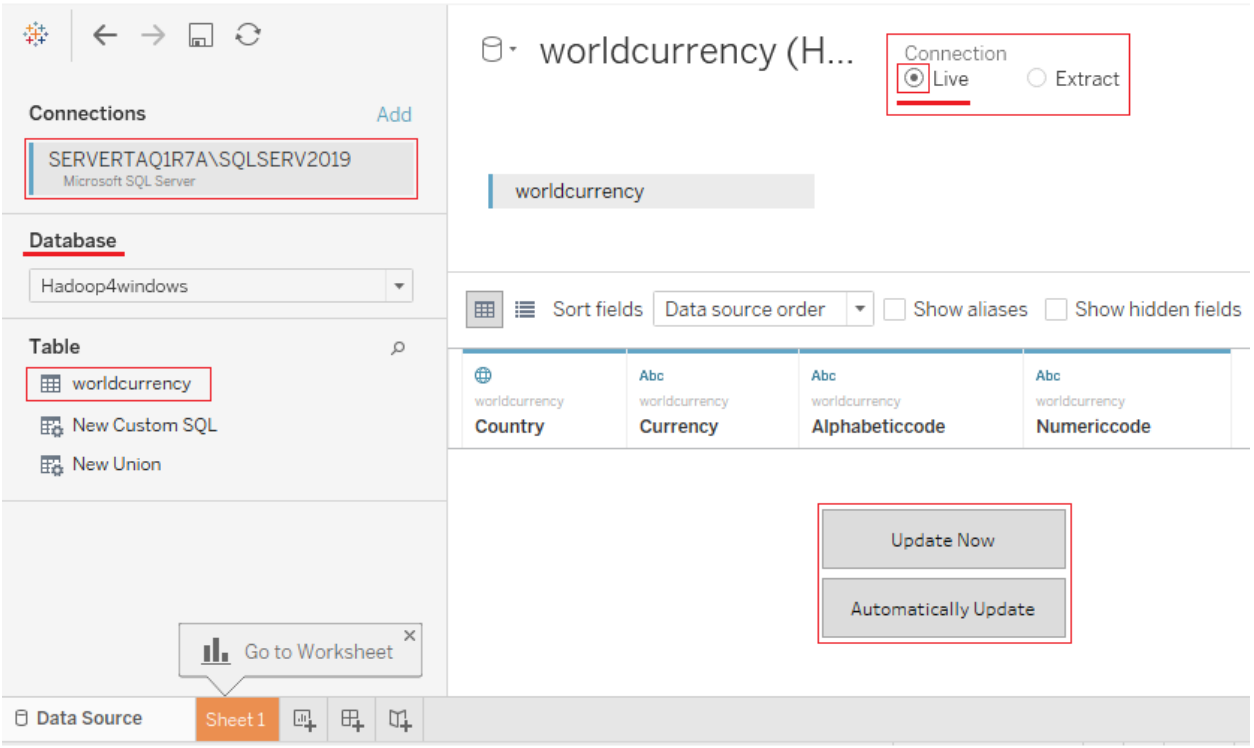


Figure 139: Live connection to SQL Server from Tableau connecting directly to HDFS

Live connection does not mean a reduced feature set; without being connected to the internet, you can map data from the countries listed in the Hadoop data file.

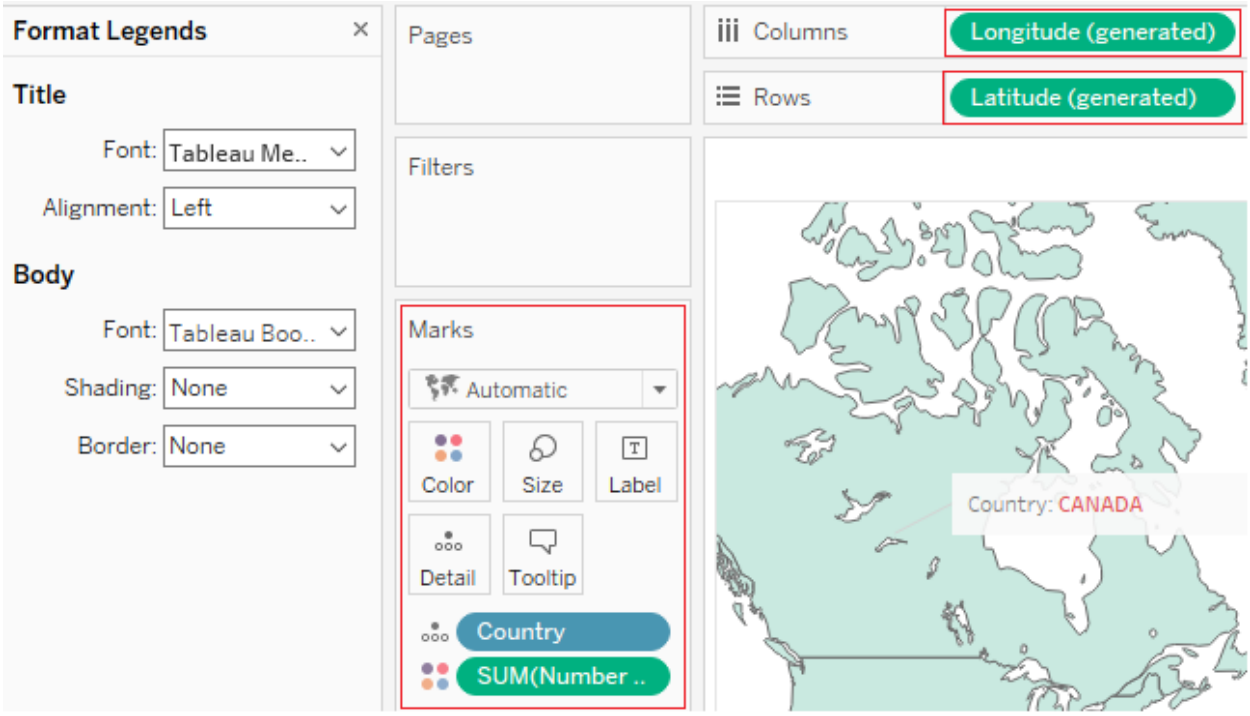


Figure 140: Features are not restricted when using live connection in Tableau

Tableau has passed the test with flying colors—there was no discernible difference in performance, and you'd never guess you were connecting to raw Hadoop. So, the work from Microsoft to integrate Hadoop into SQL Server can benefit third-party BI tools.

www.dbooks.org

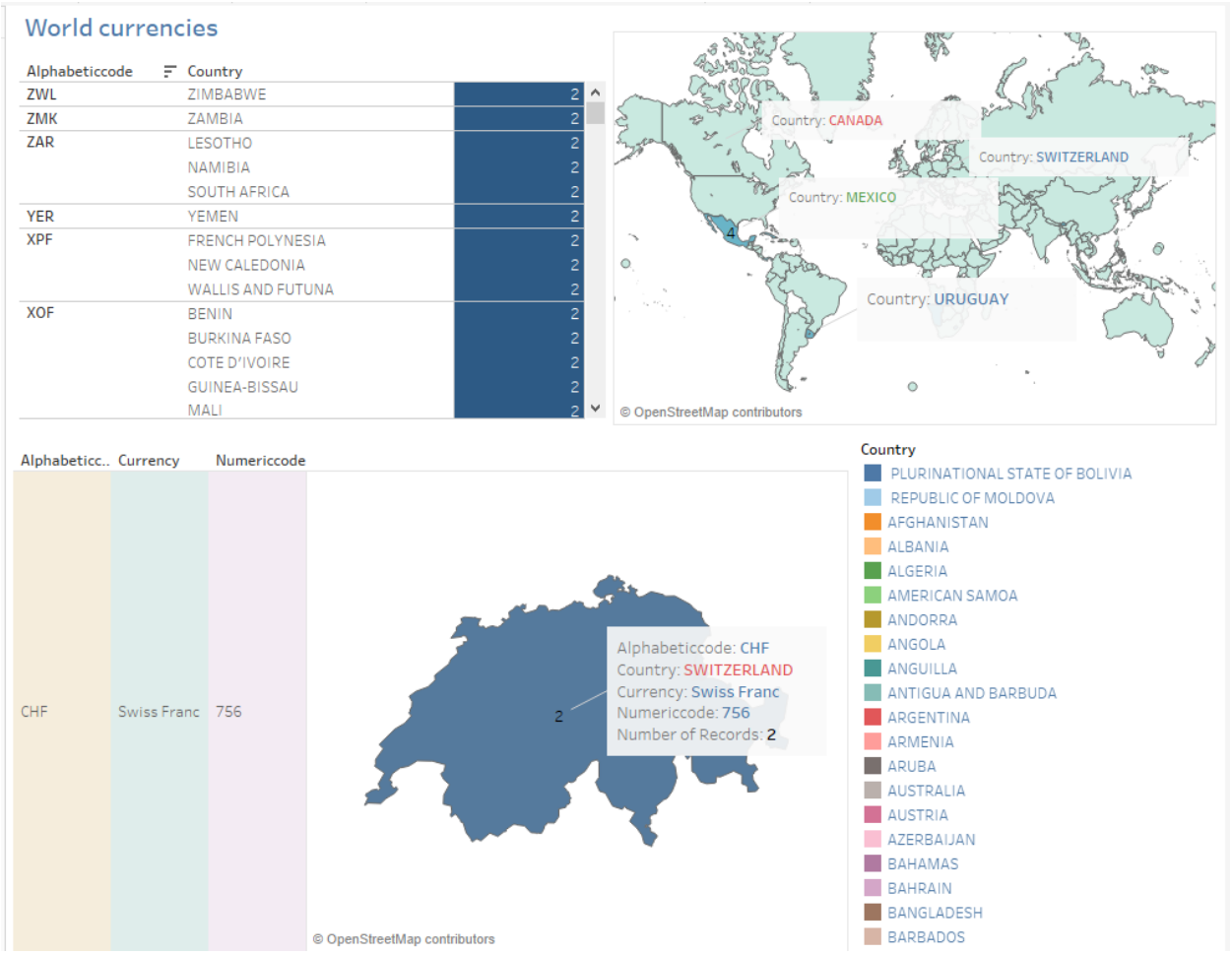


Figure 141: Tableau dashboard created from live connection to HDFS

Tableau perhaps reflects both sides of the story: the ability to connect live to Hadoop within Windows, and now, producing Tableau for Linux. Tableau Server for Linux has integrated well with SQL Server for Linux, which is making a big impression itself. Perhaps the lines are becoming blurred, and the future is not as black and white as Windows or Linux. The efforts of Microsoft to build the WSL (Windows Subsystem for Linux) is perhaps the strongest indicator of this. The ability to access both environments from within one environment is a noble aspiration, though the reality may be more difficult to achieve.

Power BI

We've already seen Power BI in Chapter 1, so we know what it can do. That said, I can confirm that Power BI can take advantage of the live SQL Server connection to HDFS by using Direct Query. Direct Query allows you to access data directly from your chosen data source. It's enabled by clicking the **DirectQuery** button highlighted in Figure 142. This eliminates the need to import data by giving you a live connection to interrogate larger data volumes.

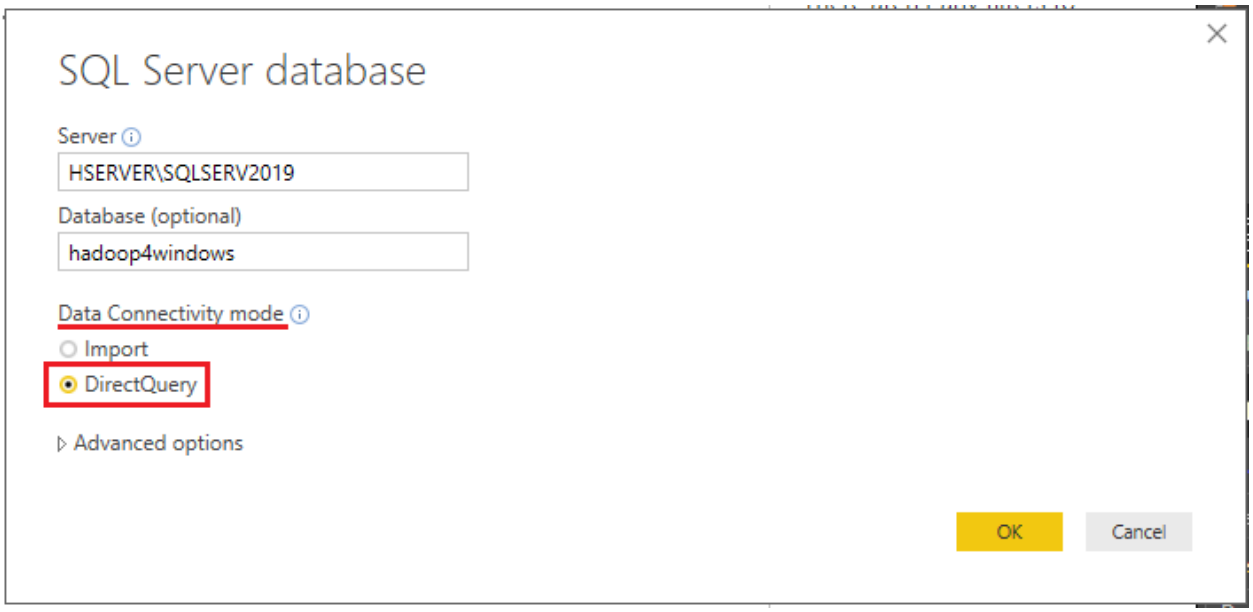


Figure 142: Power BI live connection to SQL Server to read HDFS files live

Power BI can also access the Impala high-speed query engine in Linux. It's interesting to note that there is no Hive connector in Power BI. Microsoft has confronted the problem of slow query speed with joins in Hive, and gone straight for Impala. If you really want to use just Hive, you can use an ODBC connection.



Figure 143: Power BI live connection to Impala via the direct query mechanism

Azure Data Studio

I've commented on Azure Data Studio and how to expand its feature set by adding extensions. I didn't mention its ability to tightly monitor its own resource use and container-like nature; everything seems to fit or work in a neat, lightweight box.

www.dbooks.org

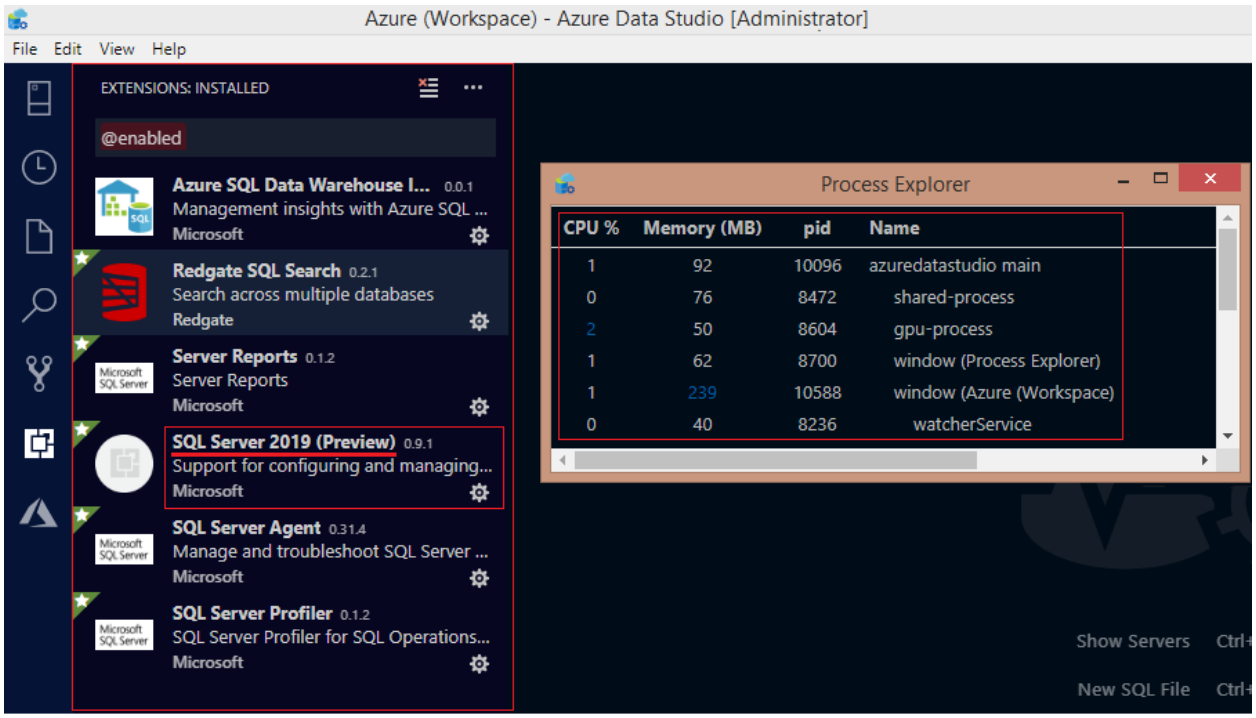


Figure 144: Azure Data Studio with extensions installed

Azure Data Studio can connect to our live HDFS connection in the most versatile way. This is achieved by using the SQL Server 2019 extension highlighted in Figure 144, and deployed as shown in Figure 145.

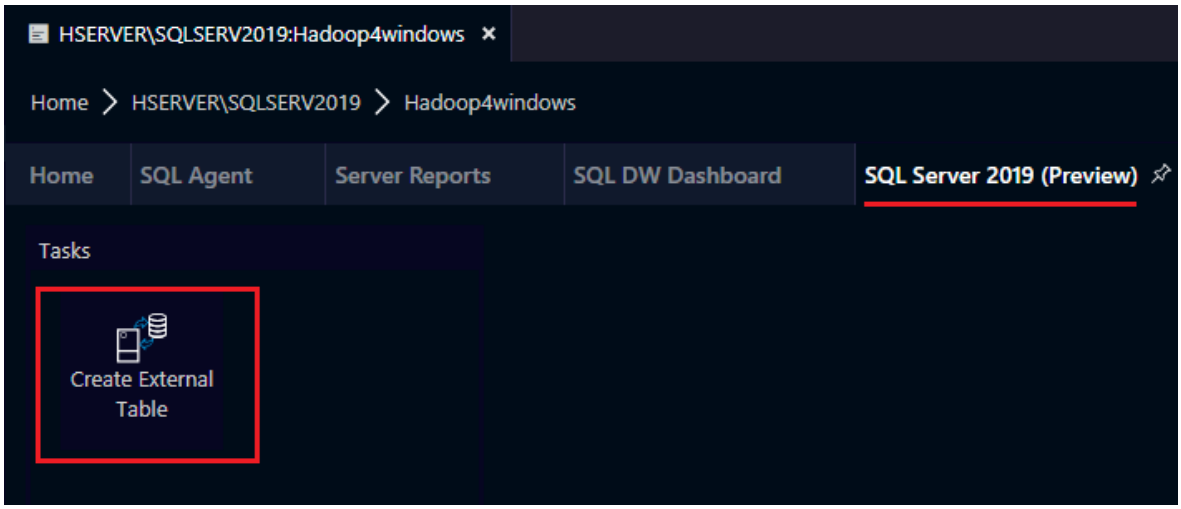


Figure 145: Deployed Azure Data Studio extension for SQL Server 2019

One of the features within the extension is the ability to create external tables. Does it reveal the thinking of Microsoft? Why should you have to be in a system to create a table? Shouldn't there be an ability not just to exchange data, but to design the components that hold it?

