

# IBM Data Engine for Hadoop and Spark

Dino Quintero

Luis Bolinches

Aditya Gandakusuma Sutandyo

Nicolas Joly

Reinaldo Tetsuo Katahira



 Analytics

Power Systems





International Technical Support Organization

**IBM Data Engine for Hadoop and Spark**

August 2016

**Note:** Before using this information and the product it supports, read the information in “Notices” on page v.

### **First Edition (August 2016)**

This edition applies to the following software:

- ▶ IBM Platform Symphony Advanced Edition V7.1
- ▶ IBM Open Platform for Apache Hadoop V4.1.0.0
- ▶ Apache Ambari V2.1.0.0
- ▶ IBM Spectrum Scale Advanced Edition V4.1.1.2

© Copyright International Business Machines Corporation 2016. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Notices</b> .....	v
Trademarks .....	vi
<b>IBM Redbooks promotions</b> .....	vii
<b>Preface</b> .....	ix
Authors .....	ix
Now you can become a published author, too! .....	x
Comments welcome .....	xi
Stay connected to IBM Redbooks .....	xi
<b>Chapter 1. Introduction to IBM Data Engine for Hadoop and Spark</b> .....	1
1.1 What is big data .....	2
1.1.1 Structured and unstructured data .....	3
1.1.2 The four Vs of big data .....	4
1.1.3 The traditional data warehouse in relation to big data .....	5
1.2 Big data analytics .....	5
1.3 What is Apache Spark .....	6
1.3.1 Apache Hadoop and MapReduce versus Apache Spark .....	8
1.4 Why use an IBM Big Data and analytics solution .....	9
1.4.1 IBM Spectrum Scale file system as alternative to Hadoop File System .....	9
1.4.2 IBM Spectrum Conductor for Spark .....	10
1.4.3 IBM Open Platform with Apache Hadoop .....	10
1.4.4 IBM Spectrum Symphony .....	11
1.4.5 IBM Platform Cluster Manager .....	11
1.5 Why big data on IBM Power Systems servers .....	12
1.6 IBM Data Engine for Hadoop and Spark .....	13
<b>Chapter 2. Solution reference architecture</b> .....	15
2.1 Overview of the solution .....	16
2.2 High-level architecture .....	16
2.3 Hardware components of the solution .....	18
2.3.1 The IBM Power System S812LC server .....	18
2.3.2 Networking .....	20
2.4 Software reference architecture .....	22
2.4.1 IBM Open Platform with Apache Hadoop clusters .....	22
2.4.2 Stand-alone products: IBM Spectrum Scale and IBM Spectrum Symphony .....	23
2.4.3 Cluster management .....	26
2.4.4 Additional analytics software: IBM Spectrum Conductor with Spark .....	26
2.4.5 Software options .....	27
2.5 Solution reference architecture .....	28
2.5.1 Configuration .....	28
2.5.2 Predefined configurations .....	32
2.5.3 Sizing the solution .....	35
2.5.4 Rack, power, and cooling information .....	36
<b>Chapter 3. Use case scenario for the IBM Data Engine for Hadoop and Spark</b> .....	37
3.1 When to use IBM Data Engine for Hadoop and Spark .....	38
3.2 When to use Hadoop and what workloads are suitable for it .....	38

3.2.1	Landing Zone . . . . .	38
3.2.2	Data warehouse offloading . . . . .	39
3.3	When to use Apache Spark and what workloads are suitable for it . . . . .	40
3.4	Greater resource utilization by using IBM Spectrum Symphony . . . . .	41
3.5	Comparing Hadoop Distributed File System and IBM Spectrum Scale . . . . .	41
3.6	Using the analytic capabilities of IBM Open Platform . . . . .	43
<b>Chapter 4. Operational guidelines . . . . .</b>		<b>45</b>
4.1	Introduction . . . . .	46
4.2	Adding a compute node . . . . .	46
4.2.1	Identifying the networks . . . . .	46
4.2.2	Defining the Central Electronics Complex group . . . . .	47
4.2.3	Updating the server firmware . . . . .	50
4.2.4	Installing the base operating system . . . . .	54
4.2.5	Configuring the host name, users, and groups . . . . .	59
4.2.6	Installing and configuring IBM Spectrum Scale . . . . .	60
4.2.7	Installing software with Ambari . . . . .	63
4.3	Configuring the Apache Spark UI . . . . .	71
4.4	Deployment and operation tools . . . . .	75
4.4.1	List of tools . . . . .	76
<b>Chapter 5. Multitenancy . . . . .</b>		<b>77</b>
5.1	Introduction to multitenancy . . . . .	78
5.2	IBM Spectrum Computing resource manager . . . . .	79
5.3	Configuring multitenancy for MapReduce workloads . . . . .	80
5.3.1	Monitoring MapReduce jobs by using IBM Spectrum Symphony . . . . .	80
5.3.2	Creating an application profile . . . . .	84
5.3.3	Adding users or groups to an existing application profile . . . . .	91
5.3.4	Configuring the share ratio between application profiles . . . . .	93
5.3.5	Configuring slot mapping . . . . .	95
5.3.6	Configuring the priority for running jobs . . . . .	98
<b>Appendix A. Ordering the solution . . . . .</b>		<b>99</b>
	Predefined configuration . . . . .	100
	How to use the IBM Configurator for e-business (e-config) . . . . .	100
	Services . . . . .	101
<b>Appendix B. Script to clone partitions . . . . .</b>		<b>103</b>
	Clone partitions script . . . . .	104
<b>Related publications . . . . .</b>		<b>107</b>
	IBM Redbooks . . . . .	107
	Online resources . . . . .	107
	Help from IBM . . . . .	108

# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.


## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	IBM®	Power Systems™
BigInsights®	IBM Spectrum™	POWER8®
Bluemix®	IBM Spectrum Conductor™	PowerVM®
Cognos®	IBM Spectrum Scale™	Redbooks®
DB2®	IBM Spectrum Symphony™	Redbooks (logo)  ®
Decade of Smart™	InfoSphere®	Smarter Planet®
Global Business Services®	Optim™	Symphony®
GPFS™	POWER®	

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.



## Find and read thousands of IBM Redbooks publications

- ▶ Search, bookmark, save and organize favorites
- ▶ Get personalized notifications of new content
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



## Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



[ibm.com/Redbooks](http://ibm.com/Redbooks)  
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

# Preface

This IBM® Redbooks® publication provides topics to help the technical community take advantage of the resilience, scalability, and performance of the IBM Power Systems™ platform to implement or integrate an IBM Data Engine for Hadoop and Spark solution for analytics solutions to access, manage, and analyze data sets to improve business outcomes.

This book documents topics to demonstrate and take advantage of the analytics strengths of the IBM POWER8® platform, the IBM analytics software portfolio, and selected third-party tools to help solve customer's data analytic workload requirements. This book describes how to plan, prepare, install, integrate, manage, and show how to use the IBM Data Engine for Hadoop and Spark solution to run analytic workloads on IBM POWER8. In addition, this publication delivers documentation to complement available IBM analytics solutions to help your data analytic needs.

This publication strengthens the position of IBM analytics and big data solutions with a well-defined and documented deployment model within an IBM POWER8 virtualized environment so that customers have a planned foundation for security, scaling, capacity, resilience, and optimization for analytics workloads.

This book is targeted at technical professionals (analytics consultants, technical support staff, IT Architects, and IT Specialists) that are responsible for delivering analytics solutions and support on IBM Power Systems.

## Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Dino Quintero** is a Complex Solutions Project Leader and an IBM Level 3 Certified Senior IT Specialist with the ITSO in Poughkeepsie, New York. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

**Luis Bolinches** has been working with IBM Power Systems servers for over 15 years and has been with IBM Spectrum™ Scale (formerly known as IBM General Parallel File System (IBM GPFS™)) for over 7 years. He is at IBM Lab Services, and has been an IBM employee in Spain, Estonia, and now in Finland.

**Aditya Gandakusuma Sutandyo** is a technical sales for IBM Analytics in IBM Indonesia. He has been working at IBM for 4 years and handling the Analytics Platform portfolio with a focus on IBM DB2®, IBM InfoSphere® Optim™, IBM Cognos® Business Intelligence, Streams, and IBM BigInsights®. His areas of expertise include data management and big data analytics. He has been working with many clients across many industries to build solutions that are related to database management system, data warehouse, business intelligence, data lifecycle management, and big data.

**Nicolas Joly** is a pre-sales architect with IBM Systems in New York City, New York. His areas of knowledge include software-defined infrastructure, analytics solutions, storage, technical computing, and clustering solutions. He is working with major customers in the finance and telecommunication industry. Before joining IBM US, Nicolas was working for IBM France, where he was working as a technical sales specialist for analytics and technical computing solutions. Nicolas holds a master degree in Computer Science with a major in parallel and distributed computing from Institut Polytechnique de Bordeaux (ENSEIRB-MATMECA), France.

**Reinaldo Tetsuo Katahira** is an Information Architect, working for the Client Innovation Center under IBM Global Business Services® (GBS) in Brazil. He is a Thought Leader Certified IT Specialist in Data Management by IBM, and he has DB2, Oracle, and MS SQL Server certifications. Reinaldo holds a degree in Computer Engineering from the Escola Politecnica da Universidade de Sao Paulo. He has 19 years of experience in database management systems (DBMSs) and over 10 years of experience working at IBM, where he filed patent applications that relate to optimization of relational DBMS.

Thanks to the following people for their contributions to this project:

Michael Schwartz  
**International Technical Support Organization, Poughkeepsie Center**

Linda Cham and Nathan Falk  
**IBM Poughkeepsie, New York**

Haohai Ma  
**IBM Canada**

Additional contributors with technical presentations and documentation:

Anand Haridass, Jim Woodbury, Kyle Wurgler, Steve Roberts, David C. Maddison  
**IBM USA**

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:  
[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an email to:

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ Stay current on recent Redbooks publications with RSS Feeds:

<http://www.redbooks.ibm.com/rss.html>





# Introduction to IBM Data Engine for Hadoop and Spark

This chapter provides an introduction to the IBM Data Engine for Hadoop and Spark solution.

IBM Data Engine for Hadoop and Spark is a recent IBM offering that combines the recently added line of OpenPOWER Linux servers that is designed for big data and analytics with an open source Apache Hadoop and Spark distribution along with optional advanced analytics capabilities.

The following topics are described in this chapter:

- ▶ What is big data
- ▶ Big data analytics
- ▶ What is Apache Spark
- ▶ Why use an IBM Big Data and analytics solution
- ▶ Why big data on IBM Power Systems servers
- ▶ IBM Data Engine for Hadoop and Spark

## 1.1 What is big data

As the planet becomes more integrated, the rate of data growth is increasing exponentially. This data explosion is rendering commonly accepted practices of data management inadequate. As a result, this growth gives birth to a new wave of business challenges regarding data management and analytics. Many people are using the term *big data* (sometimes referred to as Big Data) to describe this latest industry trend. To help you understand it better, this chapter provides a foundational understanding of big data, what it is, and why you must care about it. In addition, it describes how IBM is poised to lead the next generation of technology to meet and conquer the data management challenges that it presents. The increasing volume and detail of information, the rise of multimedia and social media, and the *Internet of Things* are expected to fuel continued exponential data growth for the foreseeable future. The Internet of Things is generating large volumes and various data from sources as varied as ebooks, vehicles, video games, television set-top boxes, and household appliances. Capturing, correlating, and analyzing this data can produce valuable insights for a company.

For those individuals whose professions are heavily based in the realm of information management, there is a good chance that they are fully involved in big data projects. It is becoming increasingly popular to incorporate big data in data management discussions. In a similar way, it was previously popular to bring the advent of service-oriented architecture (SOA) and Web 2.0, just to give a few examples. The term big data is a trendy talking point at many companies, but few people understand what exactly is meant by it. Instead of volunteering an arbitrary definition of the term, a better approach is to explore the evolution of data along with enterprise data management systems. This approach ultimately arrives at a clear understanding of what big data is and why you must care.

Beginning in 2008 during a speech to the Council of Foreign Relations in New York, IBM began its IBM Smarter Planet® initiative. Smarter Planet is focused on the development of leading-edge technologies that are aimed at advancing everyday experiences. A large part of developing such technology depends on the collection and analysis of data from as many sources as possible. This process is becoming increasingly difficult as the number and variety of sources continues to grow. The planet is exponentially more instrumented, intelligent, and integrated and it continues to expand with better and faster capabilities. In January 2010, IBM inaugurated the *IBM Decade of Smart™* at the Chatham House in London, where important initiatives regarding smarter systems to solve the planet's most alarming issues were presented to achieve economic growth, near-term efficiency, sustainable development, and societal progress.