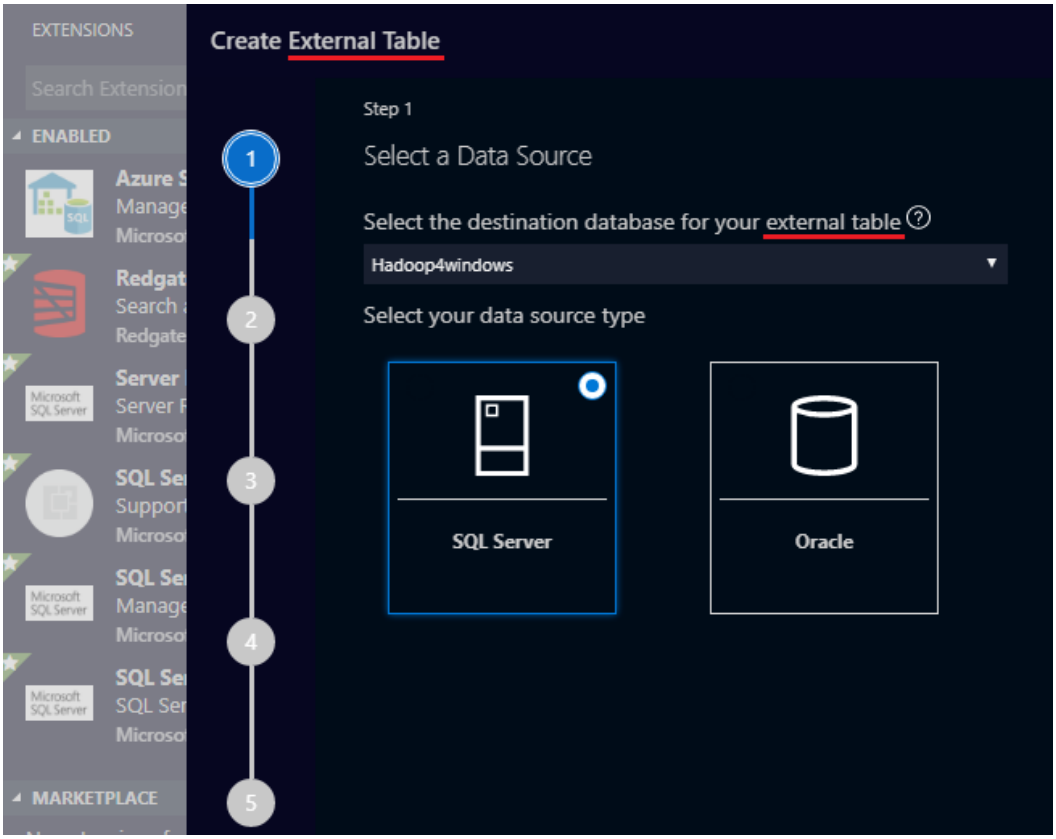


Figure 146: Create external table in Azure Data Studio

Security issues have been overcome by the creation of a database master key to secure the credentials used by an external data source.

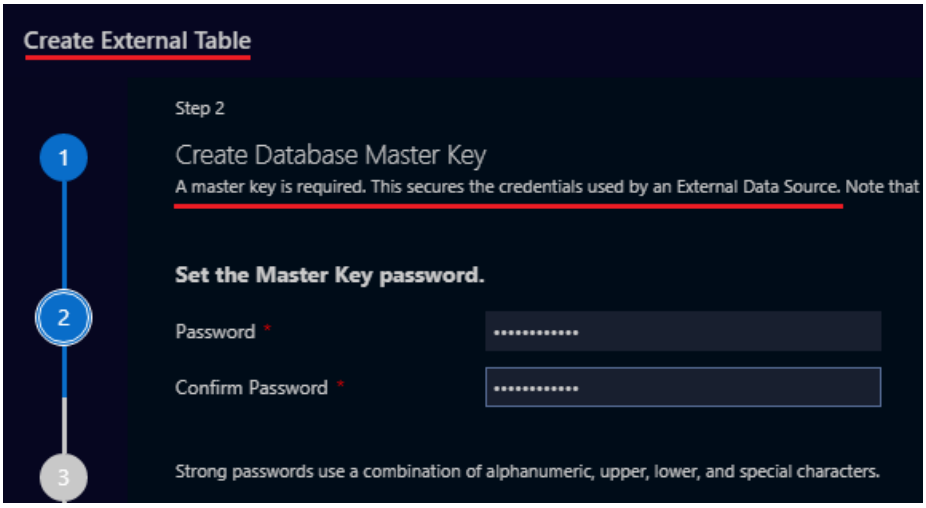


Figure 147: Creating database master key in Azure Data Studio

The following figure shows the Azure Data Studio Server dashboard. The four SQL Server extensions installed are highlighted in red. You can view and search for other databases, including our highlighted **Hadoop4windows** database. You simply click on each extension to access its functionality.

www.dbooks.org

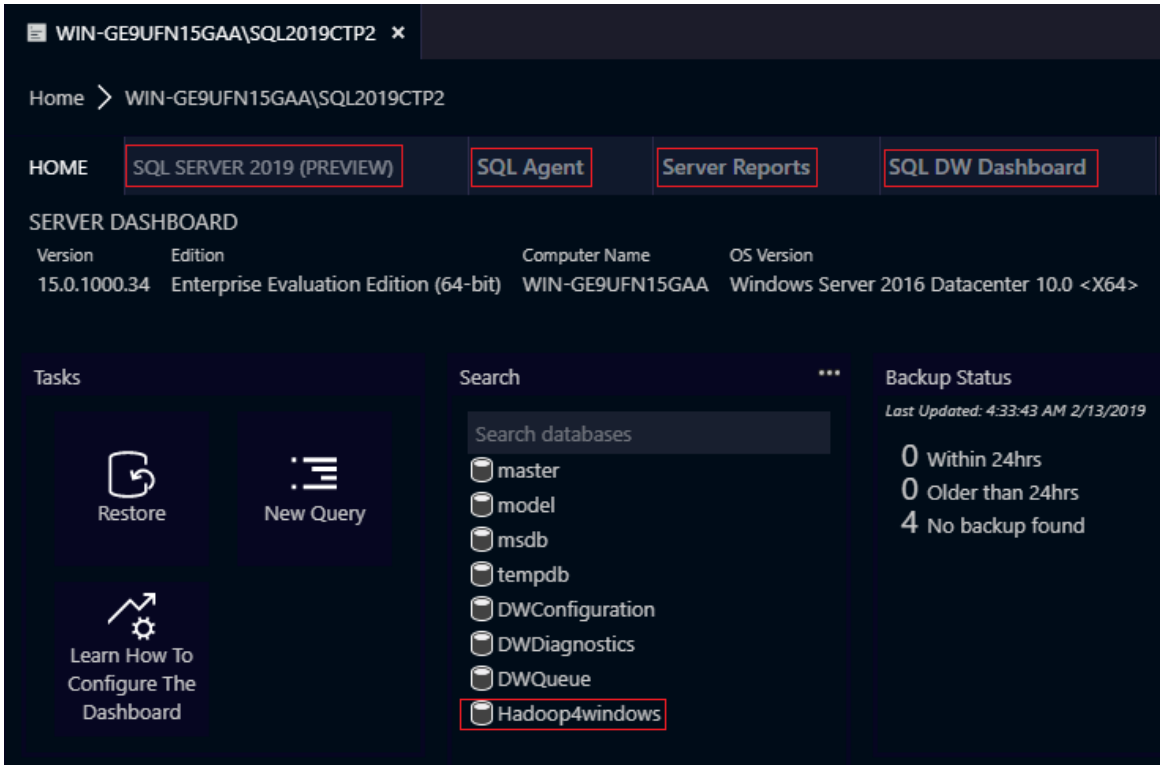


Figure 148: Azure Data Studio Server Dashboard

In order to connect to a server, you need to click the server symbol, as highlighted in Figure 149. You will find it on the top, left-hand side of the screen.

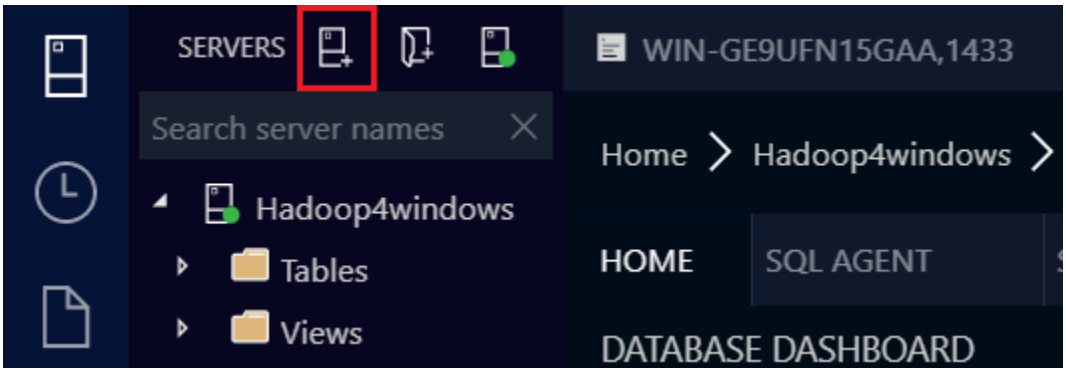


Figure 149: Making a connection to SQL Server 2019 in Azure Data Studio

You are then taken to the login screen, as shown in the following figure.

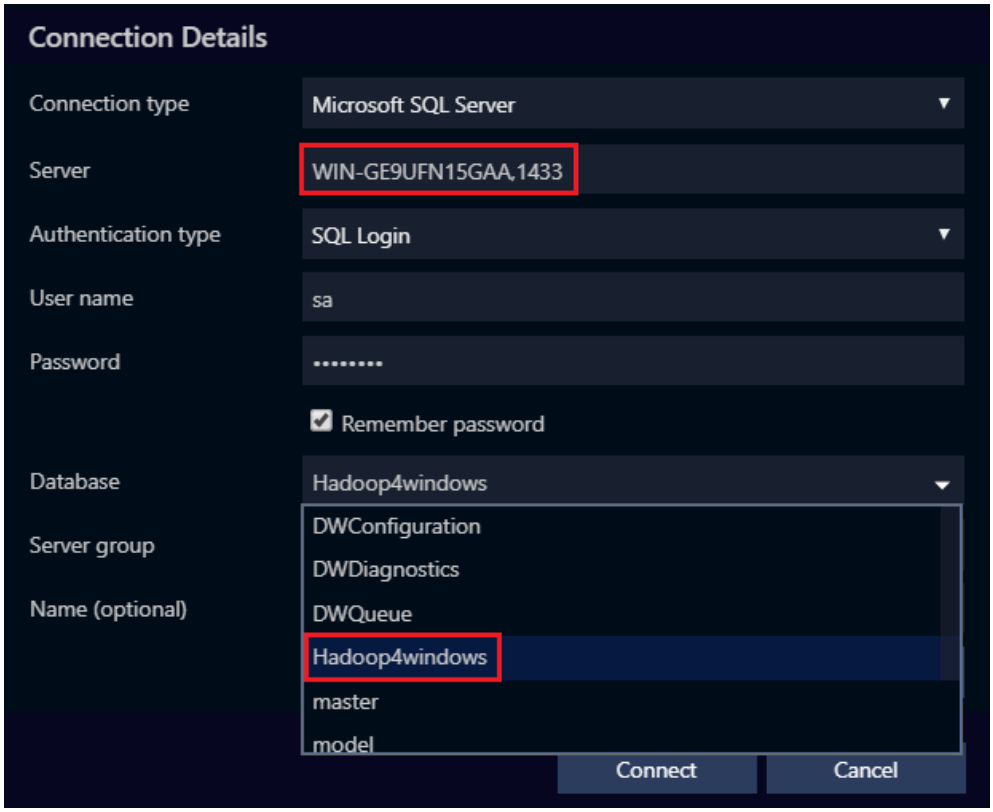


Figure 150: Azure Data Studio login

The two fields highlighted show the server and the individual database login fields. We then enter server name and port number, after making sure the Server is configured to accept remote connections. This includes allocating a port number for SQL Server to accept connections on. If you don't do this, you may only be able to connect to a local server. After you log in, you see the server dashboard screen we saw earlier, plus more details on the left-hand side. You see additional folders, including server objects and endpoints, as shown in the following figure.

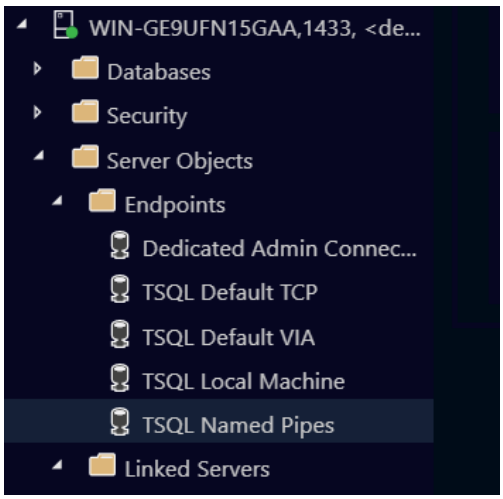


Figure 151: Azure folders showing server objects and endpoints

www.dbooks.org

When you're in Azure Data Studio, you can change the look and feel of the app completely. You can also access the live Hadoop data connection we created, as shown in Figure 152.

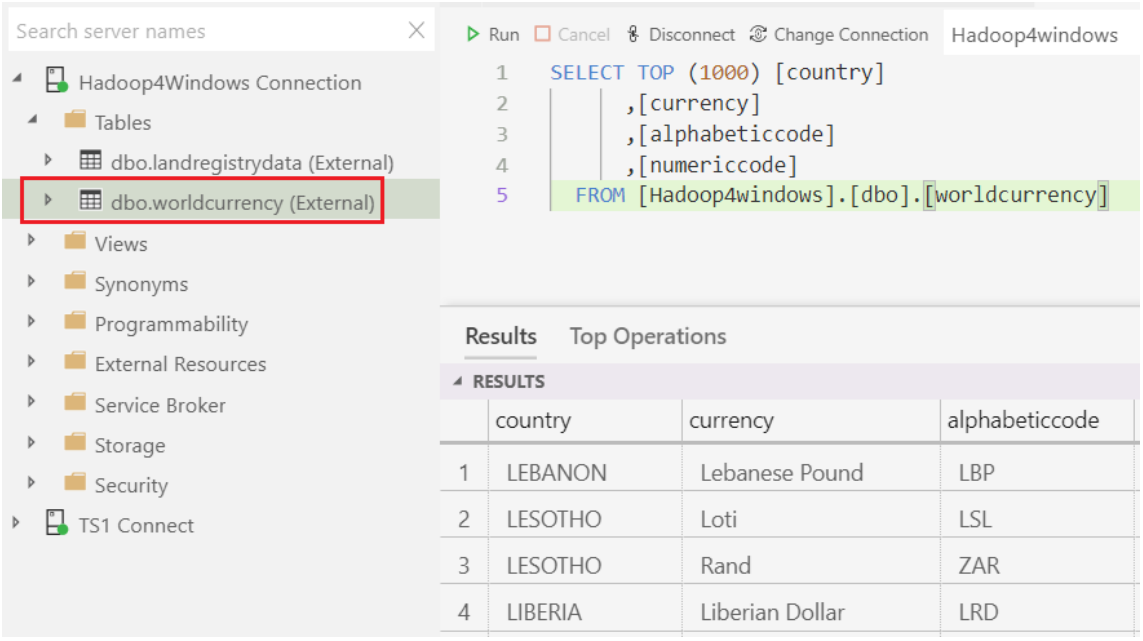


Figure 152: Live connection to HDFS from Azure Data Studio

The problem of slow joins in Hadoop is eliminated in the Microsoft big data solution.

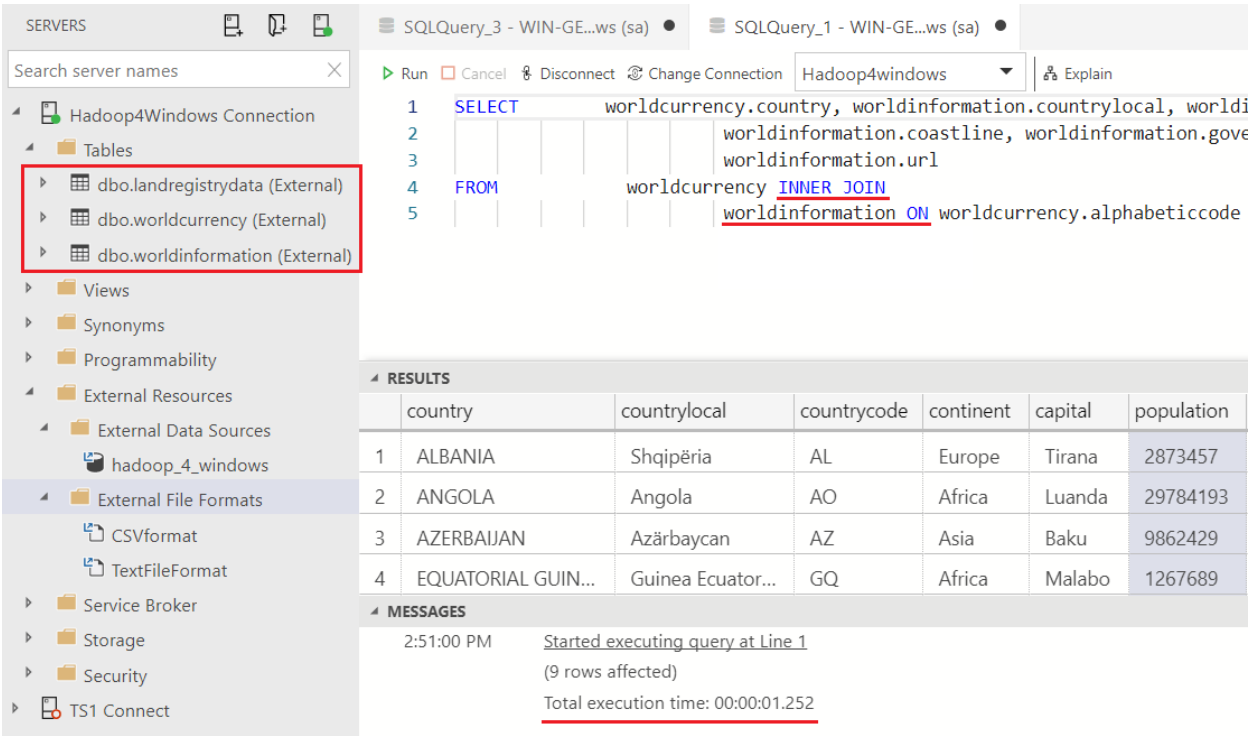


Figure 153: Fast execution of Hadoop queries involving joins as fast as Impala

In the preceding figure, you can see the additional table among those highlighted, called **worldinformation**. The tables are followed by the word **external**, as they are live connections to Hadoop. Running a query joining that table to our **worldcurrency** table was executed in 00:00:01.252 seconds. In any version of Hive in any Hadoop release, this would take at least a few minutes. This is one of the reasons why some people prefer what Microsoft is doing with big data and Hadoop. I'm able to do all this from an 80-MB app that is around 400 MB when installed, plus extensions. It's a lot of bang for your buck, or it would be if it wasn't free of charge on Mac, Linux, and Windows. The fun doesn't stop here: you can create dashboards from your live Hadoop queries. Charts are created "on the fly" and shown on the Chart tab, as highlighted in Figure 154. You can make as many charts as you wish to populate your dashboards.