The World Wide Web is also truly living up to its name, and through its continued expansion, the web is driving our ability to generate and have access to virtually unlimited amounts of data. There was a point earlier in history where only home computers and web-hosting servers were connected to the web. If you had a connection to the web and ventured into the world of chatrooms, you communicated by instant messaging with someone in another part of the world. Hard disk drives (HDDs) were 256 MB, CD players were top-shelf technology, and cell phones were as large as lunch boxes. Today, the chances are that you can download this book from your notebook or tablet at the same time you are sending an email, sending instant messages back and forth with a friend overseas, or texting your significant other, all at the same time you are enjoying your favorite clothing retailer's Facebook page. The point is, you now generate more data in 30 seconds than you can in 24 hours ten years ago.

We are now at the crux of a data explosion with more items continuously generating data. Where exactly is this data coming from?



Web-based applications, including social media sites, now exceed standard e-commerce websites in terms of user traffic. Facebook roughly produces 25+ TBs of log data daily. Twitter creates 12+ TBs of tweet data (made up mostly of text, despite the 140-character tweet limit), even more if there is a major global event (#IBMBigDataRedbook). Most everyone has an email address (often multiple), a smartphone (sometimes multiple as well), usually a cache of photo images and video (whether they choose to share with the social network or not), and can voice their opinion globally with their own blog. In this increasingly instrumented world, there are sensors everywhere constantly generating and transmitting data. In the IT realm, machine data is being generated by servers and switches, and they are always generating log data (commonly known as data exhaust). Also, these software applications are all 24x7 operational and continuously generating data.

The number of insights and trends that can be extracted from the stored but unexplored data, which is the new natural resource, is unlimited. Executives have relied on experience and intuition to formulate critical business decisions in past decades, but in the new era of big data, key decisions can be supported by decision support systems.

Despite establishing that there is more data generated today than there was in the past, big data is not just about the sheer volume of data that is being created. With a myriad of unstructured sources creating this data, a greater variety of data is now available. Each source produces this data at different rates or *velocity*. In addition, you still must decipher the veracity of this new information as you do with structured data.

### 1.1.1 Structured and unstructured data

*Structured data* implies that data elements are stored according to a predefined data model. Sets of entities, tables, and files that are organized into attributes, fields, columns, and lines with predefined data types are examples of data models. A data type is the predefined type, length, and format of stored data. For example, a time stamp format might be represented as YYYY-MM-DD HH:mm:SS, and an instance of that data type is 2015-11-05 11:00:00. Every instance of one entity, table, or file is considered a new record, row, or line.

A spreadsheet, which is an example of structured data, is a set of tabs where every cell is an intersection of a column and a row. A variant data type is also considered a data type based on its definition, which is the most common data type of spreadsheet cells.

A relational database, which is another example of structured data, consists of a set of tables that are organized into rows according to the columns' predefined data types. In addition, relational databases enforce relationship constraints between tables to establish the consistency of data across the database.

*Unstructured data* has no predefined data model. However, it can be scanned and analyzed to provide the required data. Text data, for example, a digital copy of a contract, is considered unstructured data because the established date of the contract is not necessarily described in a predefined field or format and it can be scanned and found throughout that file. Other examples of unstructured data are video, image, and audio files that you can analyze to identify patterns and anomalies, and extract valuable insights.

You might ask yourself about something in between structured and unstructured data. *Semi-structured data* is unstructured data that is combined with metadata that provides tags and instructions for the position, format, length, or type of a specific data element to be addressed within the unstructured data source. XML files and JSON messages are examples of semi-structured data with a self-describing structure.

Big data can be subcategorized as *data in motion* and *data at rest*. The process of analyzing data dynamically without storing is referring to data in motion, and data that is collected and stored as inactive with no or few updates is data at rest.

## 1.1.2 The four Vs of big data

From traffic patterns and music downloads to web history and medical records, data is recorded and stored. Depending on the industry and organization, big data encompasses information from multiple internal and external sources, such as transactions, social media, enterprise content, sensors, and mobile, such as smartphones and wearable devices that are connected to the Internet of Things. Companies can take advantage of data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

IBM data scientists break big data into four dimensions:<sup>1</sup>

- ▶ The *volume* of data that is generated is exponentially growing and the insights that possibly can be extracted from this source is unprecedented. It is estimated that 2.5 quintillion bytes of data are created each day, and most companies in the US have at least 100 TB of stored data. Six billion people have cell phones out of seven billion people in the world. 40 ZB of data will be created by 2020, an increase of 300 times from 2005. Big data solutions must present scalability where you can add computing and storage capacity as required to meet your business necessities.
- ▶ The *velocity* of consumed data gives you an opportunity to analyze streaming data. Modern cars have close to 100 sensors that monitor items such as fuel level and tire pressure. By 2016, it is estimated that there will be 18.9 billion network connections, which is almost 2.5 connections per person on earth. The New York Stock Exchange captures 1 TB of trade information during each trading session. The analysis of streaming data can instantly provide fraud detection for transactions and save lives by preventing car accidents.
- ▶ The *variety* of forms of data, where source and format can differ completely. By 2014, 420 million wearable, wireless health monitors are estimated to be in use. More than 4 billion hours of video are watched on YouTube each month. 400 million tweets are sent per day by about 200 million monthly active users. 30 billion pieces of content are shared on Facebook every month. As of 2011, the global size of data in healthcare was estimated to be 150 EB. Structured, unstructured, and semi-structured data from multiple sources and formats must be fully addressed by big data solutions.
- ▶ The veracity of data, even with the uncertainty of data, with which business decisions are still made. Poor data quality costs the US economy around 3.1 trillion US dollars a year. 27% of respondents in one survey were unsure of how much of their data was inaccurate. 1 in 3 business leaders do not trust the information that they use to make decisions.

Some essays have mentioned *value* and *visualization* as other Vs of big data, and other studies have also identified a stunning total of 12 Vs besides all the previous ones: *variability*, *validity*, *volatility*, *verbosity*, *vulnerability*, and *verification*. However, IBM data scientists collectively and officially consider the four Vs of big data as *volume*, *velocity*, *variety*, and *veracity*.

---

<sup>1</sup> Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTec, and QAS.

### 1.1.3 The traditional data warehouse in relation to big data

Some people might have the opinion that big data presents nothing new. They might say that it is already addressed by the data warehouse (DW). Some might suggest that their DW works fine for the collection and analysis of structured data, and that their Enterprise Content Management (ECM) solution works well for their unstructured data needs. DW design is a mature practice in the data management arena and affords those who implemented a DW and IBM Open Platform with significant value by enabling deeper analytics of the stored data. Traditional DWs are now a foundational piece of a larger solution.

Typically, DWs are built on some enterprise-level relational database management systems (RDBMSs). Regardless of the vendor, at their core these platforms are designed to store and query structured data. This approach was solid until the need to do the same thing with unstructured data rose. As the need for this function became more prevalent, many vendors included unstructured data storage and query capabilities in their RDBMS offerings. The most recent example is the ability to handle XML data. IBM introduced the basic XML data type in its 2006 release of DB2 9.1. Furthering this capability to enforce structure on unstructured data, text search and analysis tools were developed to enable the extraction and reformatting of data. This data then could be loaded into the structured DW for query and analysis.

There are in-memory solutions that are aimed at faster analysis and processing of large data sets. However, these solutions still have the limitation that data must be primarily structured. Thus, in-memory solutions are subject to the same pitfalls as traditional DWs as it pertains to management of big data.

## 1.2 Big data analytics

It is estimated that a staggering 70% of the time that is spent on analytics projects is concerned with identifying, cleansing, and integrating data because of the following issues:

- ▶ Data is often difficult to locate because it is scattered among many business applications and business systems.
- ▶ Frequently, the data needs reengineering and reformatting to make it easier to analyze.
- ▶ The data must be refreshed regularly to keep it up-to-date when it is in use by analytics.

Acquiring data for analytics in an *ad hoc* manner creates a huge burden on the teams that own the systems that supply the data. Often, the same type of data is repeatedly requested and the original information owner finds it hard to track who has copies of which data.

As a result, many organizations are considering implementing a *data lake* solution. A data lake is a set of one or more data repositories that are created to support data discovery, analytics, *ad hoc* investigations, and reporting. The data lake contains data from many different sources. People in the organization are welcome to add data to the data lake and access any updates as necessary.

However, without proper management and governance, a data lake can quickly become a *data swamp*. A data swamp is overwhelming and unsafe to use because no-one is sure where data came from, how reliable it is, and how it can be protected.

IBM proposes an enhanced data lake solution that is built with management, affordability, and governance at its core. This solution is known as a *data reservoir*. A data reservoir provides the correct information to people so they can perform the following activities:

- ▶ Investigate and understand a particular situation or type of activity.
- ▶ Build analytical models of the activity.
- ▶ Assess the success of an analytic solution in production to improve it.

A data reservoir has capabilities that ensure that the data is properly cataloged and protected so subject matter experts (SMEs) have access to the data that they need for their work. This design point is critical because SMEs play a crucial role in ensuring that analytics provides worthwhile and valuable insights at appropriate points in the organization's operation. With a data reservoir, line-of-business (LOB) teams can take advantage of the data in the data reservoir to make decisions with confidence.

Two new key technologies enable a computing infrastructure: Hadoop MapReduce and Streams (stream computing). When these new infrastructures are combined with traditional enterprise data marts, analytics can use the full range of data. Persistent context glues the environments together. Hadoop enables redesigned analytics to ingest and use quickly enormous data sets, and to combine data that previously was impossible to bring together because of the rigidity of traditional database schemas. The ability of Hadoop to use all of the data reduces the chance of missing low-level anomalies within predictive models. Models that are embedded in streams can assess the relevance of each new data element on arrival. Analytic accuracy, including the reduction in false positives and false negatives, is enhanced by using a context-based historical data set.

Persistent context can be used to identify emergent patterns within the data, such as patterns of life, and anomalies in the data. The combination of streams and persistent context allows for the real-time assessment of each new data for cumulative relevance or contributions to models such as threat scores.

The following key technologies, among others, are vital to ensure effective intelligence:

- ▶ Feature extraction
- ▶ Context and situational awareness
- ▶ Predictive modeling
- ▶ Data analysis upon arrival

In most cases, these technologies have existed for some time, but are now enjoying much greater scale and performance because of new computing capabilities and architectures. When looked at individually, each is powerful. If you can envision using them collectively, the opportunity is vast.

## 1.3 What is Apache Spark

Apache Spark is an open source cluster computing framework for large-scale data processing. Like MapReduce, Apache Spark provides parallel distributed processing, fault tolerance on commodity hardware, and scalability. With its in-memory computing capabilities, analytic applications can run up to 100 times faster on Apache Spark compared to other technologies on the market today.

Apache Spark is highly versatile and known for its ease of use in creating algorithms that harness insight from complex data. In addition to its ease of use, this framework covers a wide range of workloads through its different modules:

- ▶ Interactive queries through Apache Spark SQL
- ▶ Streaming data, with Apache Spark Streaming
- ▶ Machine Learning, with the MLib module
- ▶ Graph processing with GraphX

Applications can be built by using simple APIs for Scala, Python, and Java:

- ▶ Batch applications leveraging the MapReduce compute model
- ▶ Iterative algorithms that build upon each other
- ▶ Interactive queries and data manipulation through “Notebooks” (a web-based interface)

Apache Spark runs on Hadoop clusters such as Hadoop YARN or Apache Mesos, or even stand-alone with its own scheduler.

Figure 1-1 presents the Apache Spark architecture.

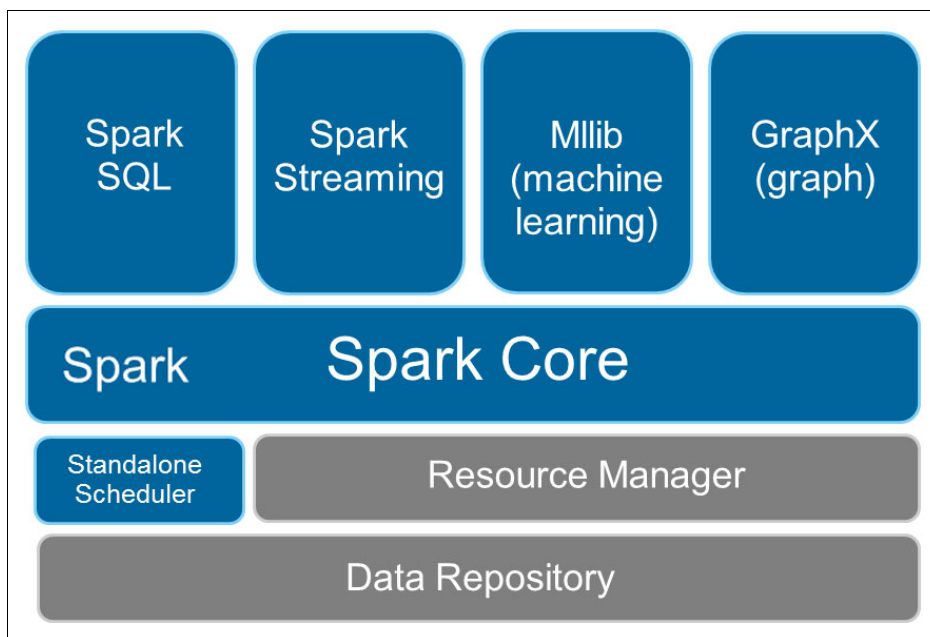


Figure 1-1 Apache Spark software stack

Developed in the AMPLab at the University of California at Berkeley, Apache Spark was elevated to a top-level Apache Project in 2014 and continues to expand today.

With all the challenges for enterprises and the advantages of Apache Spark, it is understandable why the Apache Spark community of both users and contributors has grown so significantly; even commercial vendors are now offering Apache Spark-based solutions for key analytics insight, such as fraud detection and customer insight.

In June 2015, IBM announced a major commitment to Apache Spark, including plans to put more than 3,500 IBM developers and researcher to work on Apache Spark-related projects worldwide, contributing the IBM SystemML machine learning technology to the Apache Spark open-source community, and intending to offer Apache Spark as a service on IBM Bluemix®. IBM also established the Apache Spark Technology Center with a focus on contributing features and function to the open source Apache Spark community. Over time, select technologies that are offered by IBM can use Apache Spark as part of the underlying platform to take advantage of Apache Spark's flexibility, in-memory analytics, and built-in set of machine learning libraries.

Despite its relative ease of use, Apache Spark deployments present multiple challenges in the enterprise:

- ▶ Lifecycle management: The Apache Spark framework, as with many other open source projects, has a quick release cycle.
- ▶ Expertise: You must configure and install various tools for monitoring, management, and workflow.

### 1.3.1 Apache Hadoop and MapReduce versus Apache Spark

Although Apache Hadoop and MapReduce provide a parallel computing cluster for the analysis of large and complex data sets, ease of use and timely execution of queries have sometimes been barriers to adoption, falling short in terms of user expectations and experience. In some cases, Hadoop implementations remain as a data lake or reservoir to store data for future analysis. Apache Spark aims to address complexity, speed, ease of use, and why it is key to the future success of analytics, and the impact it is having and continues to have on organizations, the business, their analytics strategies, and data scientists and developers.

Analytics is increasingly a part of day-to-day operations at today's leading businesses, and transformation is also occurring through huge growth in mobile and digital channels. Previously acceptable response times and delays for analytic insight are no longer viable, with more push toward real-time and in-transaction analytics. In addition, data science skills are increasingly in demand. As a result, enterprise organizations are attempting to take advantage of analytics in new ways and transition existing analytic capability to respond with more flexibility, at the same time making the most efficient use of highly valuable data science skills.

Although the demand for more agile analytics across the enterprise is increasing, many of today's solutions are aligned to specific platforms and tied to inflexible programming models, which require vast data movements into data lakes. These lakes quickly become stale and unmanageable, resulting in pockets of analytics and insight that require ongoing manual intervention to integrate into coherent analytics solutions.

With all these impending forces converging, organizations are poised for a change. The recent growth and adoption of Apache Spark as an analytics framework and platform is timely and helps meet these challenging demands.

## 1.4 Why use an IBM Big Data and analytics solution

The IBM Big Data and analytics platform gives organizations a solution stack that is designed specifically for enterprise use. The IBM Big Data and analytics platform provides the ability to start small with one capability and easily add others over your big data journey because the pre-integration of its components reduces your implementation time and cost.

### 1.4.1 IBM Spectrum Scale file system as alternative to Hadoop File System

The Hadoop File System (HDFS) provides a cost-effective way to store large volumes of structured and unstructured data in one place for deep analysis. IBM provides a non-forked, open source Hadoop version and augments it with capabilities, such as enterprise-class storage by using an IBM Spectrum Scale™ File System, security by reducing the surface area and securing access to administrative interfaces and key Hadoop services, and workload optimization by using the Adaptive MapReduce algorithm that optimizes execution time of multiple small and large jobs.

IBM Spectrum Scale also can act as a Portable Operating System Interface (POSIX) compliant file system so that other computers that are not part of the Hadoop cluster can access these files normally. If only the HDFS implementation is used, then all the operations must be done on the **hadoop** command line. If ingress or outgress of data must be presented to non-Hadoop nodes, data must be copied back and forth from the non-HDFS to and from the POSIX file system. This impacts the time that is needed and operations and costs storage. However, IBM Spectrum Scale offers both HDFS and POSIX interfaces to the same data in parallel.

Like HDFS for Hadoop, IBM Spectrum Scale is designed to support distributed computing with the ability to take advantage of direct attached storage and HDFS-style data locality. However, unlike HDFS, IBM Spectrum Scale is a general-purpose distributed file system that supports much broader types of workloads beyond Hadoop. IBM Spectrum Scale has a Hadoop connector that enables Hadoop applications (that use HDFS APIs) to access transparently the IBM Spectrum Scale file system.

In addition to the data locality and replication that HDFS and IBM Spectrum Scale both give, IBM Spectrum Scale allows the creation of multiple online snapshots, move the data transparently to cheaper tiers or different types of storage, including external cloud providers, based on how often the data is accessed and many other Information Lifecycle Management (ILM) operations that simply are not in HDFS.

When it comes to disaster recovery, the approach matters, and efficiency matters even more for big data. The IBM Spectrum Scale approach leverages kernel level metadata and facilities, and provides file system block level granularity.

**Note:** For more information about IBM Spectrum Scale, see the following website:

<http://www.redbooks.ibm.com/abstracts/sg248254.html?Open>

## 1.4.2 IBM Spectrum Conductor for Spark

IBM Spectrum Conductor™ for Spark is a complete enterprise-grade multitenant solution for Apache Spark. It is designed to address the requirements of users needing to adopt Apache Spark technology and to integrate it into their environment.

To address Apache Spark challenges, IBM Spectrum Conductor for Spark delivers the following benefits:

- ▶ Accelerate results: Run Apache Spark natively on a shared infrastructure without the dependency of Hadoop. This situation reduces application wait time and increases time to results.
- ▶ Reduce administration costs: Proven architecture at extreme scale, with enterprise class workload management, monitoring, reporting, and security capabilities.
- ▶ Increase resource utilization: Fine grain, dynamic allocation of resources maximizes the efficiency of Apache Spark instances sharing a common resource pool. Extends beyond Apache Spark and eliminates cluster sprawl.
- ▶ End to end enterprise class solution: A tightly integrated offering that combines the IBM supported Apache Spark distribution with workload, resource and data management, and IBM support and services.

IBM Spectrum Conductor for Spark provides a built-in Apache Spark version, one that is prepackaged to include Apache Spark and Apache Zeppelin. As other versions become available, you can download these versions as part of a Apache Spark package from IBM Fix Central. You can download these version packages and import them for use with Platform Conductor for Spark.

Beside the built-in Zeppelin notebook, you can use other third-party notebooks, such as iPython, by integrating them with Platform Conductor for Spark.

**Cluster sprawl:** This is the multiplicity of *ad hoc* Apache Spark and MapReduce clusters.

## 1.4.3 IBM Open Platform with Apache Hadoop

*IBM Open Platform with Apache Hadoop* is the IBM software platform for storing and analyzing data at rest, providing valuable insights for business decision-makers and automating core business processes. The principle of Hadoop entitles inexpensive computing systems to perform analysis over a huge volume of data and new nodes can be added for scalability purposes.

IBM Open Platform with Apache Hadoop V4.1 consists of Apache Hadoop open source components for use in big data analysis, such as *Ambari, HDFS, YARN, MapReduce, Flume, HBase, Hive, Kafta, Knox, Oozie, Pig, Slider, Solr, Apache Spark, Sqoop, and Zookeeper*. IBM Open Platform with Apache Hadoop extends the Hadoop open source framework with enterprise-grade security, governance, availability, integration into existing data stores, tools that simplify developer productivity, and more.

**Note:** For more information about IBM Open Platform with Apache Hadoop (second generation), see the following website:

<http://ibm.co/29tTRd8>



This software stack runs on Linux on Power. For IBM Open Platform with Apache Hadoop on Power Systems, the supported version at the time of writing is Red Hat Enterprise Linux V7.2 Little Endian.

### 1.4.4 IBM Spectrum Symphony

IBM Spectrum Symphony™ is a resource scheduler for grid environments. It works with grid-enabled applications and can provide high resource utilization rates along with low latency for certain types of jobs.

IBM Spectrum Symphony can be used in an IBM Open Platform environment as a job scheduler for MapReduce tasks. IBM Spectrum Symphony can replace the open source YARN scheduler in a MapReduce based framework and provide advantages such as the following ones:

- ▶ Better performance by providing lower latency for certain MapReduce based jobs.
- ▶ Dynamic resource management that is based on slot allocation according to job priority and server thresholds.
- ▶ A fair-share scheduling scheme with 10,000 priority levels for jobs of an application.
- ▶ A complete set of management tools for providing reports, job tracking, and alerting.
- ▶ Reliability by providing a redundant architecture for MapReduce jobs in terms of name nodes (in case the HDFS is in use), job trackers, and task trackers.
- ▶ Support for rolling upgrades, hence maximizing uptime of your applications.
- ▶ Open, so it is compatible with multiple APIs and languages, such as Hive, Pig, Java, and others. Also, it is compatible with both HDFS and IBM Spectrum Scale.

Using IBM Spectrum Symphony as a scheduler for an IBM Open Platform environment is a choice that you can make while you are installing IBM Open Platform because IBM Spectrum Symphony might be configured with IBM Open Platform in an integrated fashion.

### 1.4.5 IBM Platform Cluster Manager

Managing many systems is a time-consuming, error-prone, and tedious task when done manually. In the past, before the era of virtualization, systems management did not take up much time from IT personnel because of the small number of systems that they managed for a solution. With the advent of virtualization, the increasing processing power of servers, and the requirements to process large amounts of data, it is inevitable that you face a situation in which you must manage a large server farm within your company.

IBM Platform Cluster Manager (PCM) is a cluster management software that can perform bare-metal or virtualized<sup>2</sup> systems deployment, and also can create cluster configurations on the deployed systems. Imagine that you want to deploy multiple, independent IBM Open Platform clusters. If you did that manually, not only do you have to do much work, but both installations might end up being slightly different because of the lack of automation. Also, imagine that you had multiple IBM Open Platform clusters, but some of them had peaks at the same time others were more idle in a certain period. Would you like to be able to reassign some compute nodes from one IBM Open Platform cluster to another based on your business needs and make better use of your computation power? With PCM, you can.

---

<sup>2</sup> Supported hypervisors only, such as PowerKVM and IBM PowerVM® on IBM Power Systems servers.

PCM can be used to manage IBM POWER® servers hardware, install a particular operating system (OS) image onto them, and use a few of these servers to create an IBM Open Platform cluster. This is what the authors have done in this book, and we share our experience with you. At the time of writing, there were a few IBM Power Systems servers available for use, and we used them in a bare-metal fashion with no input/output (I/O) virtualization because this is the most probable scenario that most customers do in the field if they opt to manage many servers with PCM.

Here is a list of other PCM advantages when you use it to manage your clusters:

- ▶ Elimination of cluster silos: You can grow and shrink your clusters on demand as you want, or through on cluster load monitoring.
- ▶ Multitenancy: Clusters are independent from one another, and isolated.
- ▶ Automation: You can create customization rules for OS deployment and cluster software installation.
- ▶ Self-service: After cluster creation rules are published, any authorized user can create, grow, or shrink a cluster based on the provided cluster template rules and resource utilization quotas.
- ▶ Support for multiple cluster software: In addition to IBM Open Platform with Apache Hadoop, you can use PCM to manage other clusters. You can use PCM to manage any cluster configuration by scripting the cluster software installation, adding and removing nodes from a cluster, and performing cluster software updates.
- ▶ There is support for bare-metal or hypervisor-based (virtualization) deployments.