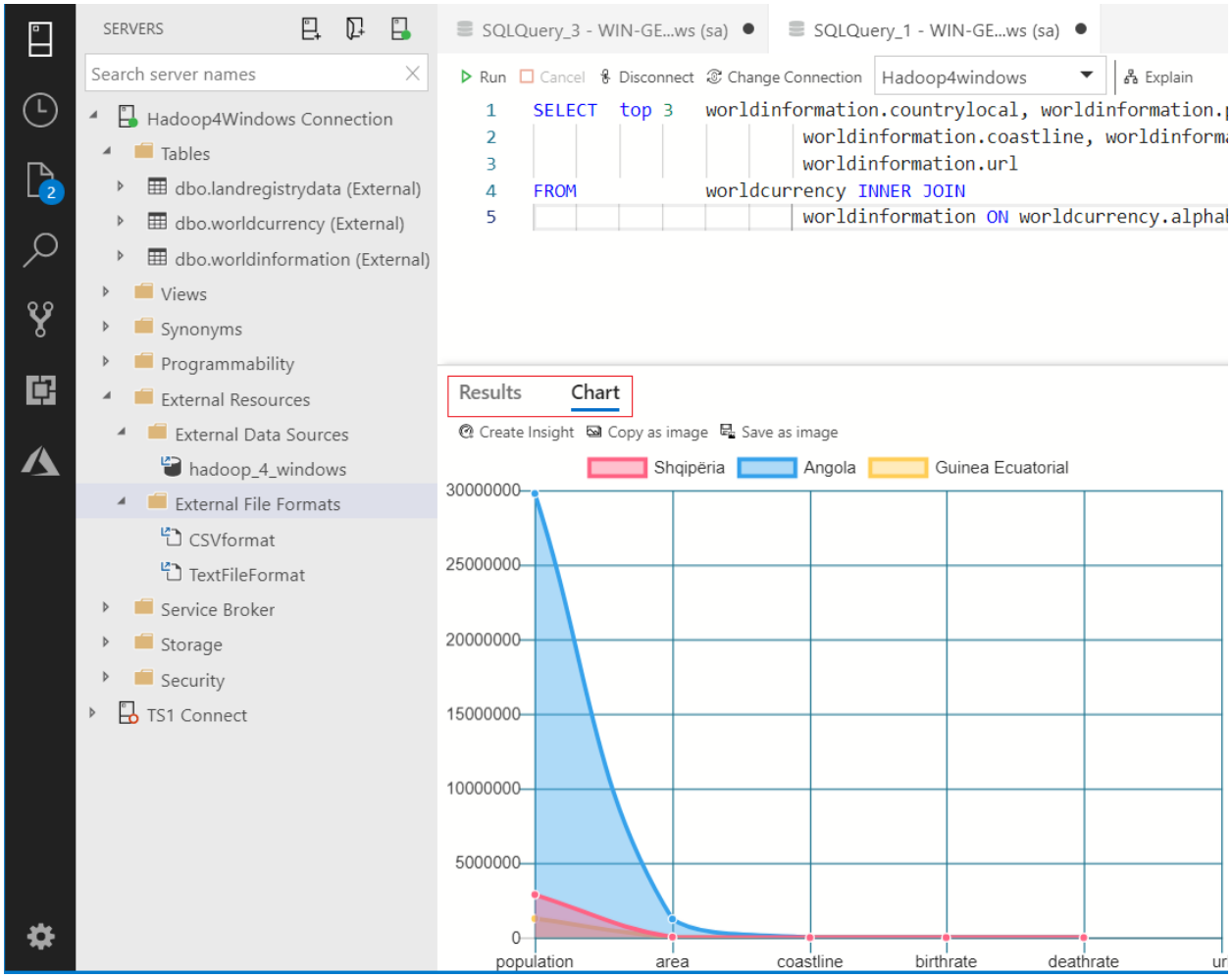


Figure 154: Live charts based on the results of high speed Hadoop queries with joins

The power of the app is such that you can change the visualization types in an instant on screen. You don't have to go to a separate area or separate mode of operation—they are instant, on-screen changes.

www.dbooks.org

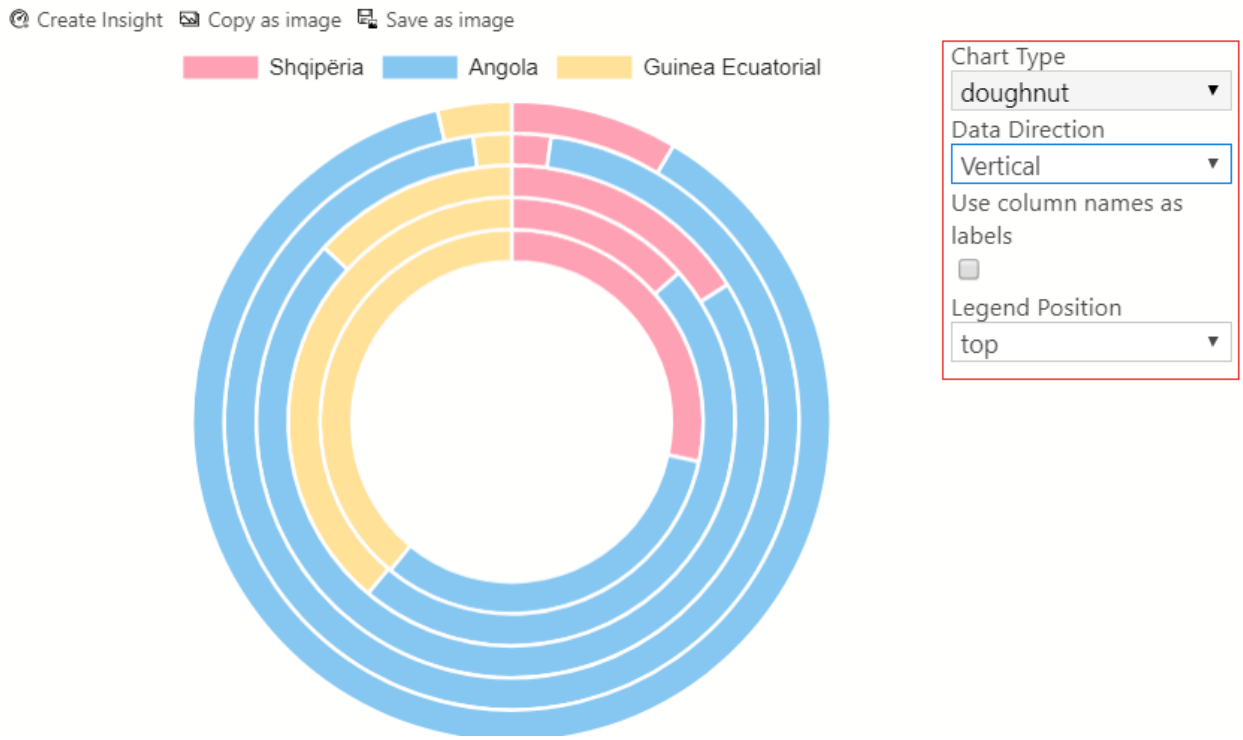


Figure 155: Instant changes of visualizations using Azure Data Studio

Arcadia Data

Arcadia comes with everything you need to connect to Hadoop contained within it.



Figure 156: Arcadia Data, the server is running

After installing Arcadia, click the **Start** button shown in Figure 156. You see a message displaying **Server is running**, at which point you click **Go**. At this point you must register the product, or you cannot proceed. You then create a Hive connection in Arcadia, as shown next.

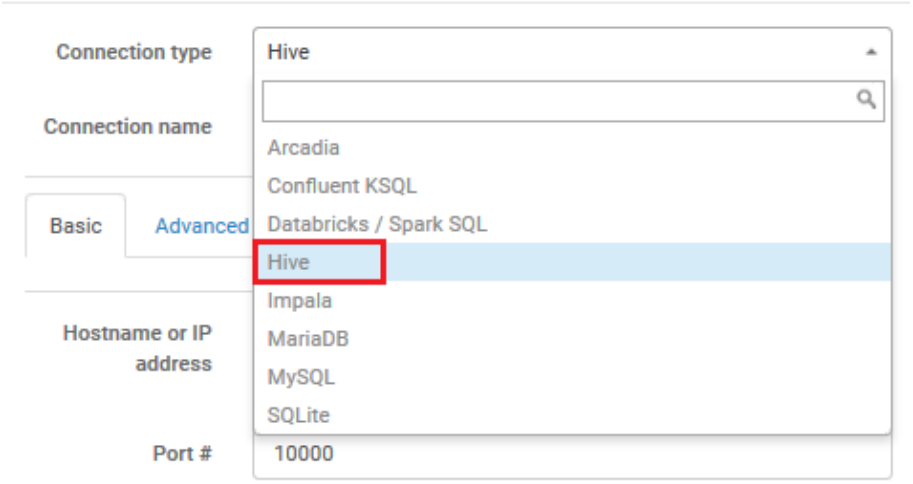


Figure 157: Hive connection in Arcadia, requires hostname, username and password to Hive

With your connection created, the Connection Explorer locates the databases in Hadoop and lists all the tables in them.

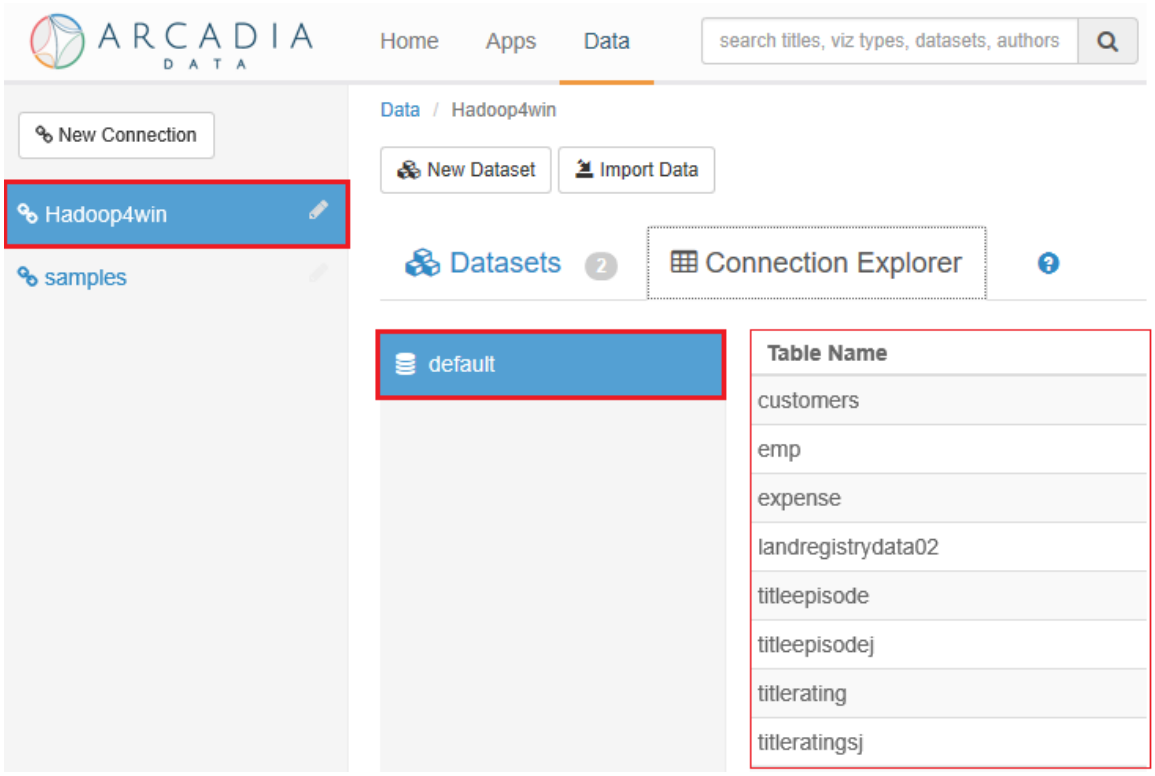


Figure 158: Arcadia Connection Explorer

www.dbooks.org

The connection to Hive is a live one, and Arcadia is an interactive experience. You drag and drop the fields from your tables into dimensions and measures to create visualizations. This is shown in Figure 159, where the fields highlighted in the red rectangle are dragged into the fields highlighted in the green rectangle.

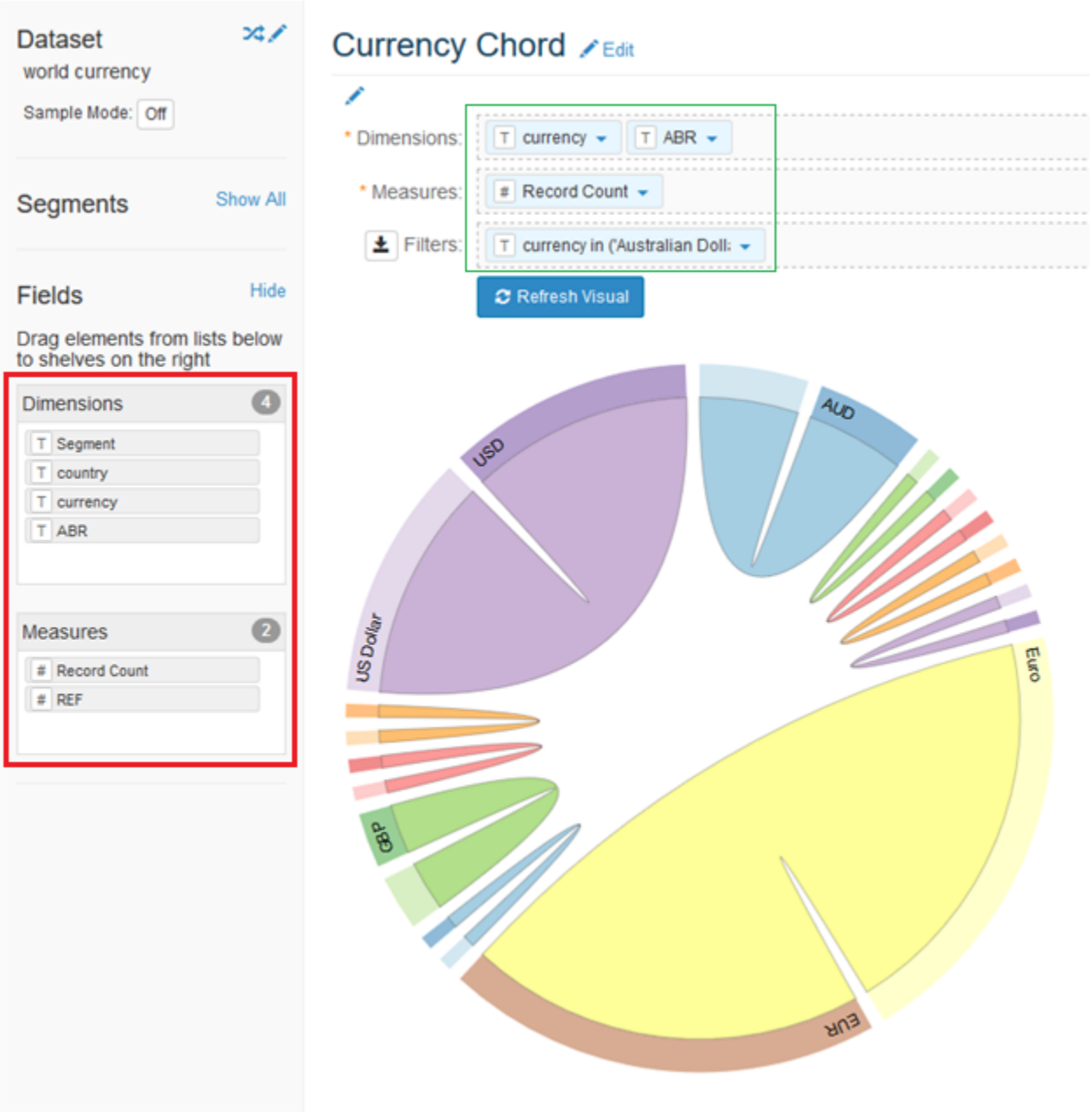


Figure 159: Creating visualizations in Arcadia Data

Arcadia is a new breed of tool that brings high quality visualizations to live, big data systems. With tools like Arcadia, you can do everything in one system; there's no ETL, as you don't have to move the data.

At first glance, Arcadia is very capable of connecting to live Hadoop instances and creating visualizations. We have looked at Arcadia 2.4, but when we revisit it, we'll look at Arcadia 5.0. We'll see if progress has been made by this tool designed to work with big data.

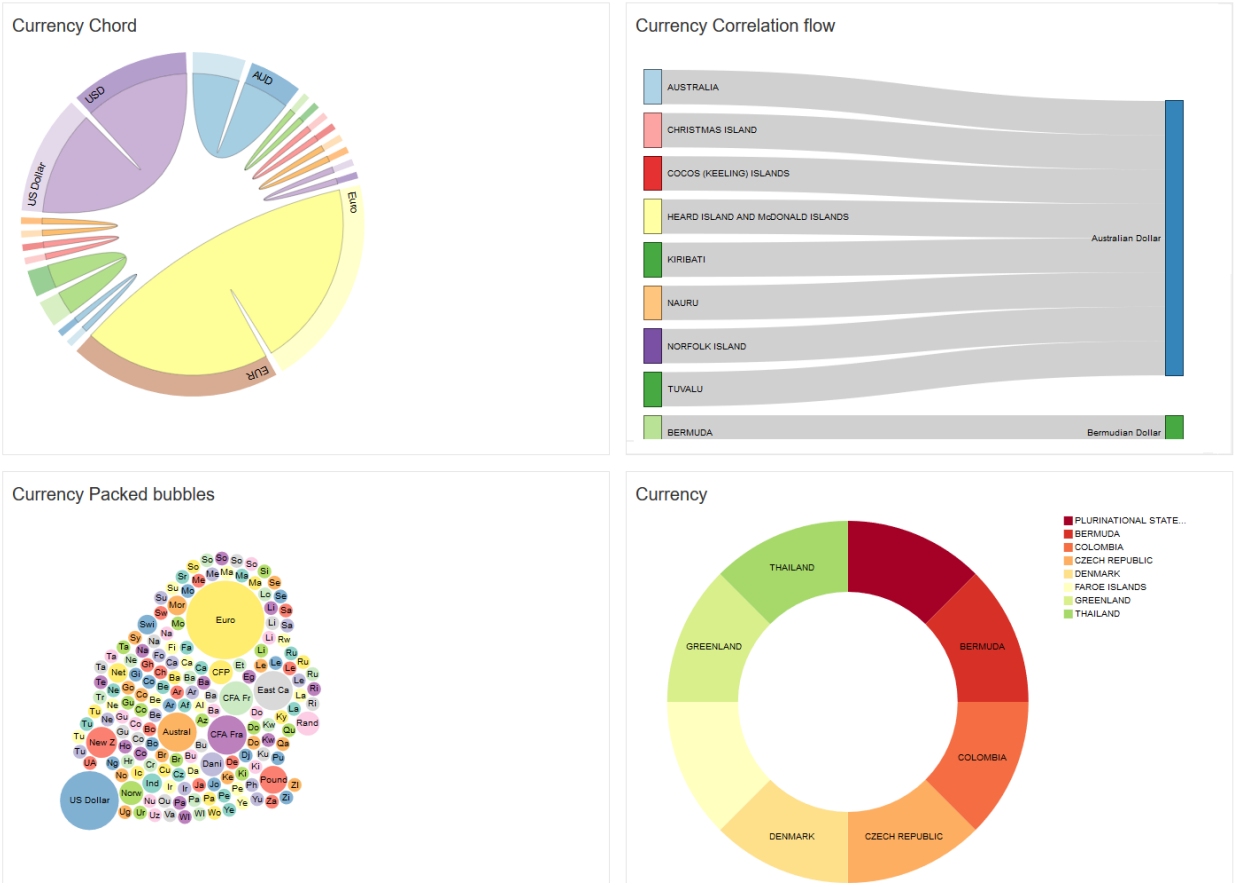


Figure 160: Arcadia dashboard visualizations

www.dbooks.org

Elasticsearch and Kibana

While it's known that Elasticsearch now uses the HDFS as a snapshot repository, it's less well-known that Kibana is being developed to ingest data commonly found in BI tools.