## 1.5 Why big data on IBM Power Systems servers

The OpenPOWER Foundation has been established in 2013 and it is an open technical membership organization that is intended to enable data centers to innovate their approach to technology. To promote the optimization for their business requirements, member companies are allowed to customize POWER CPU processors and systems platforms. This acceleration is enabled by exploration of efficient utilization of resources such as graphics processing units (GPUs), flash memory, networking, and field programmable gate arrays (FPGAs), which helps improve performance, reduce latency, and result in more workload per dollar.

IBM, as platinum member of the OpenPOWER Foundation, offers a broad set of analytics capabilities that are built on the proven foundation of a single platform, IBM Power Systems. Power Systems is an open, secure, and flexible platform that is designed for big data. It has massive I/O bandwidth to deliver analytics in real time, and it can provide the necessary capabilities to handle the varying analytic initiatives of each business.

In 2014, IBM announced POWER8, the first microprocessor that was designed for big data and analytics. POWER8 offers numerous advantages for big data and analytics solutions: processing capability, memory capacity and bandwidth, cache workspace, and the ability to move information in and out of the system at the required rapid speeds.

A few of the distinguishing capabilities of POWER8 are listed:

- ▶ Parallelism is the capability to process more concurrent queries in parallel faster and scale easily to support a growing number of users who need reports, or to perform *ad hoc* analytics.
- ▶ Increased memory bandwidth to move large volumes of data to memory faster to accelerate time to result.

- ▶ Four-level cache design in every processor. A robust cache design helps with handling large volumes of data for better response times.
- ▶ Faster I/O to ingest, move, and access large volumes of data for various data sources so that analytics results are available faster.
- ▶ Acceleration that is enabled by Coherent Accelerator Processor Interface (CAPI) technology, through which GPUs, flash memory, networking, and FPGAs connect directly to the processor, which helps to improve performance, reduce latency, and result in more workload for the dollar.

## 1.6 IBM Data Engine for Hadoop and Spark

IBM Data Engine for Hadoop and Spark is designed to run *Open on Open*, which means IBM Open Platform is intended to operate fully on OpenPOWER servers, providing the following benefits:

- ▶ IBM Open Platform with Apache Hadoop and Spark:
  - The benefits of Open Source with improved quality and support from IBM.
  - Customized to the client requirements as big data practitioner requirements.
- ▶ OpenPOWER Servers:
  - New Management and data nodes that are engineered to address the requirements of Hadoop and Spark use cases.
  - Standard, Flexible CPU, Memory, Adapter, Networking, and Storage options.
  - Competitively priced.
  - POWER8 is designed and optimized for big data and analytics.

IBM Open Platform with Apache Hadoop offers analytics features that are ahead of the competition. It works with both IBM Spectrum Scale File System and HDFS. However, IBM Spectrum Scale provides the following advantages over HDFS:

- ▶ POSIX file system compatible, which is easy to use and manage.
- ▶ Scale compute and storage independently (Policy-based ILM).
- ▶ No single point of failure, with distributed metadata in active/active configuration since 1998.
- ▶ Ingest data that use policies for data placement.
- ▶ Versatile, multi-purpose, and hybrid storage (locality and shared).
- ▶ Enterprise ready with support for advanced storage features (encryption, disaster recovery, replication, and software RAID).
- ▶ Variable block sizes, which are suited to multiple types of data and a metadata access pattern.

Referring to IBM Spectrum Symphony Differentiation versus Open Source, IBM Spectrum Symphony has a 50% faster time to insights with MapReduce.

Preemption is important for resource sharing. If the system cannot pre-empt workloads, then service-level agreements (SLAs) cannot be reliably supported. Important workloads with tight SLAs must be able to pre-empt less important workloads.

In open source, only the FAIR scheduler supports preemption, and it is not available for The capacity scheduler now. IBM Spectrum Symphony preemption is enterprise ready and has different ways of pre-empting, which gives administrators even more control.

For example, preemption of the least running jobs assumes that jobs that have run the shortest amount of time have completed the “least” amount of work. Therefore, pre-empting such jobs results in the least amount of work to be reperformed.

Round-robin preemption is a form of “fair” preemption, where each user/group has jobs pre-empted fairly.

IBM Spectrum Conductor for Spark addresses customer challenges and offers significant business benefits:

- ▶ Faster time to results because of highly efficient resource scheduling technology
- ▶ Lower capital expenditure on hardware resources resulting from maximized utilization of a shared infrastructure
- ▶ Apache Spark multitenancy:
  - Run multiple Apache Spark instances simultaneously across a shared infrastructure, taking advantage of resources that can otherwise be idle
  - Run different Apache Spark versions simultaneously, mitigating issues of managing fast-changing versions
- ▶ A single user interface to manage set up and execution of Apache Spark workloads
- ▶ Integrated IBM Spectrum Scale-File Placement Optimizer (IBM Spectrum Scale-FPO) technology is a more efficient alternative to HDFS:
  - Smaller footprint (a smaller block size compared to HDFS is more suited to transactional data)
  - POSIX compliant, unlike HDFS
  - No single point of failure, unlike HDFS (depends on availability of NameNode process)
  - (We support HDFS and other file systems for clients who have an alternative preference.)

IBM Spectrum Conductor is a complete solution that includes:

- ▶ Apache Spark distribution (from IBM Spark Technology Center)
- ▶ Resource scheduler (proven in some of the world’s most demanding customer environments)
- ▶ Workload management, monitoring, alerting, reporting, and diagnostic tests
- ▶ Data management
- ▶ All managed from a single GUI

IBM Data Engine for Hadoop and Spark is the IBM solution package of hardware and software for organizations so that they can take advantage of IBM Open Platform with Apache Hadoop and Apache Spark workloads, including the possibility of scale-out nodes. This solution can provide the processing power that meets your data growth or company requirements.



## Solution reference architecture

This chapter introduces the elements that comprise the IBM Data Engine for Hadoop and Spark solution from both a software and a hardware perspective.

The following topics are described in this chapter:

- ▶ Overview of the solution
- ▶ High-level architecture
- ▶ Hardware components of the solution
- ▶ Software reference architecture
- ▶ Solution reference architecture

## 2.1 Overview of the solution

The IBM Data Engine for Hadoop and Spark is a fully integrated infrastructure solution with integrated cluster management and analytics software that is optimized for Hadoop-based and Apache Spark-based workloads. The solution is designed to deliver superior price and performance for these workloads while improving ease of deployment and cluster operational simplicity for clients deploying big data and analytics applications to support their businesses.

The solution is based on a set of standard building blocks that can be tailored to fit the data size, throughput, and scale that is required for the target analytics scenarios.

This architecture defines the following items:

- ▶ **Complete cluster:** A comprehensive, tightly integrated cluster that is designed for ease of procurement, deployment, and operation. It includes all the required components for big data applications, including servers, network, operating system (OS), management software, Hadoop and Apache Spark compatible software, and runtime libraries.
- ▶ **Scale out architecture:** Designed with a traditional Hadoop architecture, each data node in the system includes locally attached disks that are used to create the Hadoop Distributed File System (HDFS) or IBM Spectrum Scale file system for the cluster. Data is replicated three times between different nodes to protect against data loss. Compute capacity and file system capacity are scaled together by adding additional data nodes.
- ▶ **Open software with optional value-added components:** The Open Data Platform initiative (ODPi) is a shared industry effort promoting and advancing the state of Apache Hadoop and big data technologies. The ODPi Core is a set of common open source software components, including Apache Hadoop and Apache Ambari. IBM Open Platform with Apache Hadoop is the freely available distribution of the ODPi Core components that are used in this solution. The combination of IBM Open Platform packages provides an open and comprehensive solution for Hadoop and Apache Spark workloads.
- ▶ **Open hardware with POWER8 processors:** The IBM Power System S812LC server with POWER8 processor provides high performance in a cost-effective server design. This server provides exceptional computing power for analytics workloads with eight threads per core, 8 MB of L3 cache per core, and a total max peak memory bandwidth of 170 GBps. The Power S812LC server also features storage-rich configurations, allowing for input/output (I/O) intensive workloads to run efficiently.
- ▶ **Ease of deployment:** The full solution is assembled and installed at an IBM delivery center before delivery with all the included software preinstalled. On-site services personnel integrate the solution into the customer data center. The solution includes Platform Cluster Manager - Advanced Edition (PCM - AE) to simplify deployment and monitoring of the cluster.

## 2.2 High-level architecture

From an infrastructure design perspective, a cluster that supports big data-related workloads has two key aspects: a distributed file system and a compute engine. By default, this solution implements an HDFS and a MapReduce environment by using IBM Open Platform with Hadoop. Optionally, as a stand-alone module, you can implement IBM Spectrum Scale as the distributed file system and IBM Spectrum Symphony® as the MapReduce environment.

The IBM Data Engine for Hadoop and Spark solution is composed of a pool of management nodes handling the services and managing the distributed environment, and a pool of data nodes that handles the Hadoop and Apache Spark computation workloads. A system management node is used to deploy and manage these nodes and the underlying hardware that is used in the solution (bare metal nodes, switches, and more).

In particular, the solution has four server roles:

- ▶ System management node
- ▶ Hadoop management node
- ▶ Hadoop data node
- ▶ Apache Spark worker node

Table 2-1 describes the different types of nodes that are used in the different configurations of the solution.

*Table 2-1 Different types of node in the solution*

Type of node	Role
System management node	This node is the primary provisioning and monitoring node for the cluster. The Platform Cluster Manager (PCM) console that is used to deploy and monitor all of the analytics nodes runs on the system management node.
Analytic node: Hadoop management node	These nodes encompass daemons that are related to managing the cluster and coordinating the distributed environment.
Analytic node: Hadoop data node	These nodes encompass daemons that are related to storing data and accomplishing work within the distributed environment. The major difference between these two types of analytic nodes are in hardware and corresponding software configurations.
Analytic node: Apache Spark worker node	

The number of each type of node that is required within a big data cluster depends on the client requirements. Such requirements might include the size of a cluster, the size of the user data, the data compression ratio, workload characteristics, and data ingestion.

Each type of node has a specific hardware configuration that is tailored to fit its hierarchical role. Figure 2-1 presents the infrastructure view of the solution.

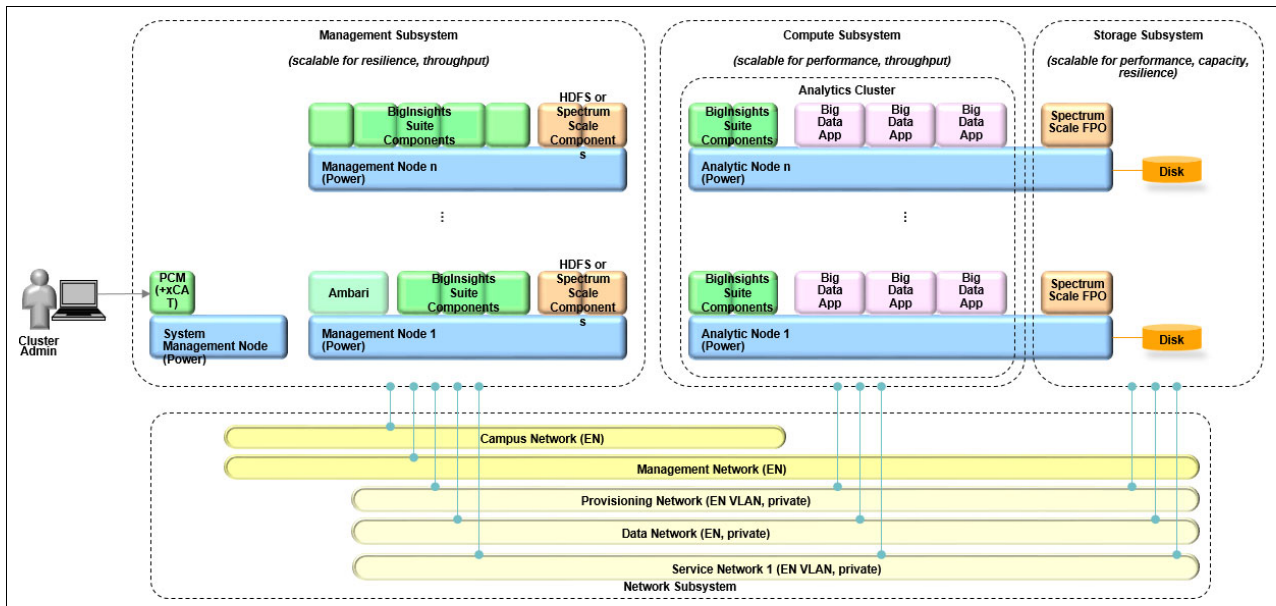


Figure 2-1 Infrastructure view of the solution

## 2.3 Hardware components of the solution

Selecting a hardware architecture to support the deployment of a big data and analytics solution requires an understanding of the relevant components and how they impact many aspects: performance, reliability, availability, and serviceability (RAS), costs, and management.