Many tools that were once the domain of Linux are now established on the Windows platform. Kibana and Elasticsearch are welcome additions to that group of products. The Windows installer may have made their installation painless, but launching both tools is still done via the command line. In Elasticsearch 6.6, the startup is automated by clicking the icon that appears after installation. You also have the option of installing it as a service with this release. You always run Elasticsearch before Kibana. After launching Elasticsearch, you'll see the following screen.

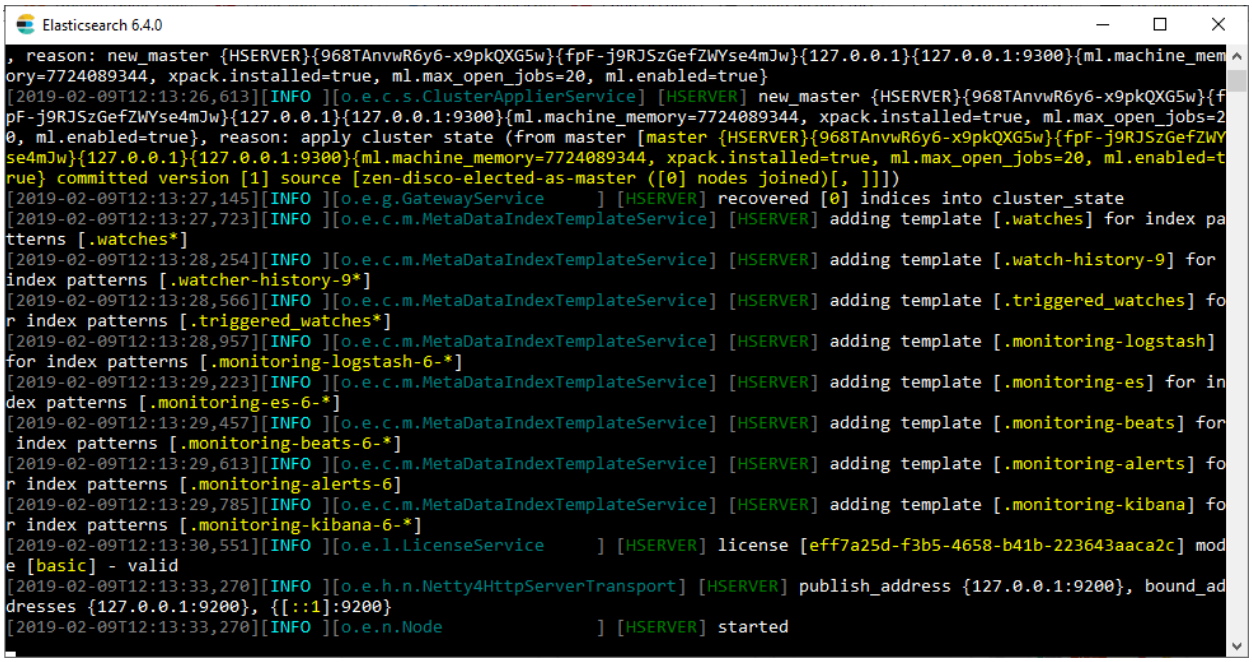


Figure 161: Elasticsearch started in Microsoft Windows

Now do the same for Kibana, but by running the .bat file in the bin folder of your installation. The next screen should then appear.

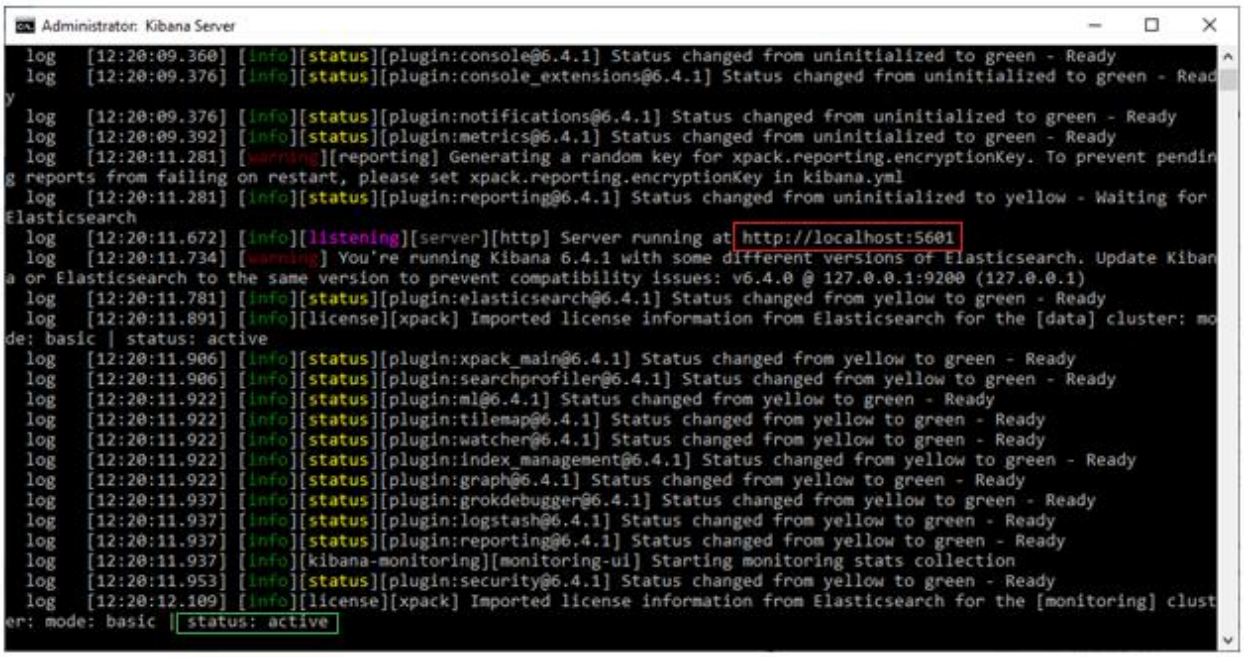


Figure 162: Kibana Server started and now active in Windows

The browser address to access Kibana is highlighted in red in Figure 162. The link shown is http://localhost:5601. After accessing the link, you'll see the screen shown in Figure 163. You're immediately invited to ingest data logs, operating system metrics, and much more. This is perhaps the Kibana and Elasticsearch sweet spot: ingesting data generated by the hour, minute, or second.

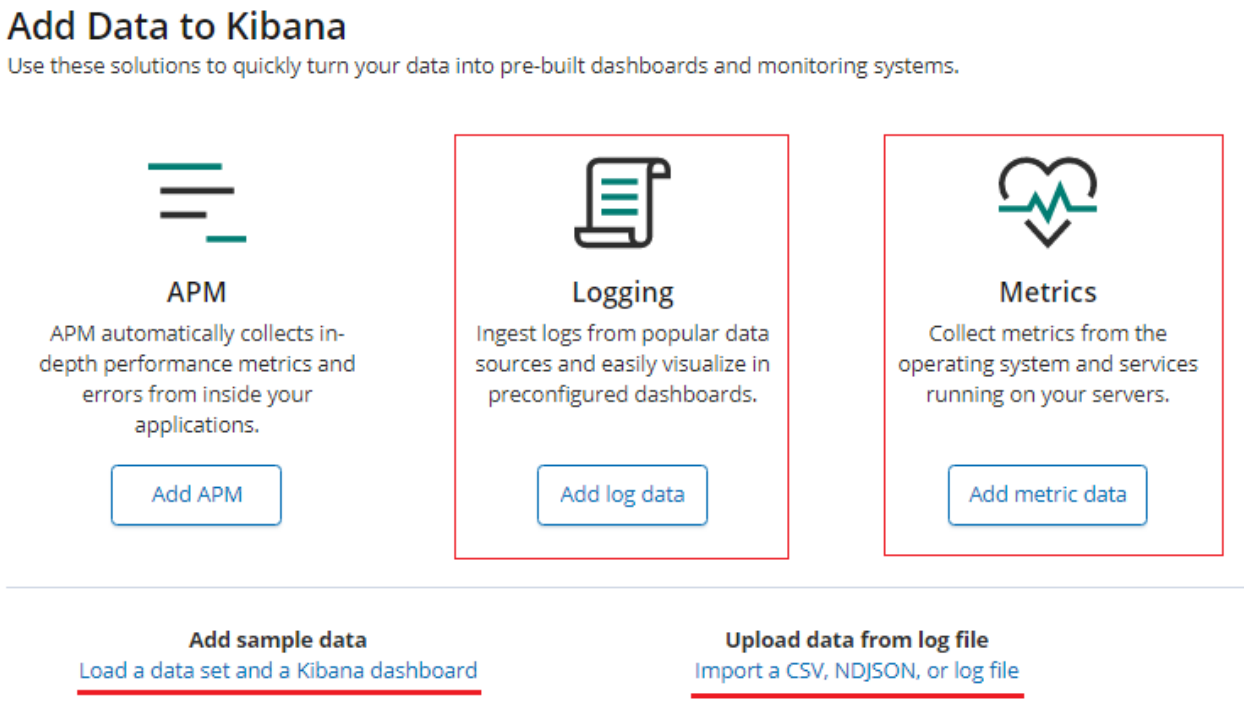


Figure 163: Kibana 6.6 main screen

www.dbooks.org

You can use the experimental feature to import data outside of the JSON format. These are perhaps the first steps in not just utilizing Hadoop for snapshots, but for "front-loading" Hadoop data for BI purposes. The experimental Import feature is shown on the following screen.

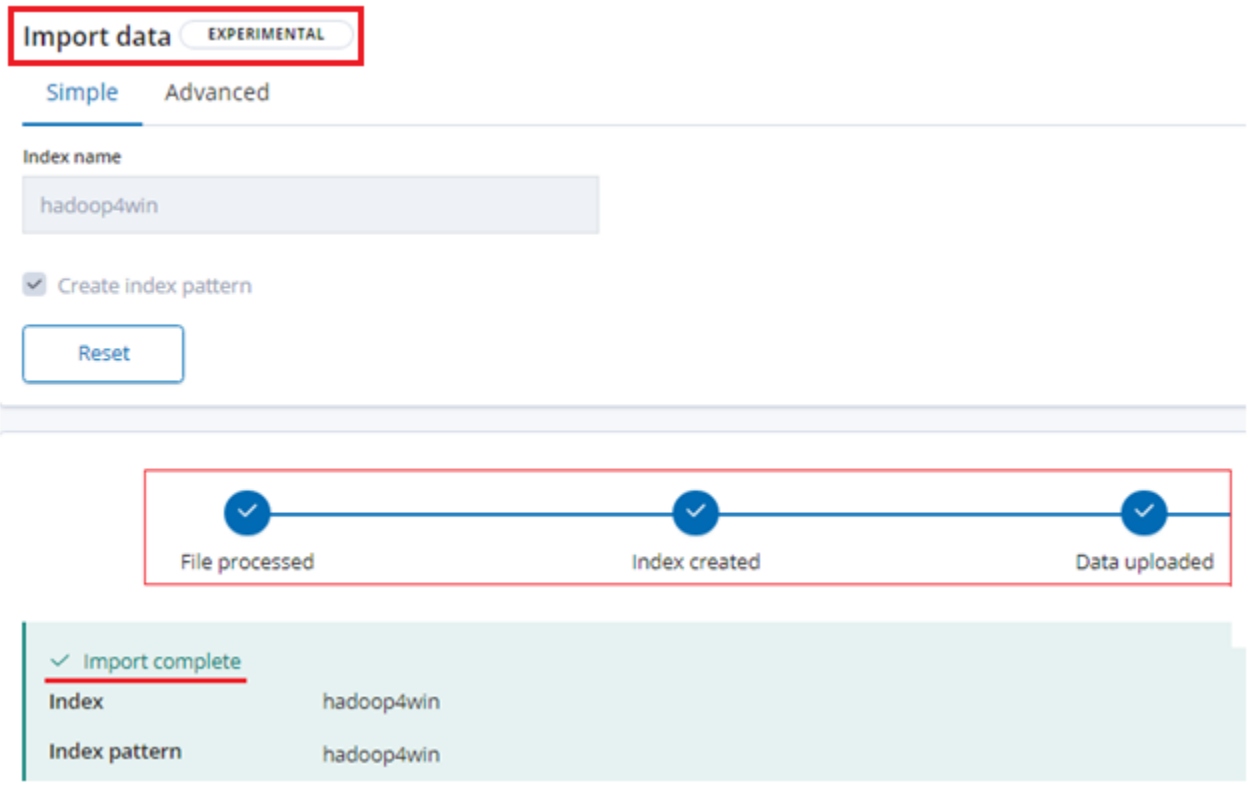


Figure 164: Front loading data into Kibana, the experimental Import data feature

This allows you to easily create outputs from a wider range of custom data sources in Windows.

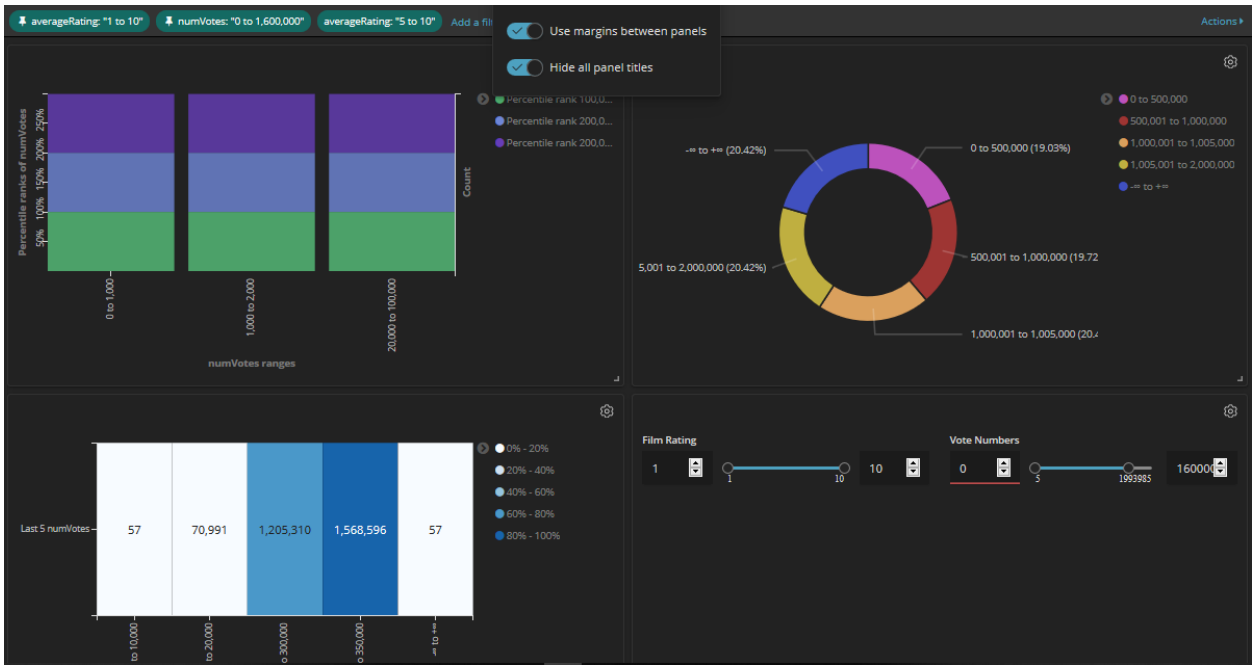


Figure 165: Kibana dashboard using data from experimental Import data tool

Syncfusion Dashboard Designer

The Syncfusion Dashboard Designer is made by the same company that produced the Syncfusion Big Data Platform. We'll look at Syncfusion Dashboard Designer in the final chapter, when we see how BI tools perform with larger data loads. Only the 4-GB Land Registry file we downloaded earlier will be used in the final section. First, let's look at elements of Hadoop in Linux that I'd like to see in Windows.

www.dbooks.org

Three features from Hadoop in Linux I'd like to see more of

While this is a book about Hadoop for Windows, we should be aware of Hadoop on Linux developments. In Cloudera (CDH), Impala is the default query editor engine. When you use Query Editor you immediately access Impala to write queries. Figure 166 shows the same query with joins that we ran earlier; it joins the Ratings table and the Episodes table. You could run this query in any version of Hive or Pig, and it would take a good few minutes. Our figure highlights the 4.31 seconds it took in Impala. The joined tables presented no problems and near-relational database speeds were achieved.

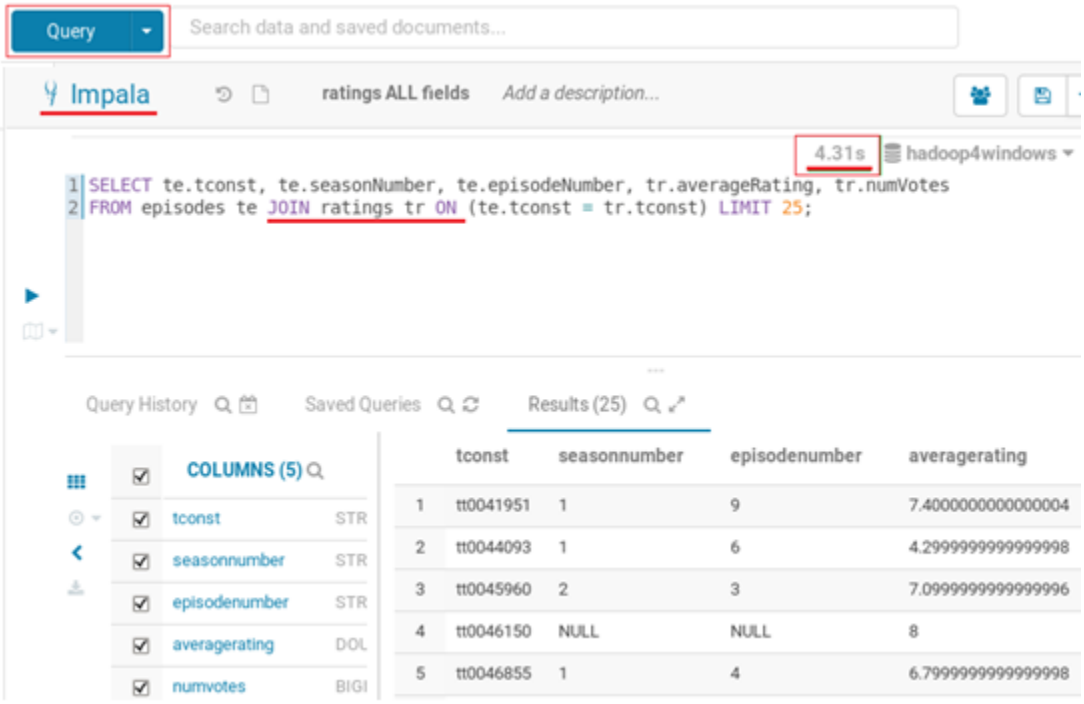


Figure 166: Impala running in Cloudera CDH on Linux

Cloudera offers basic graph and dashboard facilities within the Hadoop environment. You don't have to leave Hadoop to produce quick graphs and tables on Hadoop data. Some companies' tools are better than others, but it's a trend I expect to see continuing. The Cloudera offering is shown in Figure 167, and can auto-generate graphs from tables containing math elements. You can also build various visualizations from your chosen columns.

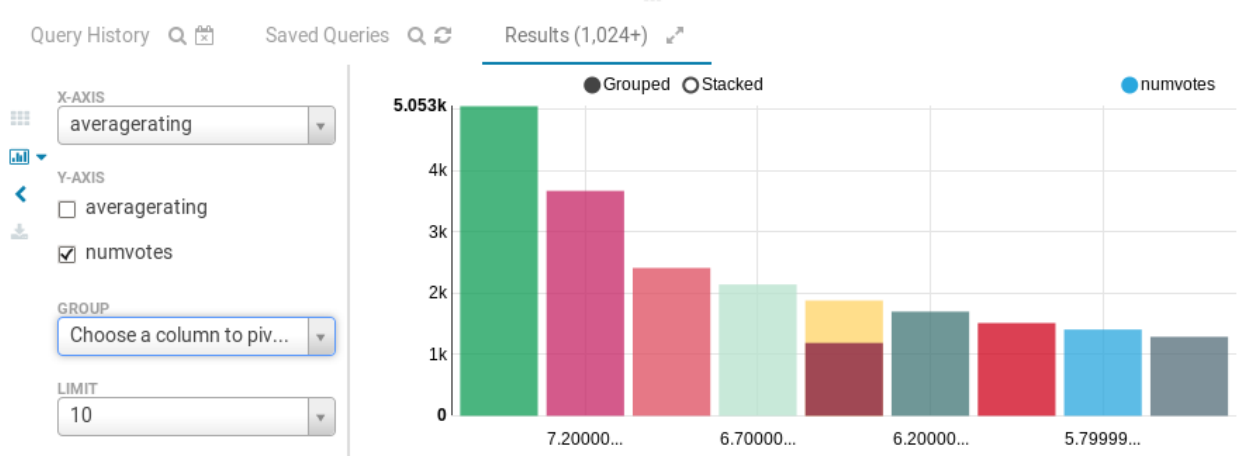


Figure 167: Basic visualization facilities in Cloudera CDH

The Hortonworks Linux platform has a function that would benefit any Hadoop distribution: automatic creation or removal of the first row as a header for data. The ability to define delimiters would also grace any Hadoop distribution.

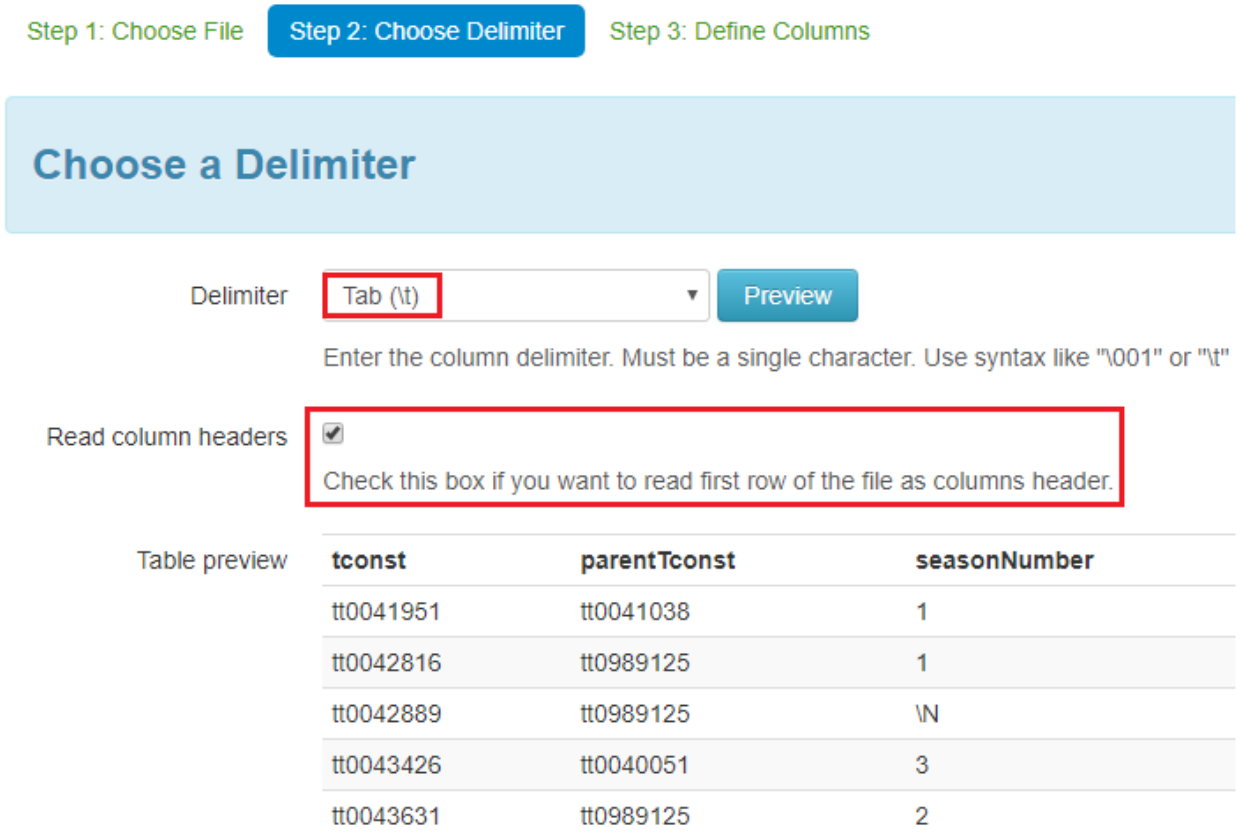


Figure 168: Automatic creation or removal of column header and defining of delimiters

www.dbooks.org

Chapter 5 When Data Scales, Does BI Fail?

Preparing the Dataset

We need to see how BI tools for Hadoop fare when faced with a larger data load in Windows. For this reason, we'll create the code for the 4-GB Land Registry database and its .csv file format.

Code Listing 23: Code for creating csv file format and Land Registry table in SQL Server 2019

```
-- CREATE EXTERNAL FILE FORMAT FOR CSV FILE IN SQLSERVER2019
USE Hadoop4windows
GO

CREATE EXTERNAL FILE FORMAT CSVformat WITH (
          FORMAT_TYPE =  DELIMITEDTEXT,
          FORMAT_OPTIONS

           (
                      FIELD_TERMINATOR =',',
           DATE_FORMAT = 'MM/dd/yyyy',
           STRING_DELIMITER = '"',
           USE_TYPE_DEFAULT = TRUE));

-- CREATE EXTERNAL TABLE FOR UK LAND REGISTRY DATA
USE Hadoop4windows
GO

CREATE EXTERNAL TABLE [dbo].[landregistrydata] (
          [transactionuid] nvarchar(100) NOT NULL,
          [price] int NULL,
          [dateoftransfer] nvarchar(100) NOT NULL,
          [postcode] nvarchar(100) NOT NULL,
          [properttype] nvarchar(100) NOT NULL,
          [newlybuilt] nvarchar(100) NOT NULL,
          [tenure] nvarchar(100) NOT NULL,
          [housenumorname] nvarchar(100) NOT NULL,
          [2ndhousenumorname] nvarchar(100) NOT NULL,
          [street] nvarchar(100) NOT NULL,
          [locality] nvarchar(100) NOT NULL,
          [towncity] nvarchar(100) NOT NULL,
          [district] nvarchar(100) NOT NULL,
          [county] nvarchar(100) NOT NULL,
          [transactiontype] nvarchar(100) NOT NULL,
          [recordstatus] nvarchar(100) NOT NULL)
```

```
    WITH (LOCATION='/ukproperty/',
          DATA_SOURCE = hadoop_4_windows,
          FILE_FORMAT = CSVformat
          );
```

For this code to work, please make sure the 4-GB Land Registry file is in a folder called **ukproperty**. To give the BI tools we're going to use a fair chance, we'll also create the table in Hive.

Code Listing 24: Code to create Land Registry table in Hive

```
-- Creating external table from Land Registry file in ukproperty folder
CREATE EXTERNAL TABLE IF NOT EXISTS landregistrydata02(transactionuid
string,price string,dateoftransfer string,postcode string,properttype
string,newlybuilt string,tenure string,
housenumorname string,2ndhousenumorname string,street string,locality
string,towncity STRING,district STRING,county STRING,transactiontype
STRING,recordstatus STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED by '\n'
stored AS textfile;
Load data INPATH '/ukproperty/pp-complete1.csv'  into table
landregistrydata02;
select * from landregistrydata02 LIMIT 25;
```

The tools we are going to use are: Tableau, Azure Data Studio, Arcadia Data, and Syncfusion Dashboard Designer.

We already know the limitations of QlikView Direct Discovery, and with Azure Data Studio and SQL Server, Microsoft products are already represented.

Using Tableau with large datasets in Windows Hadoop

I used the Hortonworks Hive Connection for Tableau after installing Hive OBDC 2.1.16 driver.

www.dbooks.org

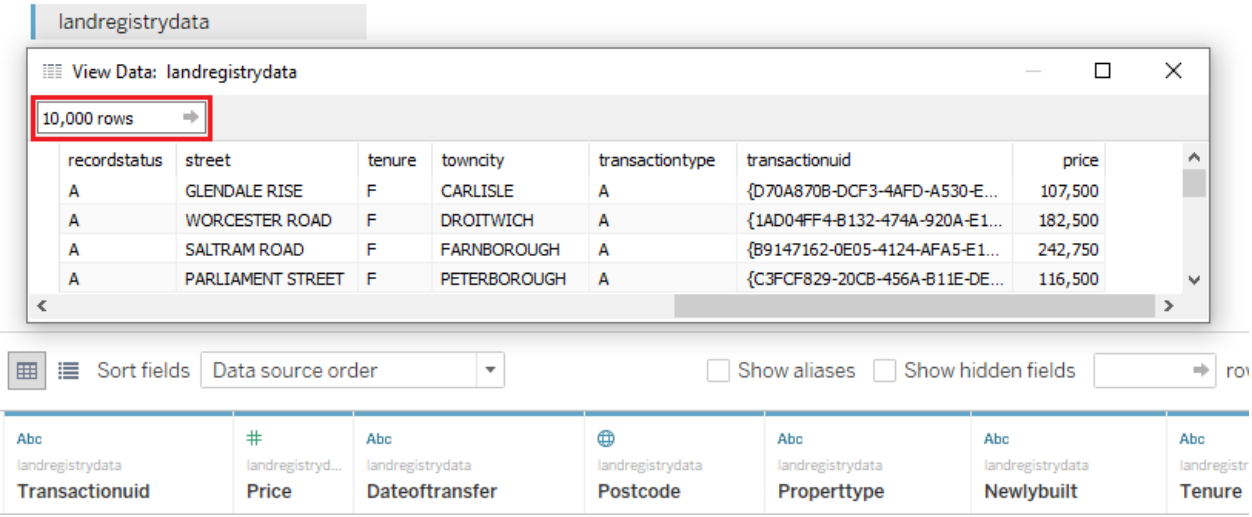


Figure 169: Tableau connecting live to 4-GB Land Registry file in Hive

The Tableau live connection was immediate and fast, automatically previewing 10,000 rows in a preview window with ease. I certainly don't need a data preview for anything that big, but it was impressive to see. I was able to use any visualization I wished to without adverse effects on performance. Overall, it's an easy pass for Tableau, though it would be good to see one or two new visualizations. As you work with larger amounts of data, standard visualizations don't always provide the best presentation choices. The Tableau visualization is shown in the following figure.

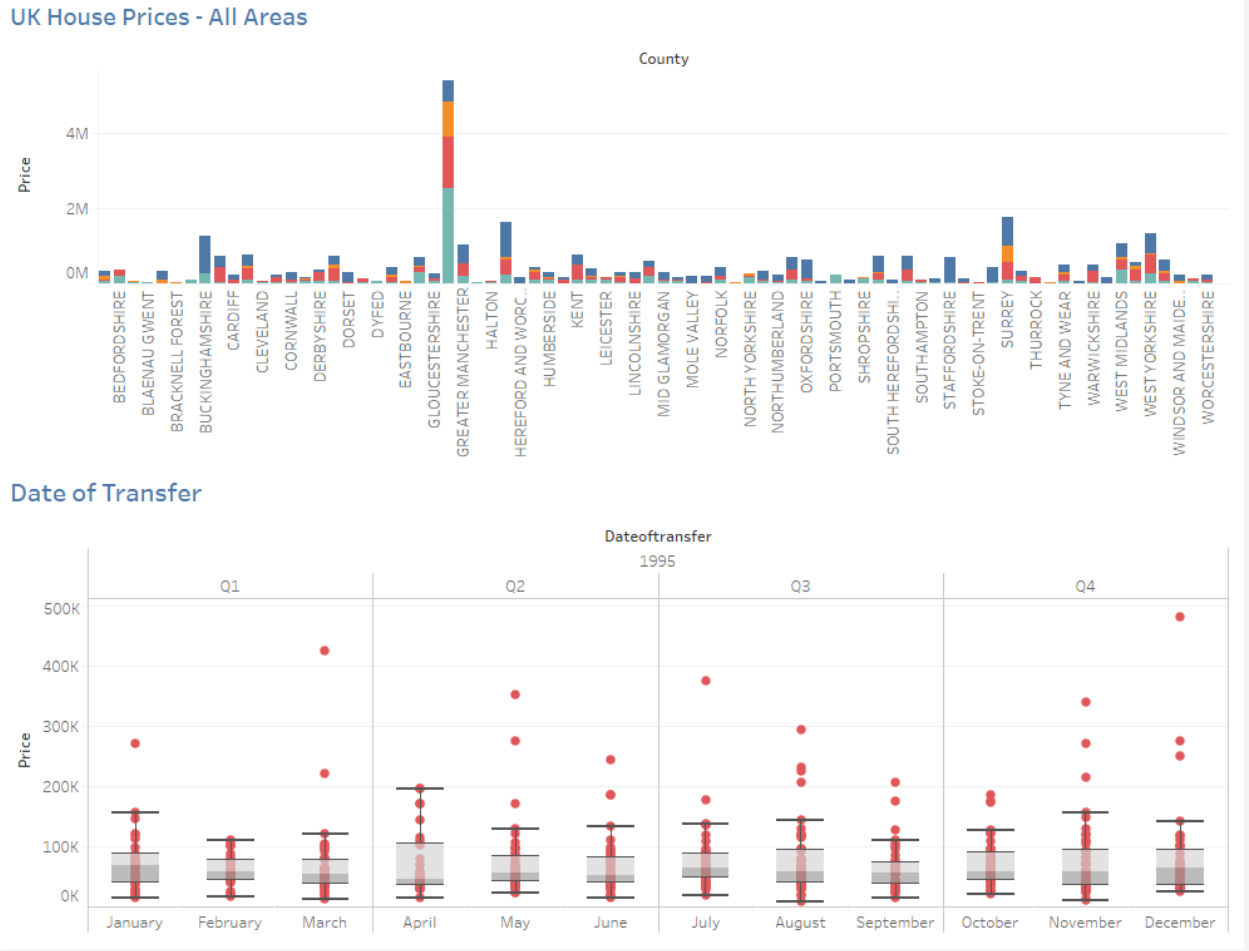


Figure 170: Tableau visualization using Land Registry data

www.dbooks.org

Using Azure Data Studio with large datasets in Windows Hadoop

Azure Data Studio pulled back the data in almost effortless fashion. The highlight was the easy use of IntelliSense on a file that big. A query with a ***where*** clause, as shown in Figure 171, works with IntelliSense. There is no delay in predicting the next word you type, or identifying words in the system. As before, there is no connection via Hive, but directly to HDFS from SQL Server. I can't find fault with the performance of this tool, and the application is compatible with the Syncfusion platform. As good as Azure Data Studio is, it's only as solid and reliable as the Hadoop system it's connected to.

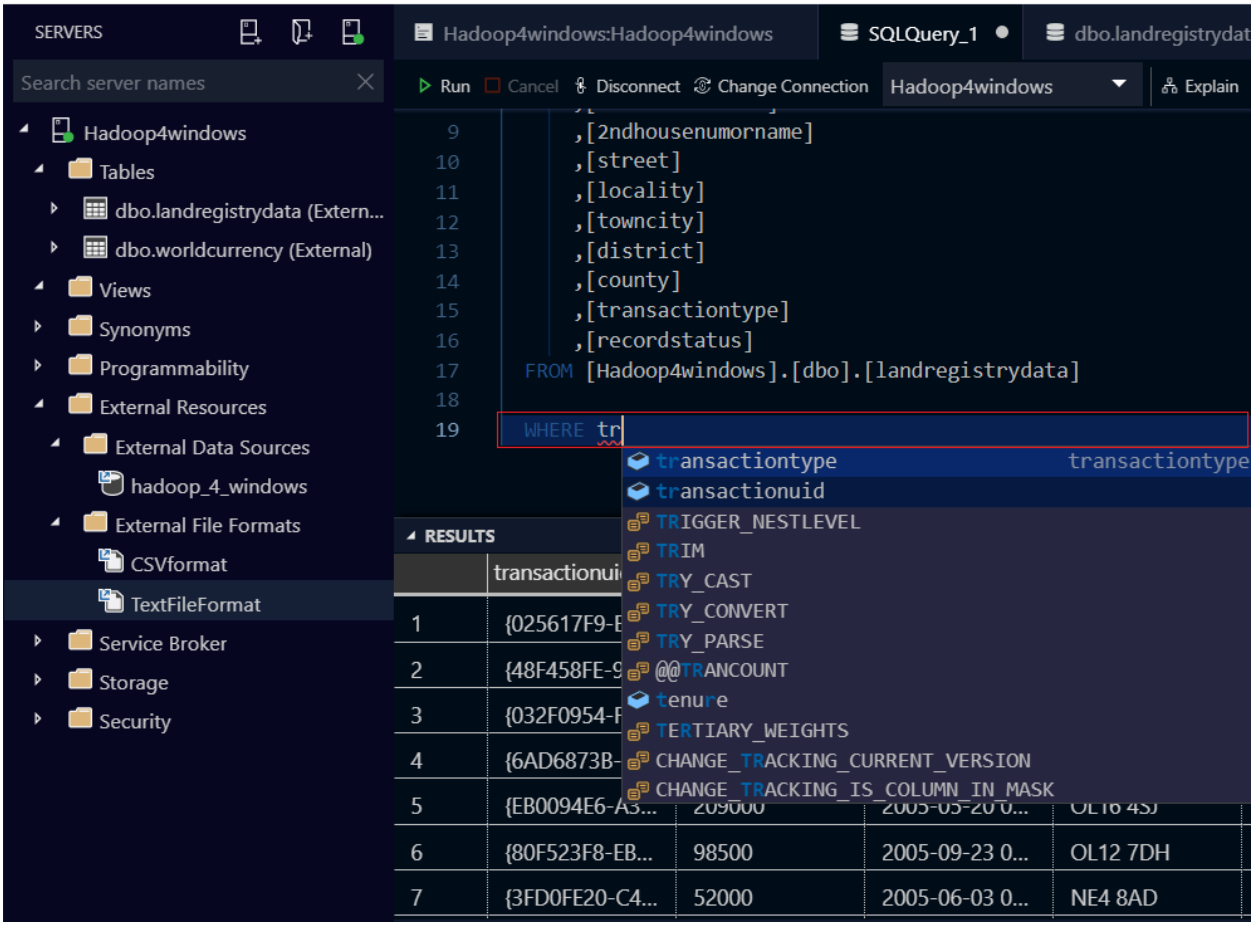


Figure 171: Azure Data Studio making a fast connection to Land Registry data file in Hadoop

Perhaps because of the small size of the app, I was expecting some kind of decrease or degradation of performance. That didn't happen though, and it's clear that Microsoft has laid the foundations for much smoother big data experiences. The word I would probably choose to describe the performance of this tool overall is “effortless.” Windows Server never felt like it was under any strain at all.