### 2.3.1 The IBM Power System S812LC server

The Power S812LC server with POWER8 processors is optimized for data and Linux. The server improves the management of Hadoop and Apache Spark workloads, has a system that is optimized for efficiency and designed for big data, and delivers superior performance and throughput for high-value Linux workloads, such as industry applications, open source, and Linux, Apache, MariaDB, and PHP (LAMP).

IBM Power System servers use the POWER8 chip, which has up to 8 - 10 cores per socket. With SMT8 technology, the POWER8 chip has eight threads per core (four times more than Intel) for running parallel Java workloads, which takes maximum advantage of the processing capability. The POWER8 chip has high memory and I/O bandwidth, which is critical for a big data system to achieve superior performance. The POWER8 architecture also offers better RAS than x86 servers.

By incorporating OpenPOWER foundation community innovations, the Power S812LC server has a low acquisition cost through system optimization (industry-standard memory, focused configurations, focused I/O and expansion, and industry-standard warranty), which makes it ideal for clients that want the advantages of running their applications on a platform that is designed and optimized for data and Linux.



The Power S812LC server is designed to deliver superior performance and throughput for cloud and business-critical applications with the only open standards-based system that ensures system utilization to achieve superior cloud economics.

The Power S812LC server is based on the POWER8 architecture. It is a high-efficiency, single-socket, 2U rack server and supports a maximum of 1 TB of memory. It has 14 SATA bays for either hard disk drives (HDDs) or solid-state drives (SSDs). Each bay includes mounting hardware for a 3.5-inch drive or a 2.5-inch drive (known as small form factor (SFF) drive). Twelve of the bays are in the front of the server, which are controlled by a PCIe RAID adapter and are hot-pluggable. In the IBM Data Engine for Hadoop and Spark architecture, these disks are used to store the distributed file system data (HDFS or IBM Spectrum Scale).

The two SATA bays in the rear of the server are not hot-pluggable, and scheduled downtime is required to add or remove safely a drive. These disks are used to install the OS.

In terms of management, the service processor or Baseboard Management Controller (BMC) is the primary control for autonomous sensor monitoring and event logging features on the Power S812LC server. BMC supports the Intelligent Platform Management Interface (IPMI 2.0) and Data Center Management Interface (DCMI 1.5) for system monitoring and management.

BMC monitors the operation of the firmware (FW) during the boot process and also monitors the hypervisor for termination. The FW code update is supported through the BMC and IPMI.

**Note:** The Power S812LC server and PowerKVM do not support IBM AIX® or IBM i guest virtual machines (VMs) and cannot be managed by an HMC.

The Power S812LC server is used to fulfill all logical server roles within the cluster: System Management Node, Hadoop Management Node, Hadoop Data Node, and Apache Spark worker node. However, different hardware configurations (memory and disks) are used depending on which role is fulfilled.

Figure 2-2 shows the front view of the Power S812LC server and its 12 facing disk bays.



Figure 2-2 Server front view of the Power S812LC server

**Note:** For more information about the Power S812LC server, see *IBM Power Systems S812LC Technical Overview and Introduction*, REDP-5284.

## 2.3.2 Networking

There are three distinct networks in this solution, each serving a specific role:

- ▶ **Service network:** This network is connected to the BMC on each IBM Power Systems server and switches' management ports. The service network allows the management software to manage and monitor the hardware on a 1-Gigabit Ethernet interconnect without requiring the node OS to be up. Typical hardware-level management functions include power-cycling the node, hardware status monitoring, FW configuration, and hardware console access.
- ▶ **Management network:** This network is used for provisioning the OS, deploying software components and applications, monitoring, and workload management. The management network uses a 1-Gigabit Ethernet interconnect.
- ▶ **Data network:** This high-performance network is used for accessing data in the cluster file system, communicating between analytics applications, and moving data in and out of the cluster. The data network uses 10-Gigabit Ethernet high-speed interconnects.

High-speed switch configurations are available to deliver fast movement of data at the scale that is required in Hadoop and Apache Spark clusters:

- ▶ Management network switch: 1 Gb Ethernet (48 x 1 Gb and 4 x 10 Gb ports)
- ▶ Data network switches:
  - 10 Gb Ethernet (24 x 10 Gb ports)
  - 10 Gb Ethernet (48 x 10 Gb and 4 x 40 Gb ports)

Figure 2-3 on page 21 shows the advanced network configuration diagram.

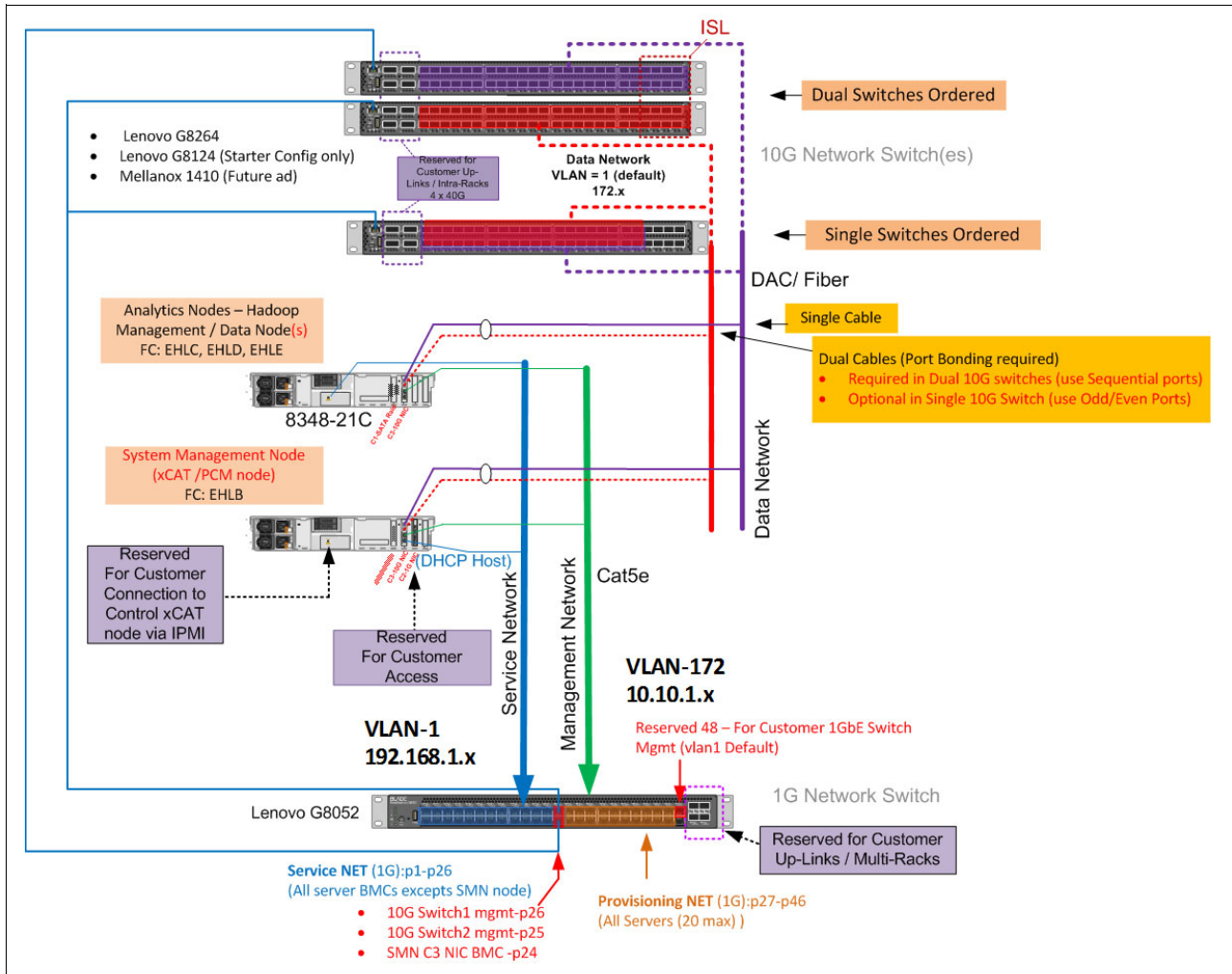


Figure 2-3 Detailed network diagram of the solution

The following racked-mounted switches are used to implement the network configuration of the solution:

- ▶ **Lenovo RackSwitch G8052**

1 GbE top-of-rack switch for the management and service networks. The connections that are needed for the service network virtual local area network (VLAN) are one physical link to the system management node, one physical link to the BMC of each server for out-of-band hardware management, and one physical link to each network switch for out-of-band switch management. The connections that are needed for the management network VLAN are one physical link to the system management node and one physical link for each analytics node.
- ▶ **Lenovo RackSwitch G8264 or G8124**

10 GbE top-of-rack switch for the data network. The connections that are needed for the data network are one or two physical links for the system management node, and one or two physical links for each analytics node. In general, the G8124 switch is used only for small clusters that do not have scaling requests. The G8264 switch is recommended for the future growth.

## 2.4 Software reference architecture

The following components are part of the software stack that is on the IBM Data Engine for Hadoop and Spark solution:

- ▶ PCM provides hardware management and monitoring functions and a web console.
- ▶ IBM Open Platform provides standard Hadoop services (such as Zookeeper, HBase, Hive, and more), its management console, and Ambari.
- ▶ IBM Spectrum Symphony (formerly IBM Platform Symphony) provides the MapReduce engine functions and job-related web console, and is available as a stand-alone item.
- ▶ IBM Spectrum Scale is an HDFS replacement for the solution and is available as stand-alone item.

Figure 2-4 shows the software stack running on the IBM Data Engine for Hadoop and Spark solution.

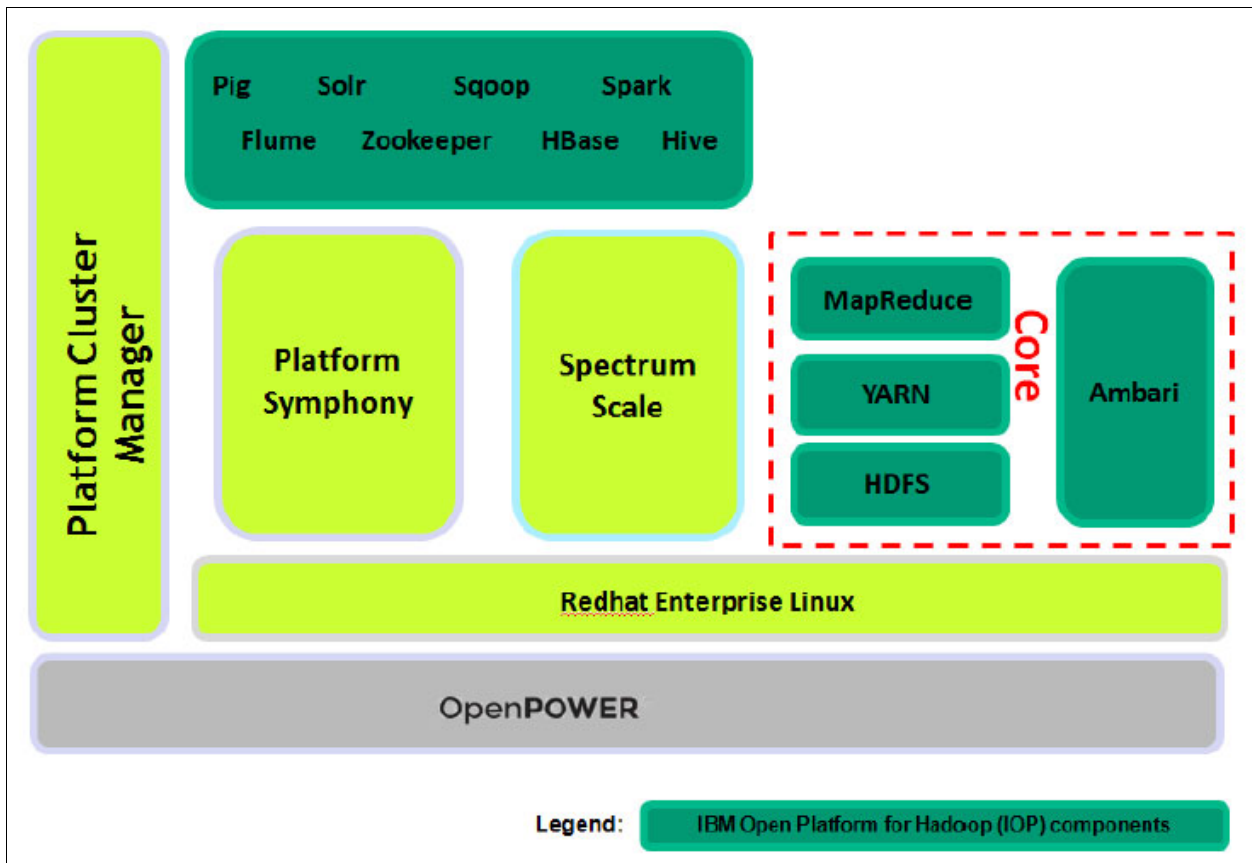


Figure 2-4 Overview of the software stack running on the solution

### 2.4.1 IBM Open Platform with Apache Hadoop clusters

IBM Open Platform with Apache Hadoop is a software platform for discovering, analyzing, and visualizing data from disparate sources. The solution is used to help process and analyze the volume, variety, and velocity of data that continually enters organizations every day.