Using Arcadia Data with large datasets in Windows Hadoop

As stated earlier, we are now using Arcadia Data 5.0. This version of the software has a feature called Direct Access, as shown in Figure 172. Direct Access enables you to query a Hadoop data source live. In this case, Hive is being queried with SQL from the Arcadia Direct Access window. The data at the bottom of the screen is the data returned by the query. This is without question the fastest Hadoop query tool I've seen—it's as if you're in Hadoop itself. You may find that it returns queries faster than queries written inside Hadoop. With the data returned you can either create a dataset to create visualizations, or download it as a .csv file.

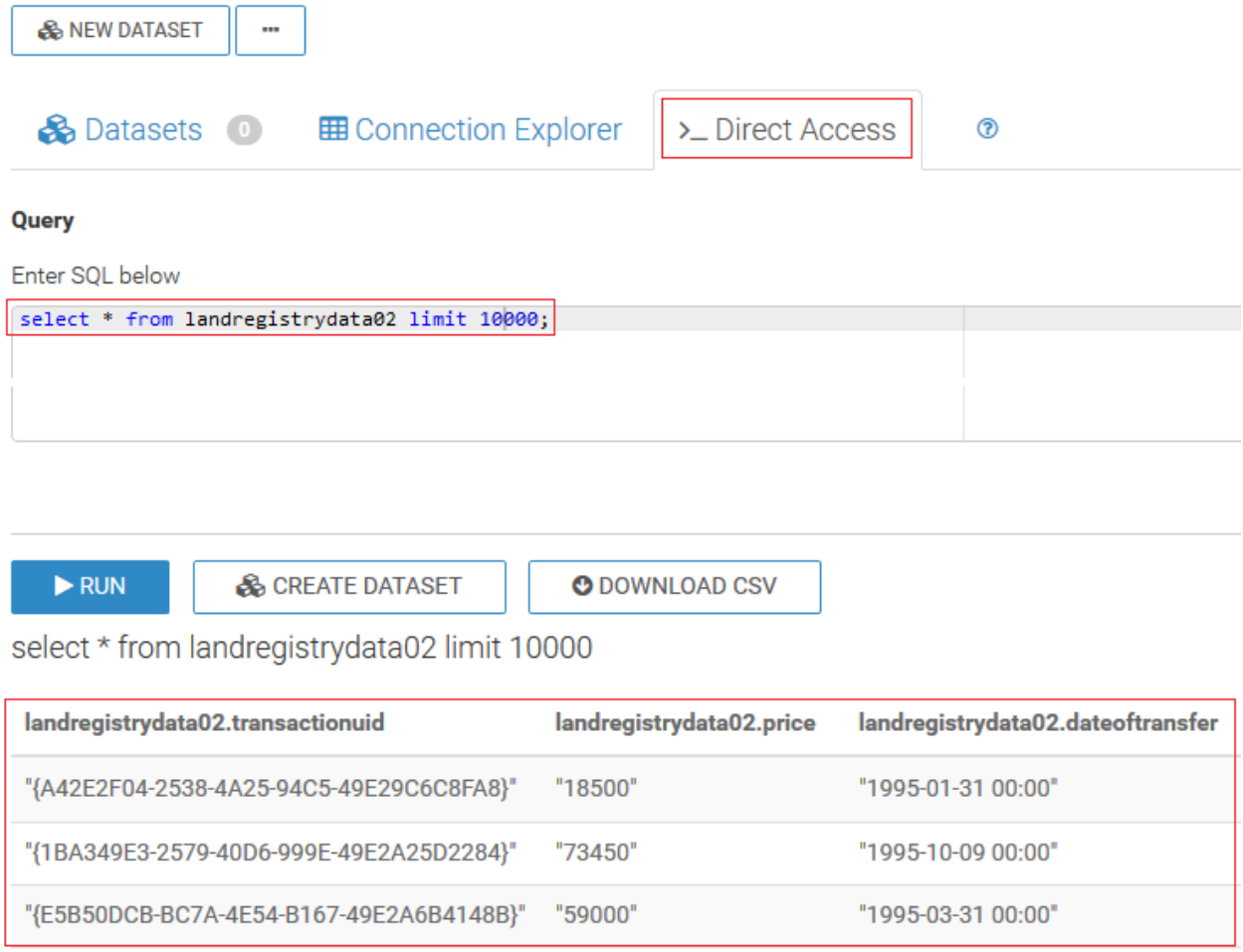


Figure 172: Direct Access in Arcadia 5.0

www.dbooks.org

Arcadia has a number of features that enable you to sustain long sessions with large amounts of data. The following figure shows the "Clear result cache" facility, something you don't see often in BI tools.

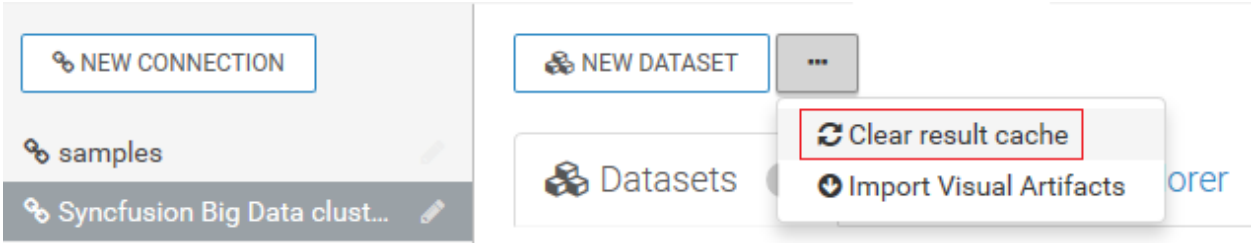


Figure 173: Clear result cache feature in Arcadia

Another feature is the sample mode, which lets you control how much data you work with in percentage terms. You can also limit the number of rows retuned in the preview. This prevents you from "flooding" the application with data, and ensures the screen redraws faster.

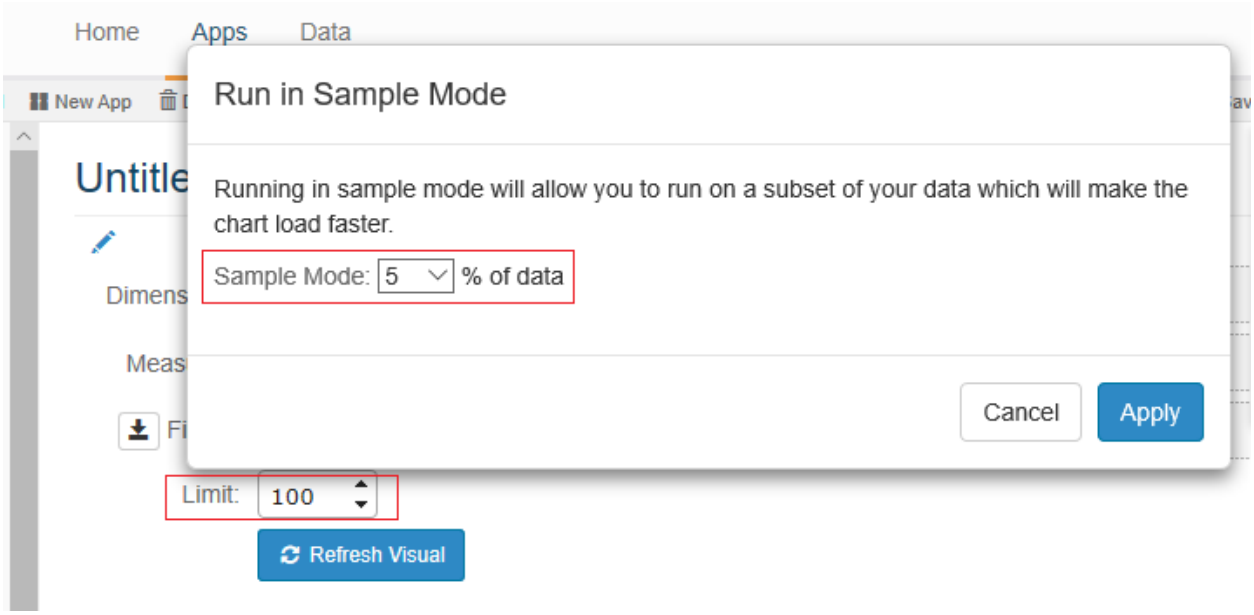


Figure 174: Sample Mode in Arcadia Data

In order to analyze data and create visuals, you create datasets from your tables. This just involves clicking **New dataset** in the same row that your table is shown in. This is portrayed in the following figure.

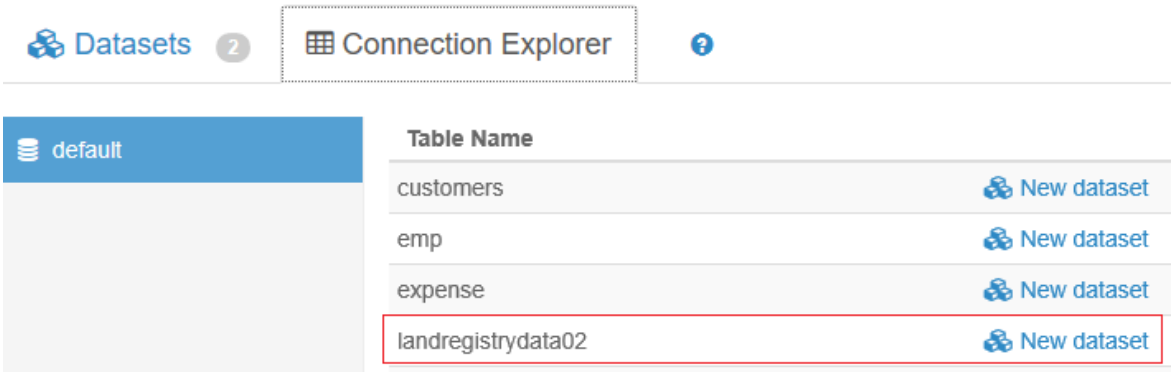


Figure 175: Creating a new dataset from a table

You can now create a dataset from your visual by clicking the **New Visual** link.

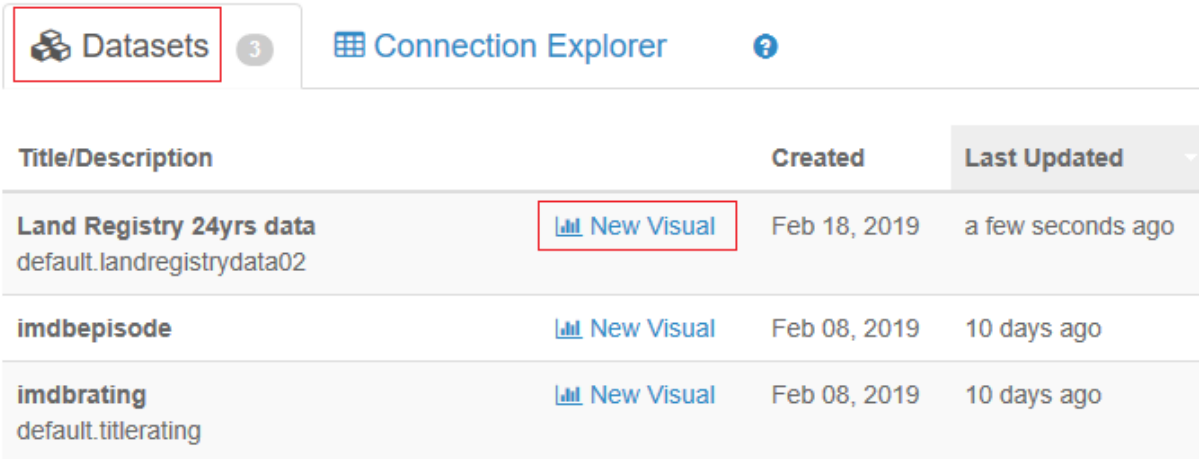


Figure 176: Creating visuals from datasets

We can create visualizations by dragging the **Dimension** and **Measures** fields highlighted in the red box shown in Figure 177 onto the **Dimensions** and **Measures** fields highlighted in the green rectangles.

www.dbooks.org

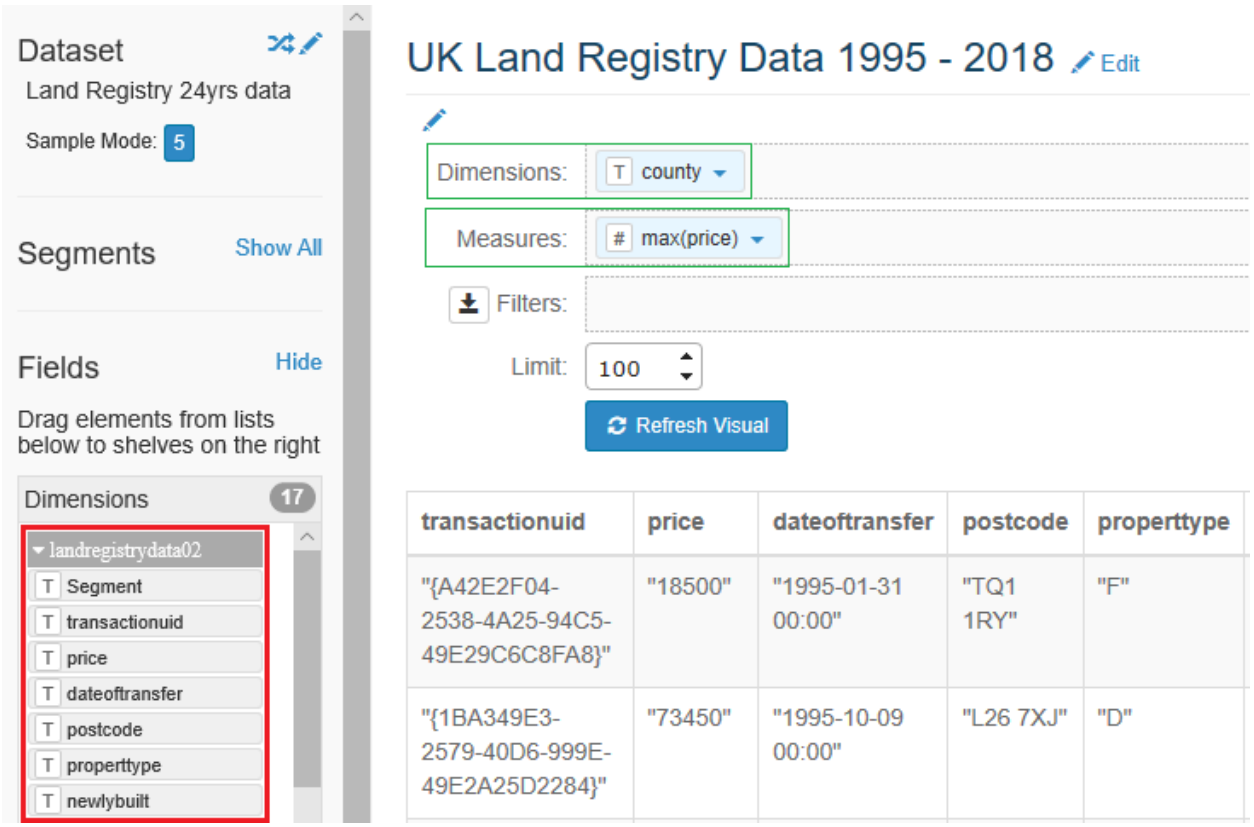


Figure 177: Choosing dimensions and measures for our visualizations

We can filter the field data and store the choice to create segments.

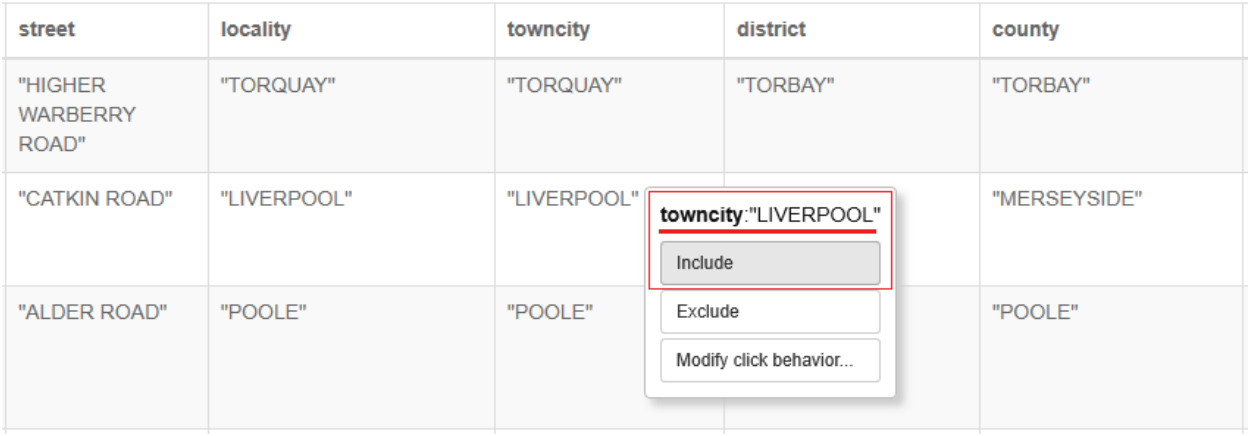


Figure 178: Filtering data for the city of Liverpool

Arcadia Data has a wide variety of visualizations, as seen in the following two figures.

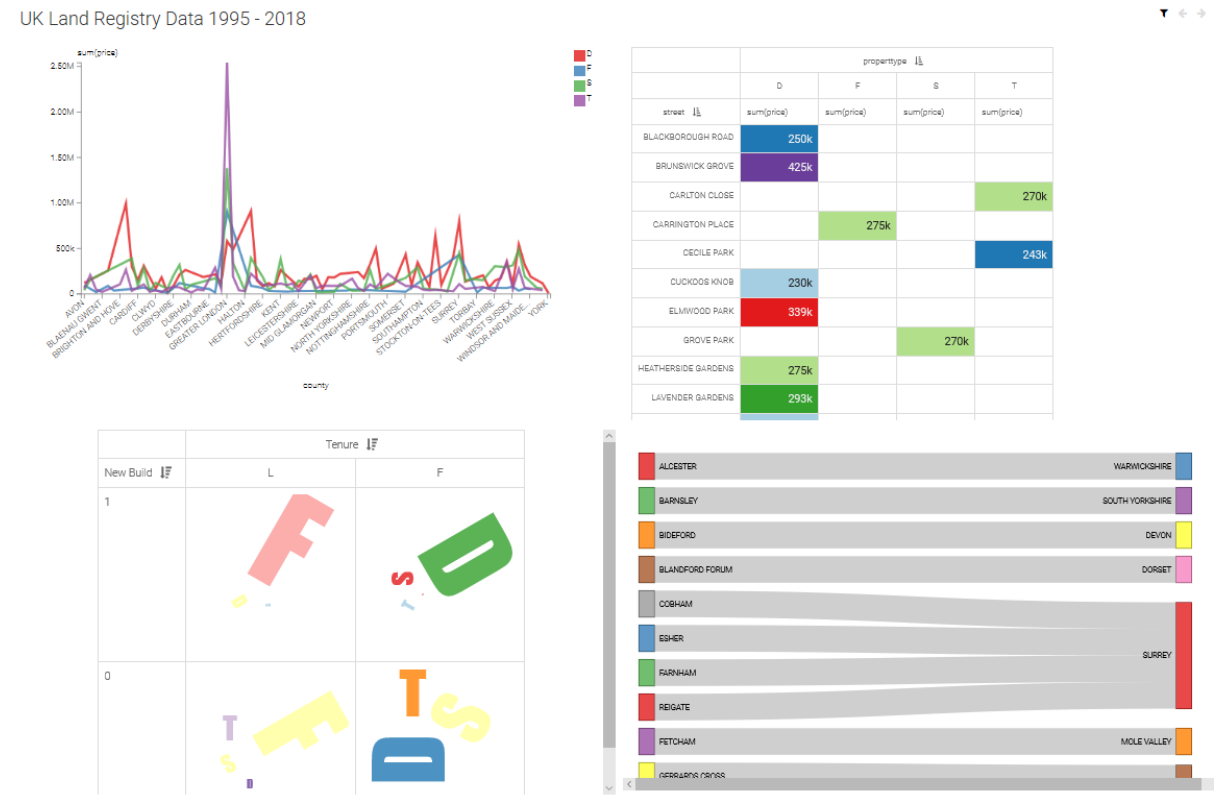


Figure 179: Dashboard design with Arcadia Data

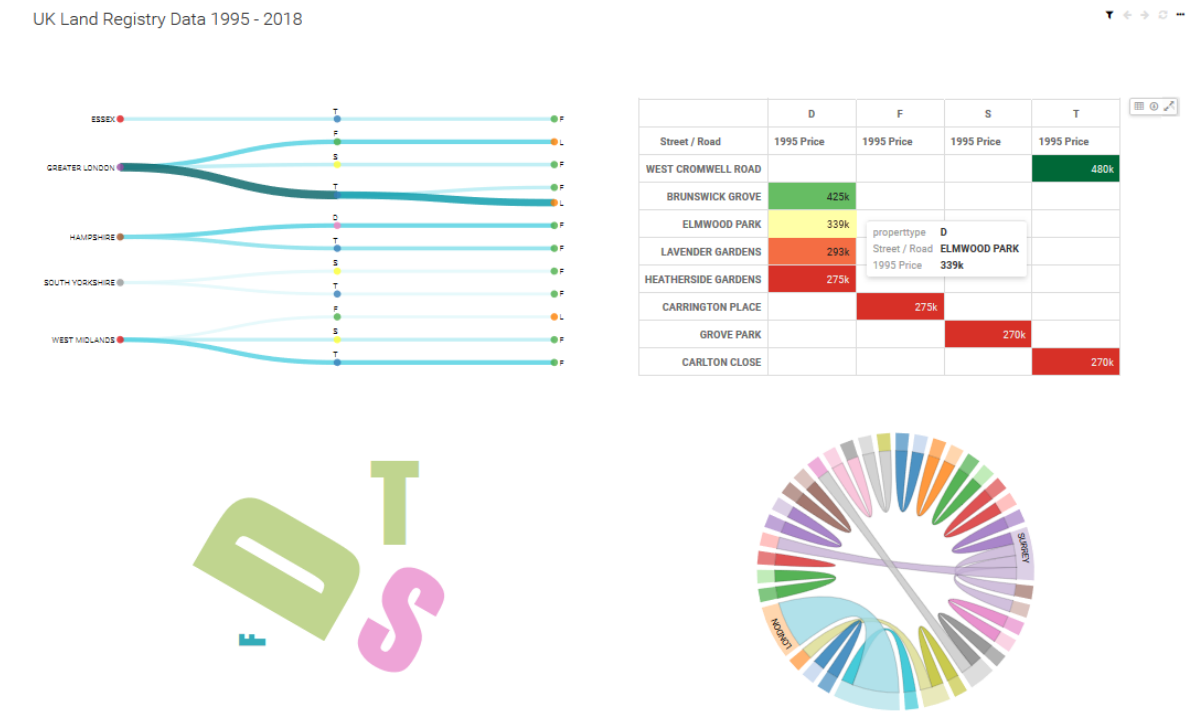


Figure 180: A second dashboard created using Arcadia Data

www.dbooks.org

Using Syncfusion Dashboard Designer with large datasets in Windows Hadoop

When you start the Syncfusion Dashboard Designer, you see what is effectively a blank canvas.

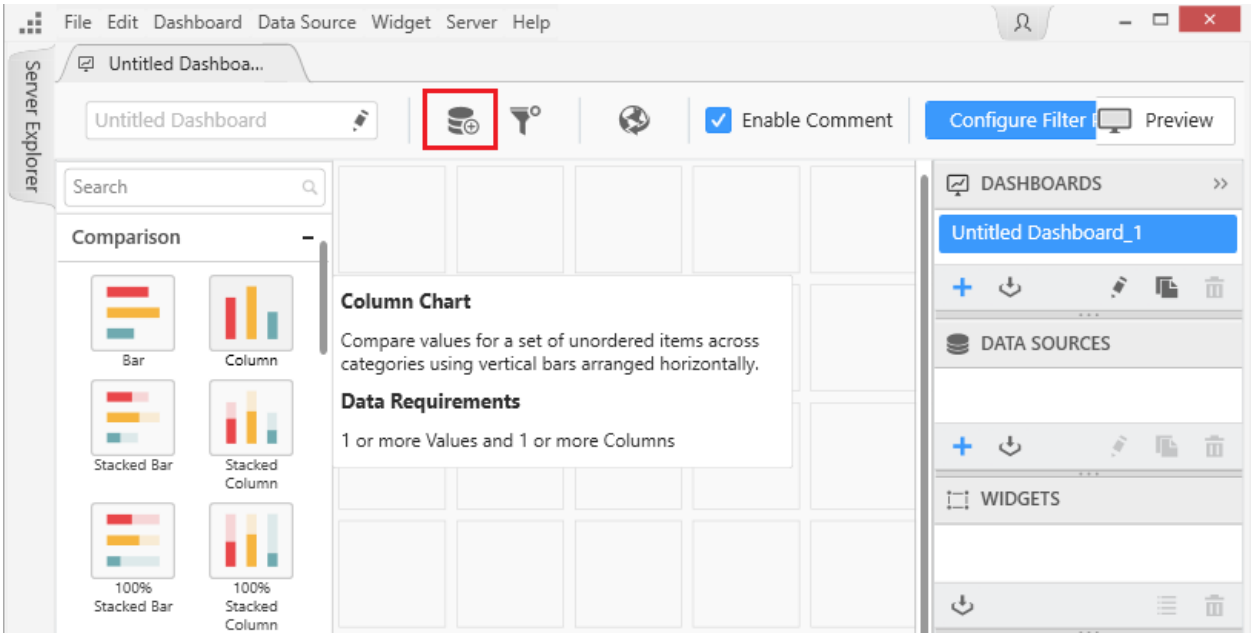


Figure 181: The blank canvas of Syncfusion Dashboard Designer

In Figure 181, the icon highlighted in red is the button for creating a data source. To connect to SQL Server 2019, you simply fill out the form shown in the following figure, then click **Connect**.

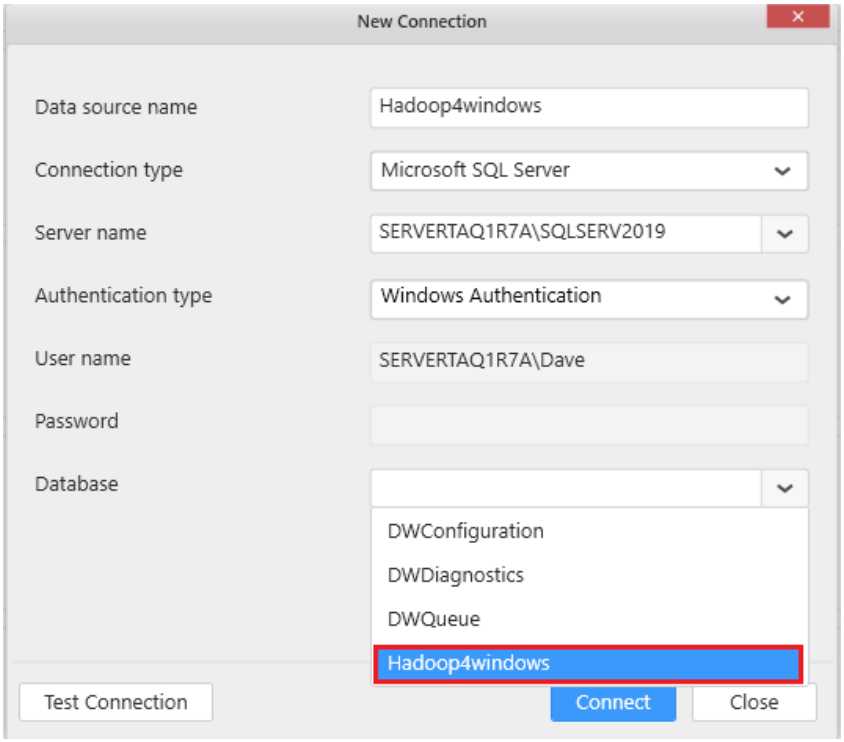


Figure 182: Connection to SQL Server 2019

If the tables you want to work with are in Hive, you can use a Hive connection or a faster Spark SQL connection. This will pick up the tables in the "default" database when you log in.

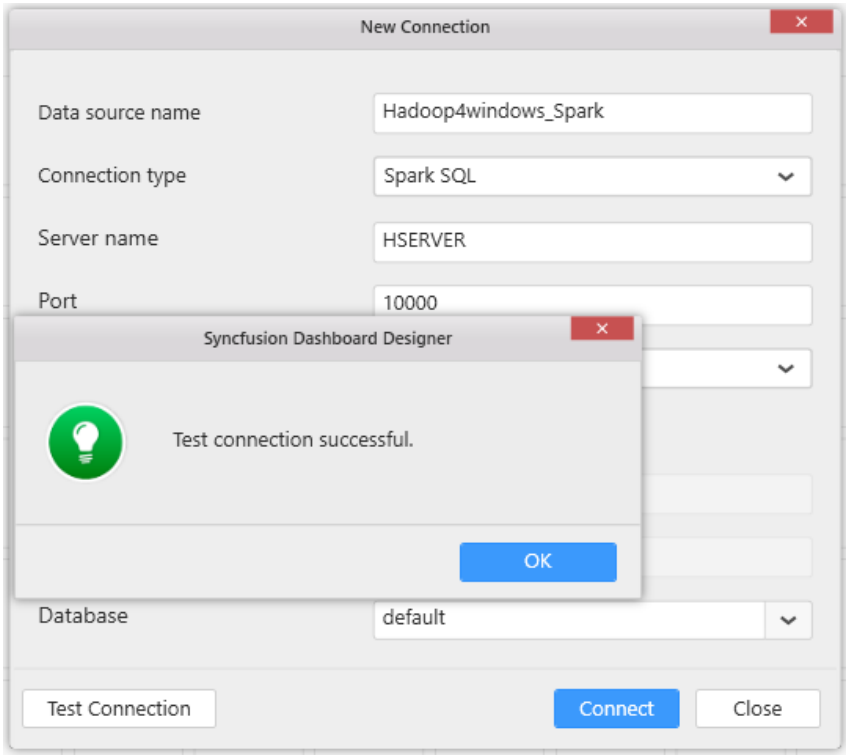


Figure 183:Spark SQL login

www.dbooks.org

You instantly connect to a blank canvas, and on the left side, you see the tables within SQL Server. You follow the onscreen message to drag and drop tables to create a virtual table.

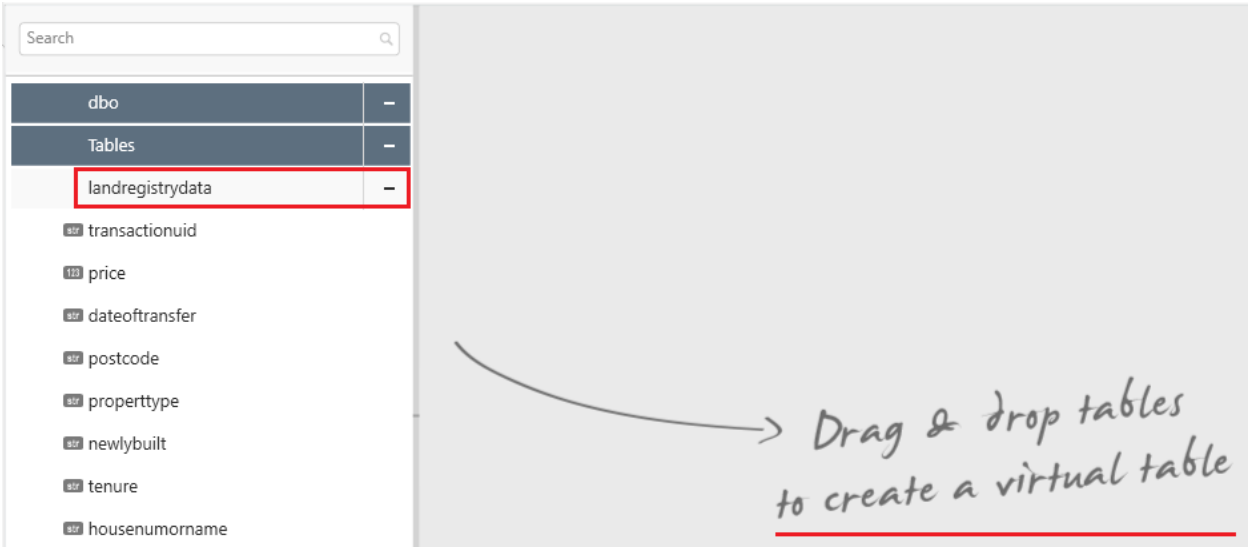


Figure 184: Invitation to drag and drop tables to create a virtual table

When you drag your table to the canvas, you can add columns as you wish. The application does not overload the table with data from the data source.

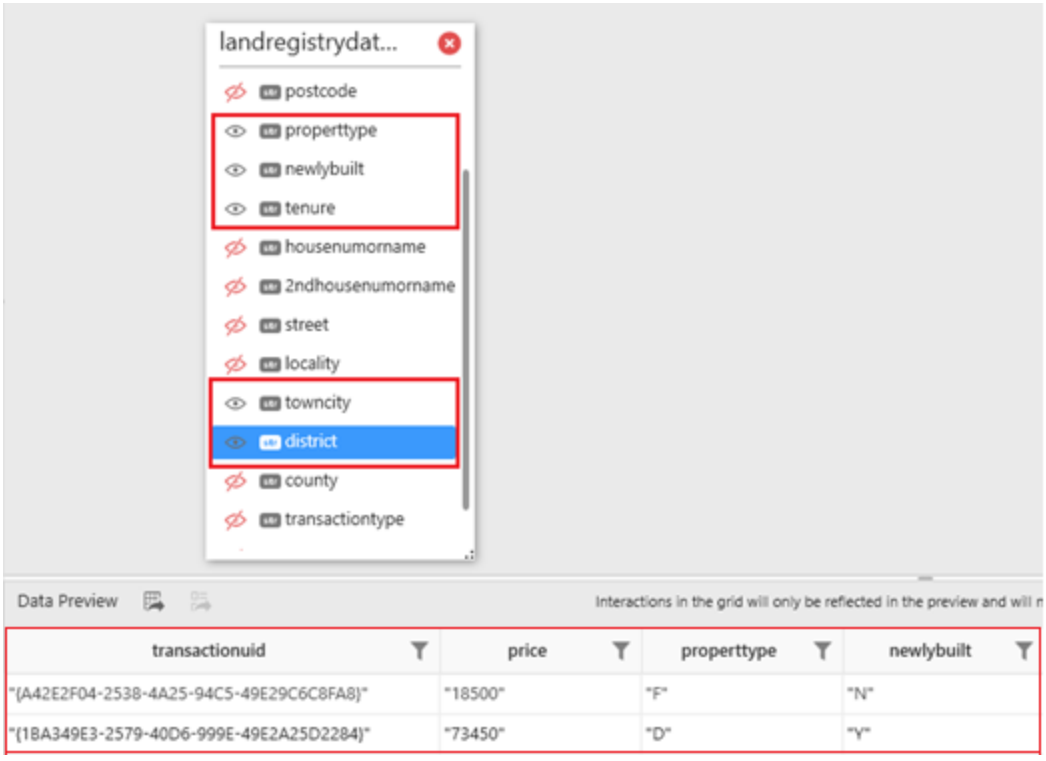


Figure 185: Selecting columns to add to the virtual table

You have the ability to select, deselect, and search records, as shown in the following figure.

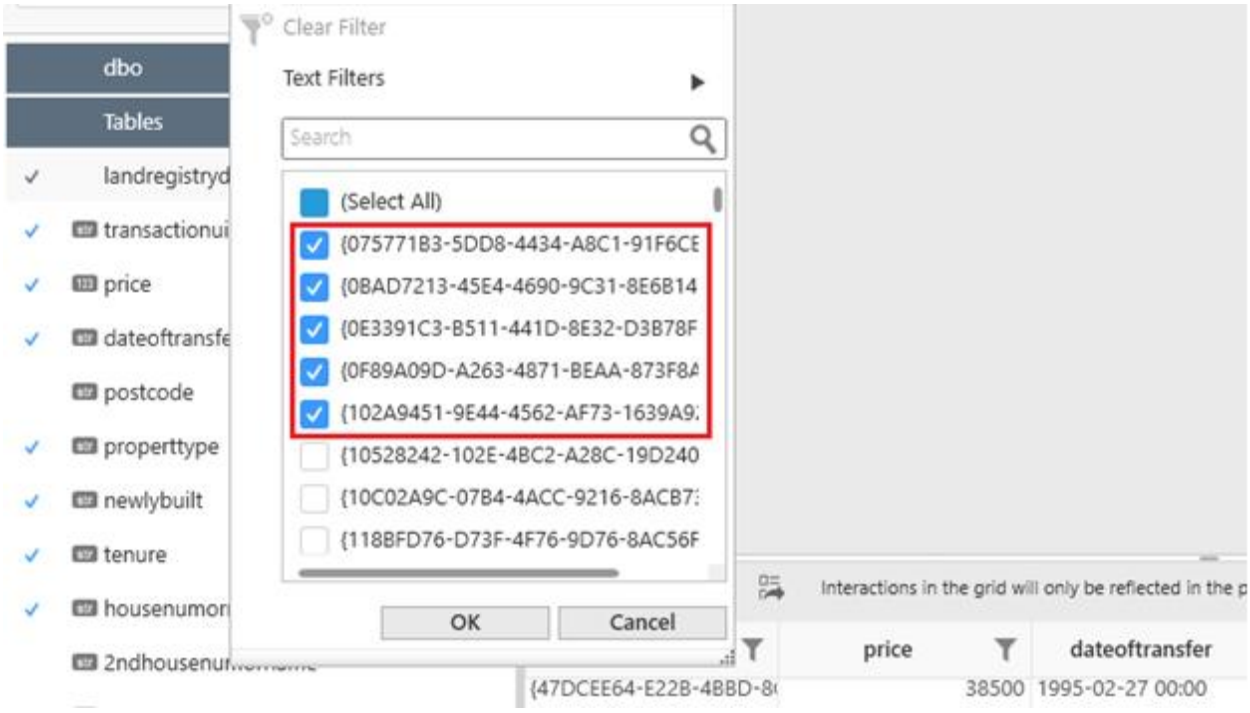


Figure 186: Filtering records you do or don't wish to select

You can change column types, rename columns, and carry out aggregation functions on the fly.