IBM Open Platform with Apache Hadoop is a collection of value-added services, which is an open Hadoop foundation.

The IBM Open Platform with Apache Hadoop stack contains the following items:

- ▶ Native support for rolling upgrades for Hadoop services.
- ▶ Support for long-running applications within YARN for enhanced reliability and security.
- ▶ Heterogeneous storage in HDFS for in-memory, with SSD in addition to HDD.
- ▶ Apache Spark in-memory distributed compute engine for dramatic performance increases over MapReduce, which simplifies the developer experience and leverages the Java, Python, and Scala languages.
- ▶ Ambari operational framework for provisioning, managing, and monitoring Apache Hadoop clusters.
- ▶ The Apache Hadoop projects that are included are HDFS, YARN, MapReduce, Ambari, Hbase, Hive, Oozie, Parquet, Parquet Format, Pig, Snappy, Solr, Apache Spark, Sqoop, Zookeeper, Open JDK, Knox, and Slider.

IBM Open Platform with Apache Hadoop extends the Hadoop open source framework with enterprise-grade security, governance, availability, integration into existing data stores, tools that simplify developer productivity, and more.

**Note:** For information about IBM Open Platform with Apache Hadoop (Second generation), see the following website:

<http://ibm.co/29tTRd8>

## 2.4.2 Stand-alone products: IBM Spectrum Scale and IBM Spectrum Symphony

IBM Spectrum Scale and IBM Spectrum Symphony are stand-alone products that might fit into your IBM Open Platform with Apache Hadoop environment; if so, then it is installed and configured as the replacement for the HDFS file system and MapReduce engine in IBM Open Platform.

### IBM Spectrum Scale-File Placement Optimizer

IBM Spectrum Scale is software-defined storage for high performance, large-scale workloads on-premises or in the cloud. Built on the award-winning IBM General Parallel File System (GPFS), this scale-out storage solution provides file, object, and integrated data analytics for the following items:

- ▶ Compute clusters (technical computing)
- ▶ Big data and analytics
- ▶ HDFS
- ▶ Private cloud
- ▶ Content repositories

As the replacement for the HDFS file system in IBM Open Platform, IBM Spectrum Scale includes several enterprise features that provide distinct advantages. Some of these features are especially useful for managing and running a big data and analytics cluster:

- ▶ Full Portable Operating System Interface (POSIX) compliance:
  - Support for a wide range of traditional applications
  - Support for common UNIX utilities to manage content in the file system, such as copy, delete, and move
  - Allows HDFS data to be stored and accessed from the same file system as all other POSIX-compliant applications
- ▶ High-performance support for MapReduce applications and other traditional applications:
  - Supports striping data across disks to speed up MapReduce split I/O
  - Includes an optimized cache mechanism that increases the throughput of random read
  - Supports concurrent reads and writes by multiple programs
- ▶ Hierarchical storage management: Allows sufficient use of disk drives with different performance characteristics, such as the mixture of SSD and HDD.
- ▶ Data replication: Supports cluster-to-cluster replication over a wide area network (WAN), which provides the capability of some kind of disaster recovery.
- ▶ Snapshots: Snapshots can be taken of the file system with capabilities to do a global snapshot of an entire file system, and a snapshot can be created of a single independent file set.

IBM Spectrum Scale offers a distributed, scalable, reliable, and single namespace file system. IBM Spectrum Scale-File Placement Optimizer (IBM Spectrum Scale-FPO) is based on a shared-nothing architecture so that each node on the file system can function independently and be self-sufficient within the cluster. Typically, IBM Spectrum Scale-FPO can be a substitute for HDFS, removing the need for the HDFS NameNode, Secondary NameNode, and DataNode services.

However, in performance-sensitive environments, placing IBM Spectrum Scale metadata on higher-speed drives might improve the performance of the IBM Spectrum Scale file system. IBM Spectrum Scale-FPO has significant and beneficial architectural differences from HDFS.

HDFS is a file system that is based on Java that runs on top of the OS file system and is not POSIX-compliant. IBM Spectrum Scale-FPO is a POSIX-compliant, kernel-level file system that provides Hadoop with a single namespace, distributed file system with performance, manageability, and reliability advantages over HDFS. As a kernel-level file system, IBM Spectrum Scale is free from the impact that is incurred by HDFS as a secondary file system, running within a JVM on top of the OS' file system.

As a POSIX-compliant file system, files that are stored in IBM Spectrum Scale-FPO are visible to authorized users and applications by using standard file access/management commands and APIs. An authorized user can list, copy, move, or delete files in IBM Spectrum Scale-FPO by using traditional OS file management commands without logging in to Hadoop.

Additionally, IBM Spectrum Scale-FPO has significant advantages over HDFS for backup and replication. IBM Spectrum Scale-FPO provides point-in-time snapshot backup and off-site replication capabilities that enhance cluster backup and replication capabilities. When using IBM Spectrum Scale-FPO instead of HDFS as the cluster file system, the HDFS NameNode and Secondary NameNode daemons are not required on cluster management nodes, and the HDFS DataNode daemon is not required on cluster data nodes. Equivalent tasks are performed by IBM Spectrum Scale in a distributed way across all nodes in the cluster, including data ones. From an infrastructure design perspective, including IBM Spectrum Scale-FPO can reduce the number of management nodes that are required.

Because IBM Spectrum Scale-FPO distributes metadata across the cluster, no dedicated name service is needed. Management nodes within the IBM Open Platform with Apache Hadoop predefined configuration or HBase predefined configuration that are dedicated to running the HDFS NameNode or Secondary NameNode services can be eliminated from the design. The reduced number of required management nodes can provide sufficient space to allow for more data nodes within a rack.

## IBM Spectrum Symphony

As the replacement for the MapReduce engine in IBM Open Platform, IBM Spectrum Symphony also provides some distinctive advantages.

IBM Spectrum Symphony can run distributed application services on a scalable, shared, heterogeneous grid. This low-latency scheduling solution supports sophisticated workload management capabilities beyond those of standard Hadoop MapReduce.

IBM Spectrum Symphony can orchestrate distributed services on a shared grid in response to dynamically changing workloads. This component combines a service-oriented application middleware (SOAM) framework, a low-latency task scheduler, and a resource orchestration layer (the IBM Spectrum Computing resource manager, also known as Enterprise Grid Orchestrator (EGO)). This design ensures application reliability while ensuring low-latency and high-throughput communication between clients and compute services.

Hadoop has limited prioritization features, but IBM Spectrum Symphony has thousands of priority levels and multiple options that you can configure to manage resource sharing. This sophisticated resource sharing allows you to prioritize for interactive workloads that are not possible in a traditional MapReduce environment. For example, with IBM Spectrum Symphony, you can start multiple Hadoop jobs and associate those jobs with the same consumer. Within that consumer, jobs can share resources based on individual priorities.

The scheduling framework of IBM Spectrum Symphony is optimized for MapReduce workloads that are compatible with Hadoop and Apache Spark.

Through its YARN and EGO integration plug-in, IBM Spectrum Symphony also supports running Apache Spark workloads with intelligent resource management on the IBM Data Engine for Hadoop and Spark solution.

**Note:** For more information about the multitenancy capability that is provided by IBM Spectrum Symphony, see Chapter 5, “Multitenancy” on page 77.

### 2.4.3 Cluster management

The Power System nodes in the IBM Data Engine for Hadoop and Spark solution are managed by PCM-AE. The BMC is used as the hardware entry point for system monitoring and management.

PCM-AE provides the following benefits:

- ▶ Management of multitenancy environments: You can create multiple, isolated clusters within your server farm.
- ▶ Support for deploying multiple products.
- ▶ On-demand and self-service provisioning: You can create cluster definitions and use them to deploy automatically the cluster nodes. A person with little or no cluster setup knowledge can then quickly deploy a cluster environment.
- ▶ Increased server consolidation: By being able to grow or shrink dynamically a cluster environment, you minimize the amount of idle resources because of the creation of siloed clusters.

PCM-AE uses Extreme Cluster/Cloud Administration Toolkit (xCAT) to manage the Power System servers through their BMC by way of the IPMI protocol. Therefore, it is possible to take advantage of xCAT commands to manage the installed nodes from the management node.

**Note:** PCM-AE provides a GUI to visualize and manage the node from a centralized management node.

The PCM-AE administration node (system management node) is in charge of deploying the bare-metal nodes of the cluster with the OS. The rest of the big data software stack is deployed through Ambari from one of the Hadoop management nodes of the configuration.

### 2.4.4 Additional analytics software: IBM Spectrum Conductor with Spark

To run a Apache Spark workload, as an alternative to the combination of IBM Open Platform with Apache Hadoop, IBM Spectrum Conductor with Spark can be ordered separately and installed on the IBM Data Engine for Hadoop and Spark solution.

IBM Spectrum Conductor with Spark is a complete enterprise-grade multitenant solution for Apache Spark.

IBM Spectrum Conductor with Spark delivers the following benefits:

- ▶ Accelerate results: Run Apache Spark natively on a shared infrastructure without the dependency of Hadoop, which helps reduce application wait time, and increases time to results.
- ▶ Reduce administration costs: Proven architecture at extreme scale, with enterprise class workload management, monitoring, reporting, and security capabilities.
- ▶ Increase resource utilization: Fine grain, dynamic allocation of resources maximizes efficiency of Apache Spark instances sharing a common resource pool. Extends beyond Apache Spark and eliminates cluster sprawl.
- ▶ End to end enterprise class solution: A tightly integrated offering that combines the IBM supported Apache Spark distribution with workload, resource and data management, and IBM support and services.



IBM Spectrum Conductor with Spark integrates notebook functions, and takes advantage of a notebook's GUI to manipulate and visualize data.

Beside the built-in Apache Zeppelin notebook, you can use third-party notebooks such as iPython-Jupyter.

## 2.4.5 Software options

The base software components that are installed and configured in this solution are listed in Table 2-2.

Table 2-2 Base software component

Name	Mode	Description
RHEL V7.2 ppc64le	Required	Red Hat Enterprise Linux V7.2 for Power (Little Endian).
PCM - AE	Required	Used for bare metal deployment of the cluster nodes.
IBM Open Platform with Apache Hadoop	Default	The IBM Open Platform with Apache Hadoop provides the Apache Hadoop open source components such as Apache Ambari, HDFS, Flume, Hive, and ZooKeeper.
IBM Spectrum Scale	Selectable	IBM Spectrum Scale is software-defined storage for high performance, large-scale workloads for on-premises or in the cloud.
IBM Spectrum Symphony	Selectable	IBM Spectrum Symphony provides an application framework that you can use to run distributed or parallel applications in a scaled-out grid environment.

The solution provides various software combinations. Each combination is composed of a set of software feature codes. The supported feature code is listed in Table 2-3.

Table 2-3 Software feature code

Feature code	Feature name	Default	Min	Max
EHLF	IBM Open Platform with Apache Hadoop Indicator	1	0	1

The following software combination feature matrix is supported:

1. EHLF  
This is the default installation, which uses HDFS in IBM Open Platform with Apache Hadoop as the file system.
2. Stand-alone products IBM Spectrum Scale and IBM Spectrum Symphony  
Install IBM Open Platform by using IBM Spectrum Scale as the file system and IBM Spectrum Symphony as the resource manager and MapReduce engine. IBM Spectrum Scale and IBM Spectrum Symphony are both installed as stand-alone products.
3. Operating system only install  
Provision the node so that the RHEL V7.2 ppc64le OS is installed and networking is configured. No additional software must be installed.