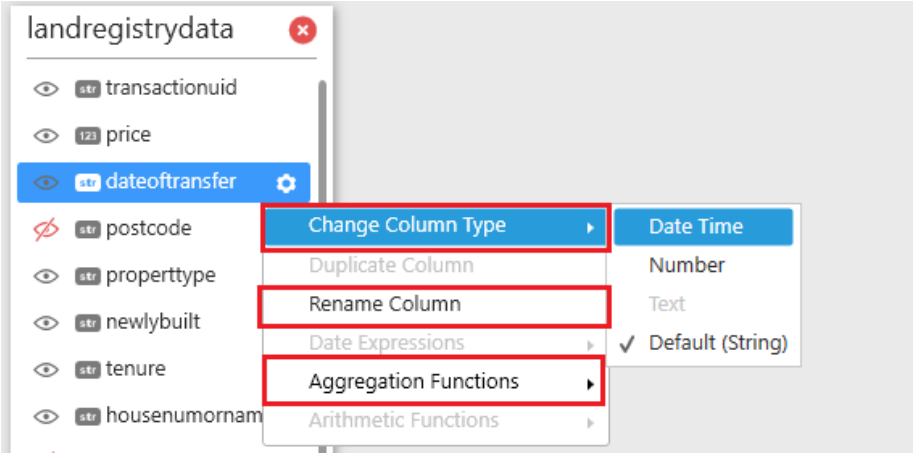


Figure 187: Change of column types, names and aggregation functions

You are now ready to create a visualization. Click a graph icon on the Dashboard tab.

www.dbooks.org

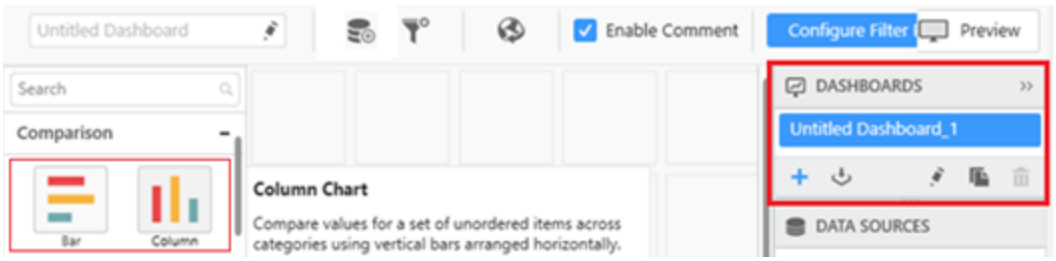


Figure 188: Bar and Column graph icons on the Dashboard tab

You create graphs by dragging Measures and Dimension fields to values, columns, and row fields. If large amounts of data are being loaded, a warning appears on screen, inviting you to filter the data. This warning is shown in the following figure.

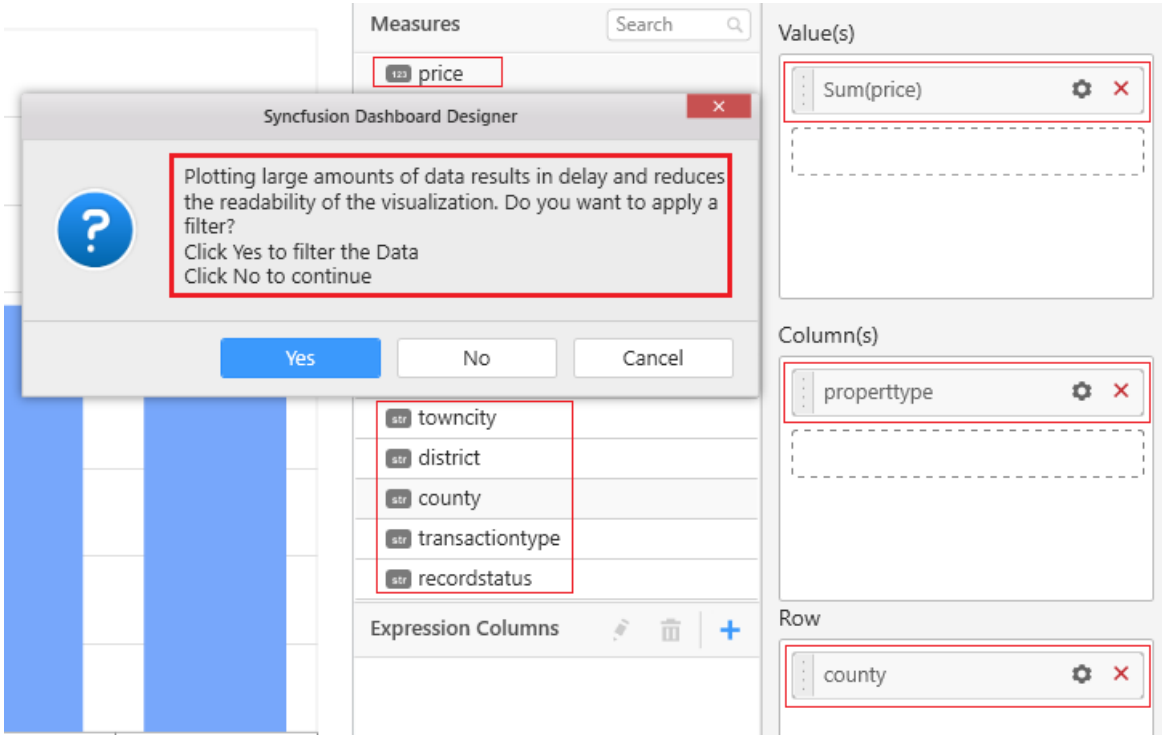


Figure 189: Adding measures and dimensions to create visualizations

You can then filter the data based on your selections.

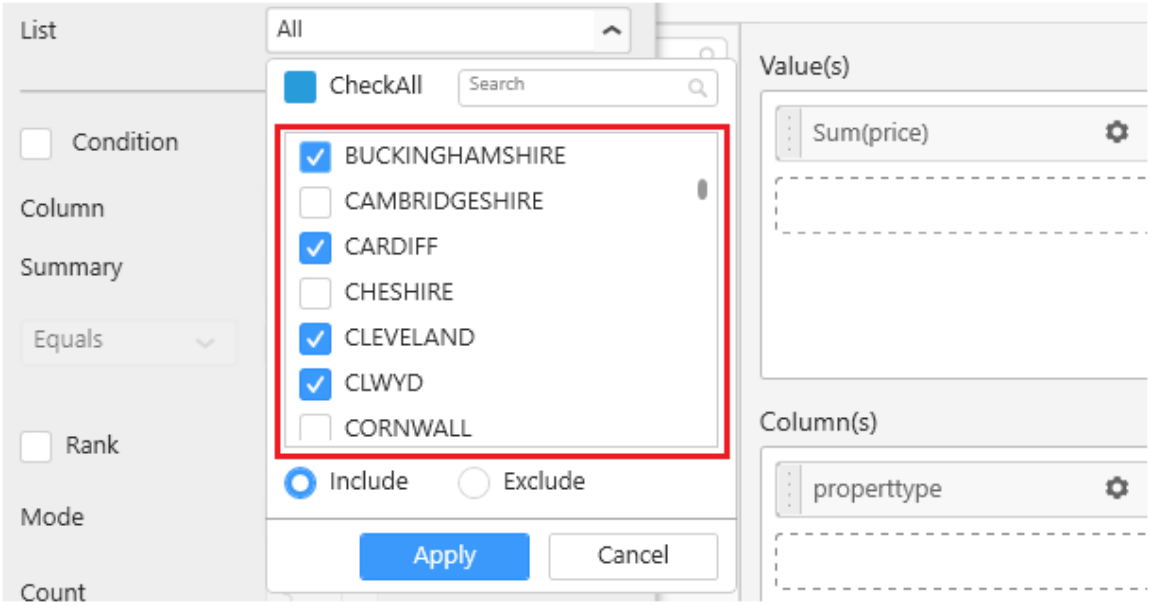


Figure 190: Filtering the records in your dataset

If you wish, you can use the Rank feature; this example ranks the top 10 average house prices.

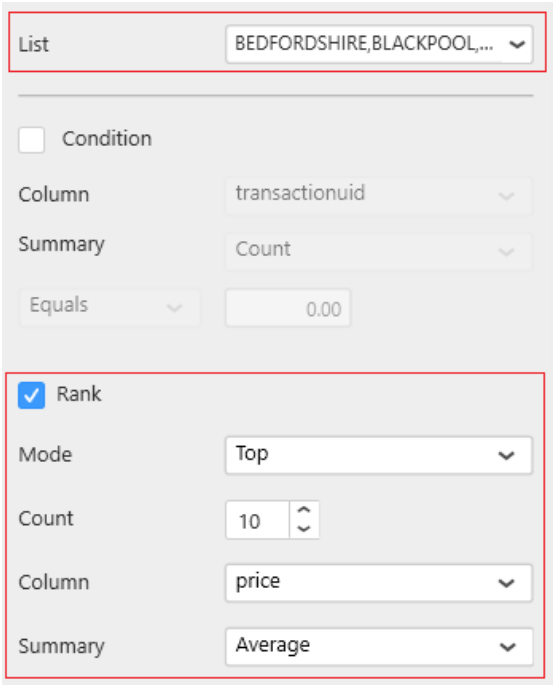


Figure 191: Top 10 average house prices by rank

We can now create visualizations with our filtered dataset. Dashboard speed is maintained by preventing unnecessary data from overwhelming the dashboard.

www.dbooks.org

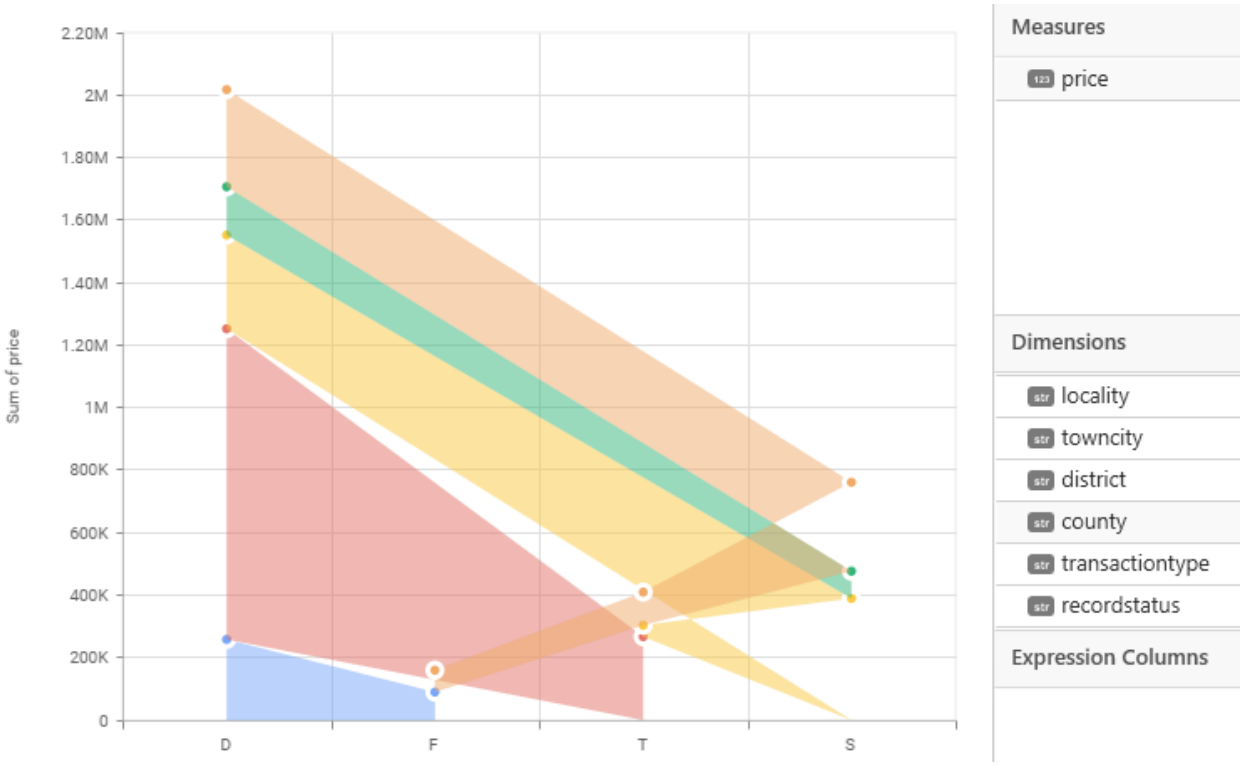


Figure 192: Creating visualizations from our filtered data

We can put together several visualizations to create dashboards; this includes pivot table visualizations shown in Figure 193. I often don't use pivot table visualizations, as the dashboard creation tools can't manage the data. You have seen throughout the use of this tool that it has been designed from the ground up to manage large amounts of data. That said, Syncfusion is the designer of the Big Data Platform, so the excellent performance of the tool here isn't surprising.

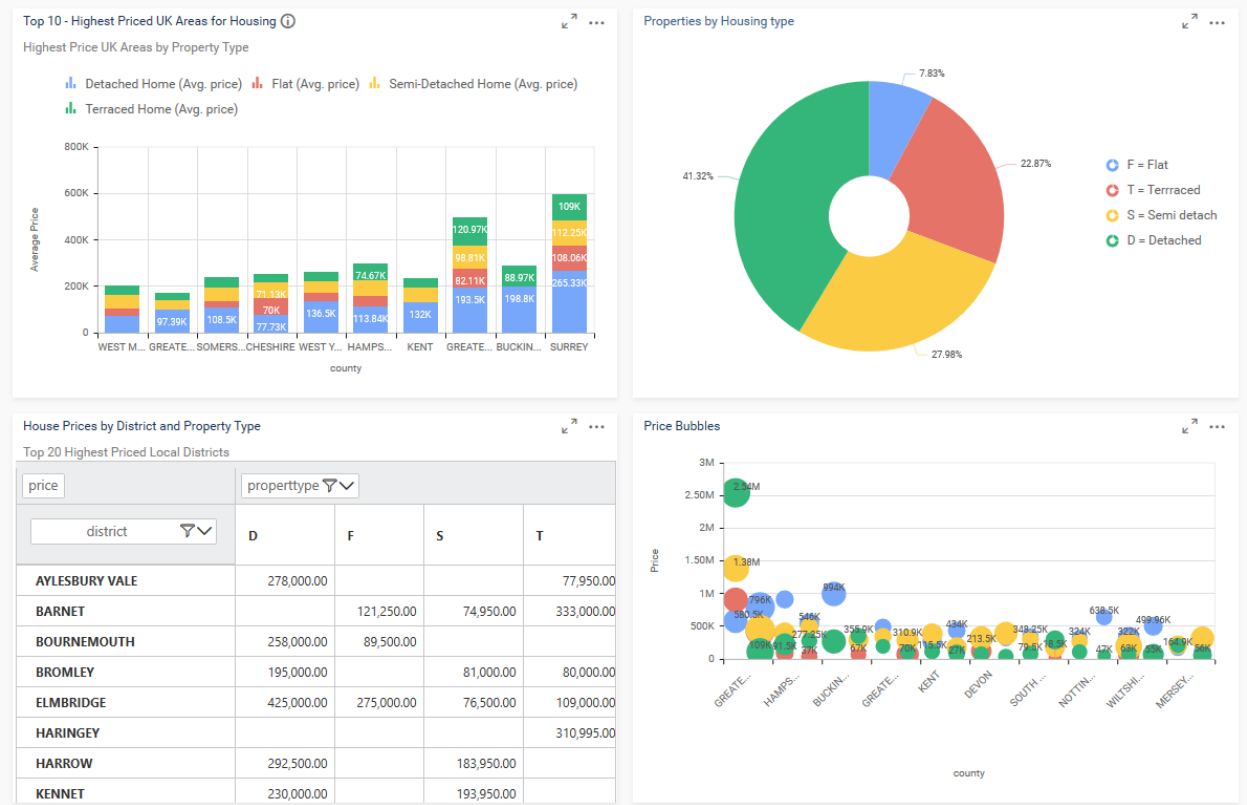


Figure 193: Syncfusion Dashboard Designer Dashboard including pivot table visualization

www.dbooks.org

Conclusion

At the start of this book, we were standing in the shadows of Hadoop on Linux. We were aware that Hadoop for Windows was seen as a novelty in some circles, a fledging aspiration perhaps. Now at the end of the book, would you say you felt the same? Why didn't we notice the resources Microsoft spent integrating Hadoop into their Windows real estate? A cynic might say Microsoft is cleverly "blurring the lines" by constantly releasing free software to Windows, Mac, and Linux users and normalizing cross-platform software. It counters the notion that Hadoop belongs to Linux, or SQL Server belongs to Windows. The 7,000,000 Linux users who downloaded SQL Server for Linux further reinforces that.

A year ago, my first thought for a new tablet would have been an iPad. But I've recently taken delivery of a Microsoft 2-in-1 cellular tablet, on which I was able to install Azure Data Studio and connect to Hadoop. The introduction of the portable and mobile experience to Hadoop is built into Microsoft strategy. Microsoft is constantly releasing new builds of these apps, so they are certainly committed to them. Why should I have to use a heavy server or PC to access Hadoop—who says things have to stay the same? Why can't I access it using an app with a tiny 80-MB install file? These innovations provide attractive new options for accessing Hadoop, and they are available on Windows. Arcadia Data has released its software on the Windows platform, and the INSTANT version is only 85 MB and allows fast connections to Hadoop.

Improvements and feature sets

Little can be done about Hadoop features available in Linux but unavailable in Windows. Hadoop is not perfect and perhaps the time has come to say big data for Windows as opposed to Hadoop for Windows. I say this as Microsoft are remixing Hadoop, creating components to interact with it that are superior to the Hadoop Ecosystem. That an 80-MB app can run Hadoop queries with joins many times faster than Hive doesn't reflect well on Hadoop. With these new Windows tools I don't need to use Sqoop or Hive, they can't do the things I want or they can't do them fast enough.

Hadoop user Communities for Windows and Linux

I am as comfortable using the Syncfusion Big Data Platform on Windows as I am using Cloudera CDH on Linux. While I can sell the idea of using Azure Data Studio to people for the reasons I've given, it's harder to sell the idea of using more traditional Hadoop for Windows. It's an additional advantage in that Azure Data Studio is available on Linux, Mac, and Windows, so there's no “us and them” anymore. I'm further aided by the 7,000,000 downloads of SQL Server for Linux by Linux users. So when I say Azure Data Studio uses SQL Server 2019, I already have their attention. Someone at Microsoft is very clever at strategy, and it's certainly working. Microsoft has doubled down on this with its free giveaway of Visual Studio Code on Linux, Mac, and Windows.

SQL Server does not "belong" to Windows anymore, and Hadoop does not "belong" to Linux. The door has already been opened—you only need to walk through it!