## 2.5 Solution reference architecture

The solution is based on a set of standard building blocks that can be tailored to the data size, throughput, and scalability that is required for the target analytics scenarios.

Five-node starter configurations are available, handling up to 216 TB of raw data and providing over 50 TB of usable data in a standard triple replica Hadoop or Apache Spark configuration.

Multi-rack configurations are available, providing up to 1.3 PB of raw data per rack.

### 2.5.1 Configuration

A typical supported configuration consists of the following components:

- ▶ Rack and power supply
- ▶ System management node:
  - Power S812LC server
  - Eight 3.32 GHz cores
  - 32 GB memory (default), with a maximum memory of 1 TB
  - Two 1 TB 3.5-inch SATA HDDs
  - One Shiner-S Ethernet adapter with two 10-Gigabit ports and two 1-Gigabit ports
- ▶ Hadoop management node:
  - Power S812LC server
  - Ten 2.92 GHz cores
  - 128 GB memory (default), with a maximum memory of 1 TB
  - Two 1 TB 3.5-inch SATA HDDs
  - One Shiner-S Ethernet adapter with two 10-Gigabit ports and two 1-Gigabit ports
- ▶ Hadoop data node:
  - Power S812LC server
  - Ten 2.92 GHz cores
  - 128 GB Memory (default), with a maximum memory of 1 TB

- Two 1 TB 3.5-inch SATA HDDs
- Twelve 6 TB 3.5-inch SATA HDDs
- One Shiner-S Ethernet adapter with two 10-Gigabit ports and two 1-Gigabit ports
- One PMC-Sierra 71605E RAID adapter, with over 530 K IOPS and up to 6.6 GBps reads and 5.7 GBps writes
- ▶ Apache Spark worker node:
  - Power S812LC server
  - Ten 2.92 GHz cores
  - 256 GB memory (default), with a maximum memory of 1 TB
  - Two 1 TB 3.5-inch SATA HDDs
  - Ten 6 TB 3.5-inch SATA HDDs
  - Two 960 GB SSDs
  - One Shiner-S Ethernet adapter with two 10-Gigabit ports and two 1-Gigabit ports
  - One PMC-Sierra 71605E RAID adapter, with over 530 K IOPS and up to 6.6 GBps reads and 5.7 GBps writes
- ▶ Network switch:
  - Lenovo RackSwitch G8052: Forty-eight 1-Gigabit and four 10-Gigabit Ethernet top-of-rack switches for the management and service network
  - Lenovo RackSwitch G8264: Forty-eight 10-Gigabit and four 40-Gigabit Ethernet top-of-rack switches for data network
  - Lenovo RackSwitch G8124: Twenty-four 10-Gigabit Ethernet top-of-rack switches for data network, and for smaller configurations without a scaling request
- ▶ Software:
  - PCM - AE
  - IBM Open Platform
  - IBM Spectrum Scale
  - IBM Spectrum Symphony Advanced Edition

### **System management node**

The system management node can be deployed in a Power S812LC server. One system management node is sufficient for a cluster of up to 128 analytic nodes. Advanced cluster management software is a standard component of the solution.

### **Analytics node: Hadoop management node**

In this solution, Hadoop management nodes run on Power S812LC servers. Hadoop management nodes encompass the following services:

- ▶ HDFS NameNode
- ▶ HDFS Secondary NameNode
- ▶ YARN ResourceManager
- ▶ IBM Spectrum Symphony Service-Oriented Application Middleware (SOAM) workload management services, such as Session Director (SD), Repository Service (RS), and Service Session Manager (SSM)
- ▶ IBM Spectrum Symphony EGO services, such as VEM Kernel Daemon (VEMKD) and EGO Service Controller (EGOSC)

- ▶ IBM Spectrum Symphony Platform Management Console (PMC)
- ▶ IBM Spectrum Symphony Reports
- ▶ Ambari server
- ▶ HBase master
- ▶ Hive server
- ▶ Other services, including Zookeeper, Oozie, and so on

IBM Spectrum Scale is a distributed file system that is supported by this solution as an alternative to HDFS. Because IBM Spectrum Scale uses different methods than HDFS to manage metadata, it does not have a centralized NameNode and does not require a Secondary NameNode.

IBM Spectrum Symphony SOAM is composed of workload management services and workload execution services.

Workload execution services are required on all data nodes. Workload management services are usually installed on a limited number of management nodes. By default, a minimum of three management nodes are recommended for the Landing zone configuration if high availability is not used. When considering cluster scaling, more management nodes can be added so that there are more SSM daemons to respond simultaneously to user jobs.

IBM Spectrum Symphony PMC is the centralized web interface for viewing, monitoring, and managing MapReduce jobs and the MapReduce running environment.

IBM Spectrum Symphony Reports provides the reporting functions by collecting historical data into a database and generating reports in graphical or tabular formats.

For a production cluster, a minimum of three Hadoop management nodes are required. If high availability is wanted, six Hadoop management nodes are required. A single Hadoop management node is recommended only for small and non-production clusters.

For a cluster with multiple racks and multiple management nodes, consult with the IBM Service team for more information about scaling.

### **Analytics node: Hadoop data node**

In this solution, the third type of analytics node is the *data node*. Data nodes run on Power S812LC servers, with one data node logically partitioned by using all of the resources of a single server. Data nodes encompass the following services:

- ▶ HDFS DataNode
- ▶ YARN NodeManager
- ▶ HBase RegionServer
- ▶ IBM Spectrum Scale Network Shared Disk (NSD) servers
- ▶ IBM Spectrum Symphony SOAM execution management services, such as Session Instance Manager (SIM) and Service Instance (SI)
- ▶ IBM Spectrum Symphony EGO services, such as Load Information Manager (LIM) and Process Execution Manager (PEM)
- ▶ Other services for IBM Open Platform

IBM Spectrum Scale is the POSIX-compatible distributed file system for this solution. It provides the HDFS access API for Hadoop services and workloads. The data nodes host IBM Spectrum Scale NSD servers, which are used for storage for the file system. A file placement optimization (FPO) configuration is used to allow data locality and at the same time maintain the distributed, fault-tolerant nature that is inherent to a IBM Spectrum Scale file system.

IBM Spectrum Symphony service-oriented architecture (SOA) execution management services are installed on all data nodes to support running and managing the MapReduce workloads that are scheduled by upper-level SOA workload management services.

IBM Spectrum Symphony EGO services are installed on all data nodes to collect computational resource information and to help the IBM Spectrum Symphony SOA workload management service to schedule jobs more quickly and efficiently.

There are also other services for IBM Open Platform with Apache Hadoop that can be installed in data nodes, depending on the application requirements for the big data cluster, must be installed, configured, and run on all data nodes.

### **Analytics node: Apache Spark worker node**

Apache Spark worker nodes run the same services as Hadoop data nodes, but these services are configured to use the extra memory and SSDs for Apache Spark workloads.

The total amount of disk space in the cluster increases linearly when you increase the number of specific data nodes. Table 2-4 provides the raw and effective disk capacity with the default configuration for the Hadoop data node and the Apache Spark worker node.

*Table 2-4 Disk space with Hadoop data node and Apache Spark worker node*

<b>Data node type</b>	<b>Raw disk space (TB)</b>	<b>Effective disk space (TB)</b>
Hadoop data node (twelve 6 TB HDDs)	72	54
Apache Spark worker node (ten 6 TB HDDs)	60	45

The effective disk space is calculated by accounting for the allocation of 25% of the raw disk space as shuffle file space, so the effective disk space is 75% of the raw disk space.

For the total usable space in a complete IBM Data Engine for Hadoop and Spark cluster, see 2.5.3, “Sizing the solution” on page 35.

## 2.5.2 Predefined configurations

There are two predefined configurations for this solution: Starter and Landing Zone.

- ▶ The Starter configuration is a small cluster that is enabled for future growth. It can be used for a test or development environment on a single rack configuration. This configuration starts with five nodes with over 200 TB (65 TB usable) of storage. The Starter configurations with Hadoop data nodes are described in Figure 2-5. Figure 2-6 on page 33 describes the Starter configuration with Apache Spark worker nodes.

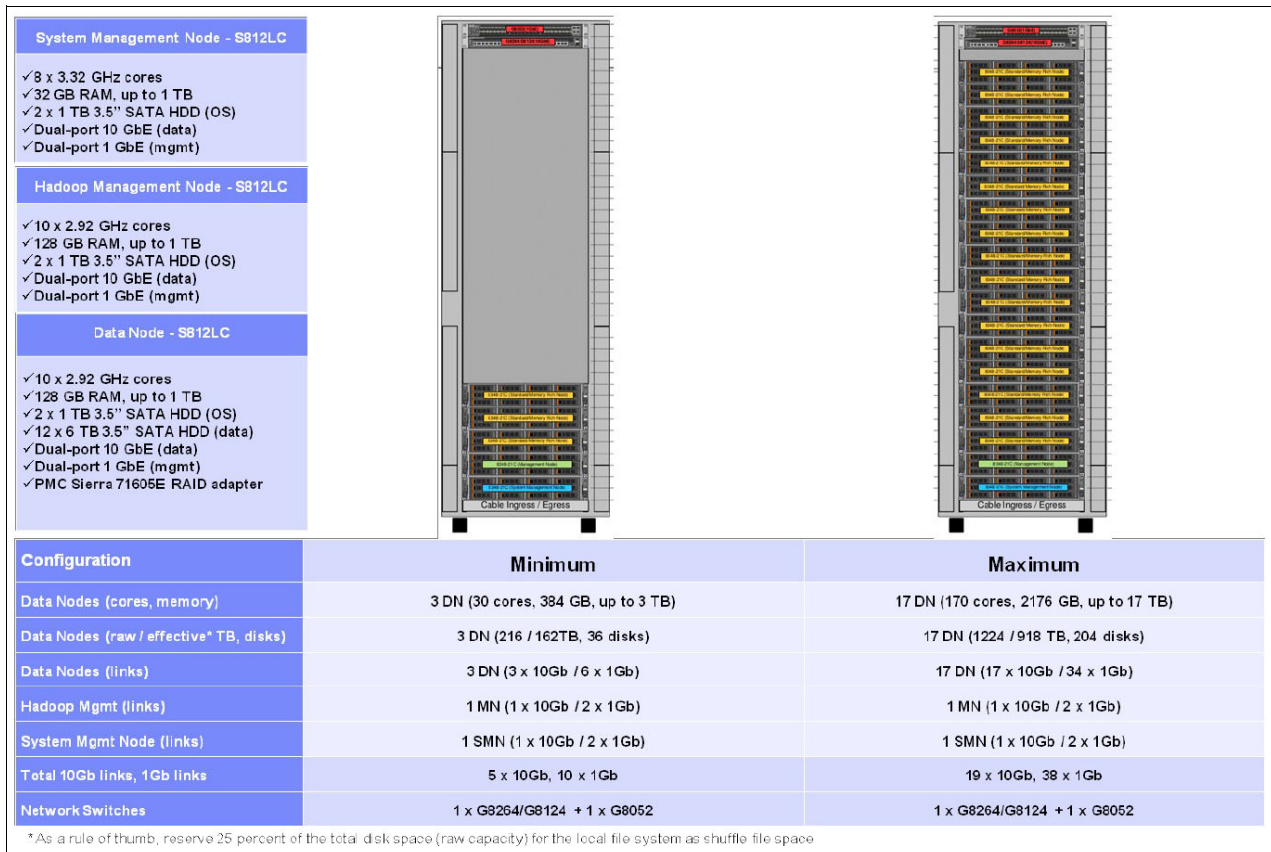


Figure 2-5 Starter configuration with Hadoop data node

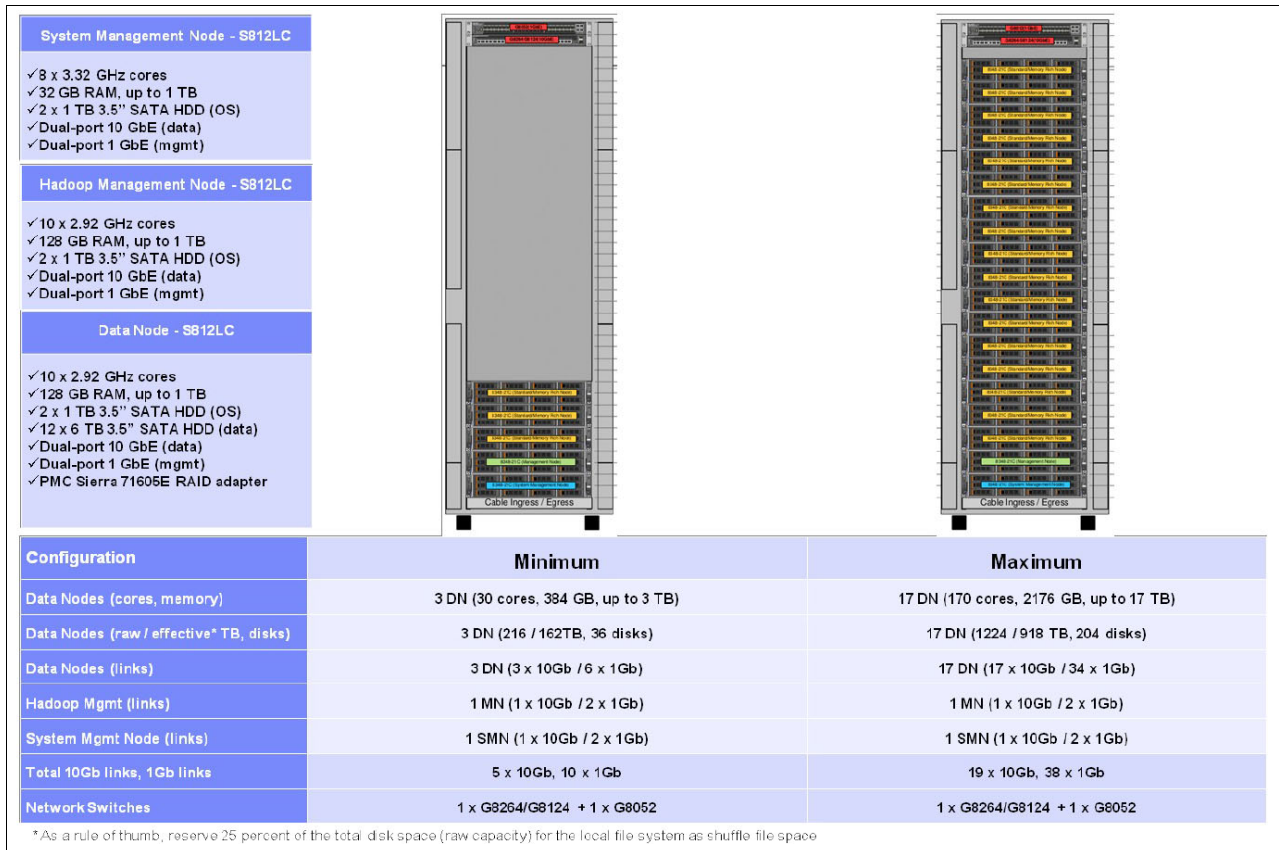


Figure 2-6 Starter configuration with Apache Spark worker nodes

- ▶ The Landing Zone configuration adds redundancy to the management nodes and to the networks and is tailored for production workloads. This configuration has a minimum of 14 nodes. The Landing Zone configurations with Hadoop data nodes are shown in Figure 2-7. Figure 2-8 on page 35 shows the Landing zone configurations with Apache Spark worker nodes.

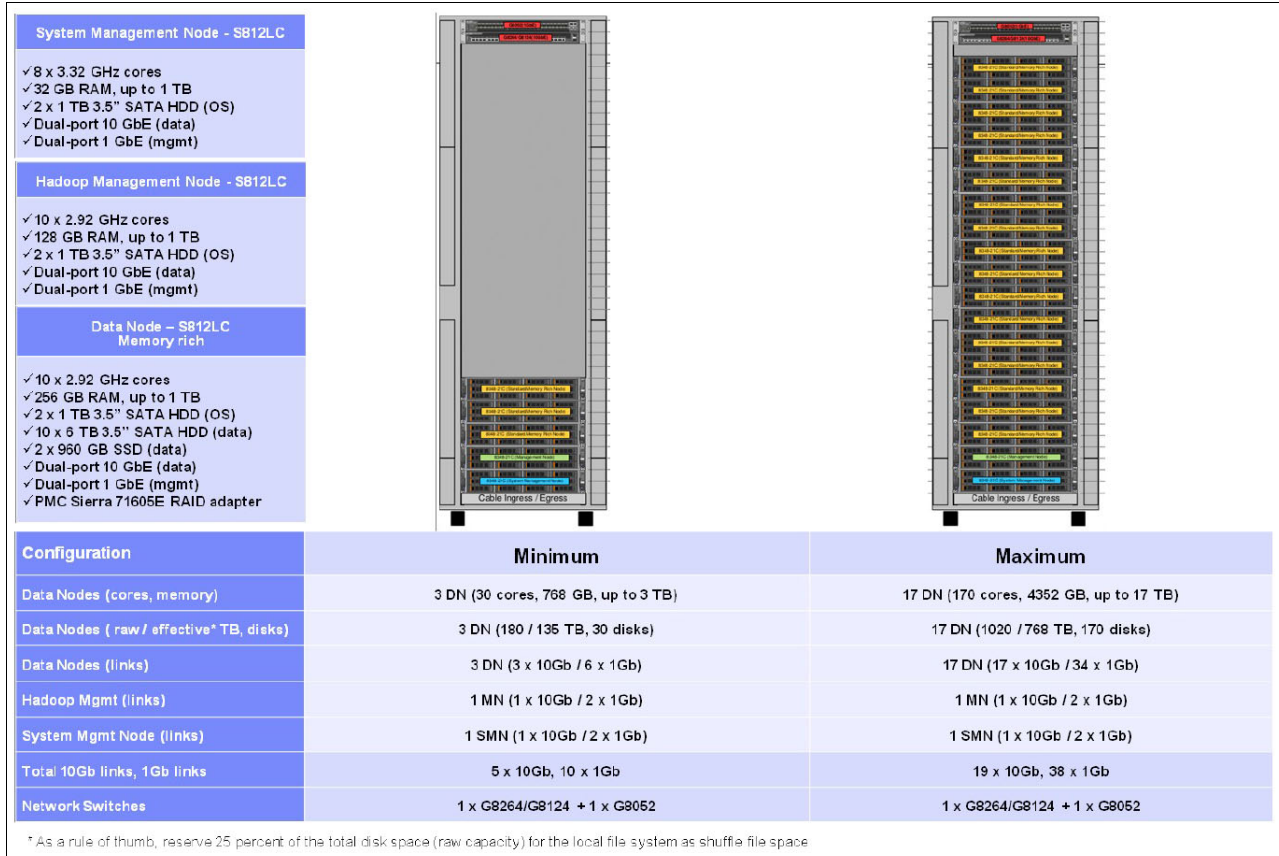


Figure 2-7 Landing Zone configuration with Hadoop worker nodes



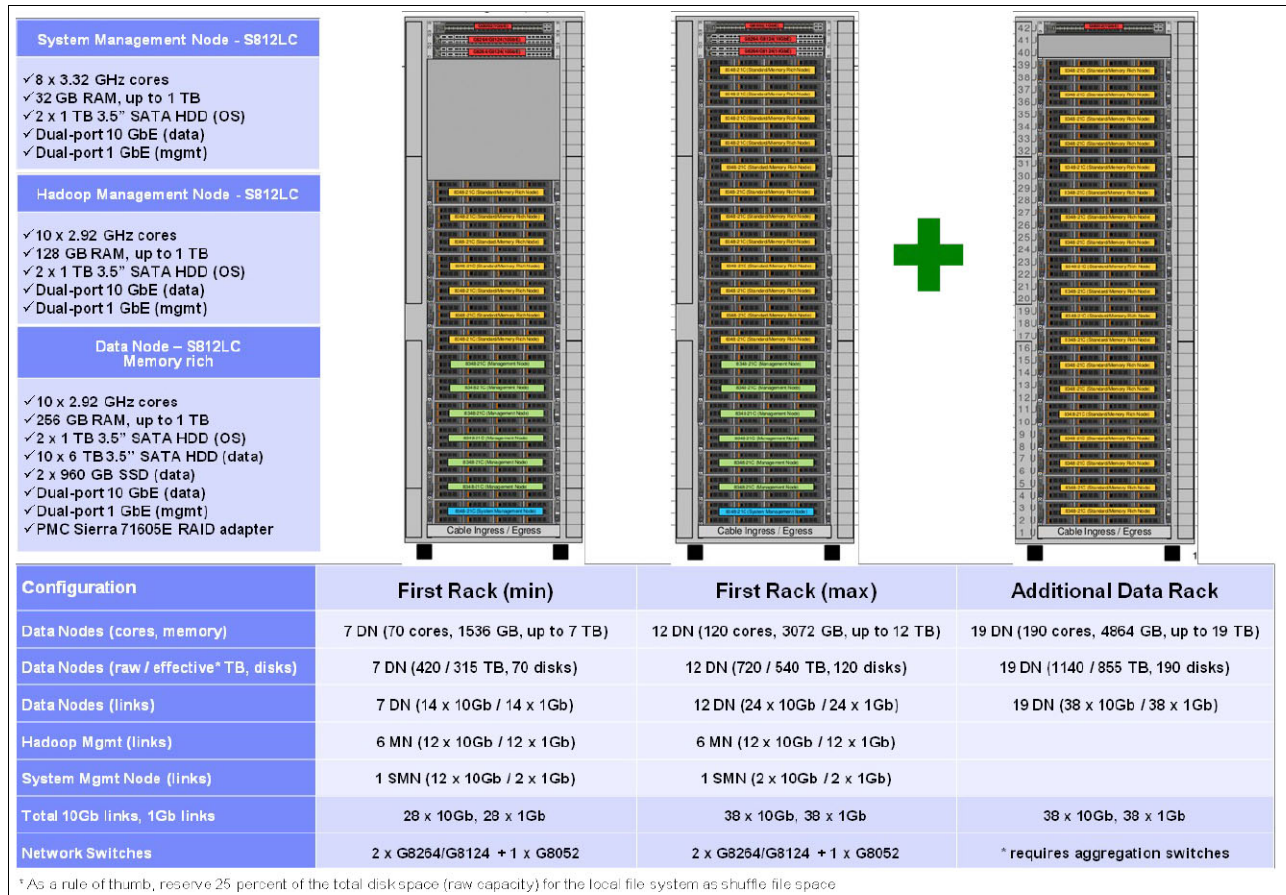


Figure 2-8 Landing Zone configuration with Apache Spark worker nodes

### 2.5.3 Sizing the solution

To size correctly which configuration fits the workload requirements, the following aspects must be considered:

- ▶ Will the system be a production system?
- ▶ Will the workloads be Hadoop or Apache Spark oriented?
- ▶ What is the raw data size required?
- ▶ What is the anticipated data growth rate?

The total disk space is a metric that can be used to drive the sizing. When estimating disk space within a Hadoop cluster, consider the following points:

- ▶ For improved fault tolerance and improved performance, HDFS replicates data blocks across multiple cluster data nodes. By default, HDFS maintains three replicas. This solution uses the default setting. If IBM Spectrum Scale is used, three replicas are also maintained by default.
- ▶ With the MapReduce process, shuffle or sort data is passed from Mappers to Reducers by writing the data to the data node's local file system. If the MapReduce job requires more than the available shuffle file space, the job terminates. As a rule of thumb, reserve 25 percent of the total disk space for the local file system as shuffle file space.

- ▶ The actual space that is required for shuffle or sort data is workload-dependent. In the unusual situation where the 25 percent rule of thumb is insufficient, available space on the OS drives can be used to provide more shuffle or sort space.
- ▶ The compression ratio is an important consideration in estimating disk space. Within Hadoop, the user data and the shuffle or sort data can be compressed. If the client's data compression ratio is not available, assume a compression ratio of 2.5.

Assuming that the default replicas are maintained by HDFS (if IBM Spectrum Scale is used, the same rule applies), the total cluster data space and the required number of data nodes can be estimated by using the following equations:

- ▶ Total Data Disk Space = (User Raw Data, Uncompressed) / 0.75 x (number of replicas) / (compression ratio)
- ▶ Total Required Data Nodes = (Total Data Disk Space) / (Data Space per Server)

The Starter configuration has only one management node and a non-redundant 10 GB data network. If the system is targeted to be a cluster, which is small initially and required to grow in future, and if it is used for proof of concept, development, testing, or evaluation, then the Starter configuration is preferable. This configuration starts with a minimum of one system management node, one Hadoop management node, and three data nodes. It can grow to one full rack with one system management node, one Hadoop management node, and 17 data nodes.

The Landing Zone configuration starts with a minimum of one system management node, six Hadoop management nodes, and seven data nodes. It provides redundancy at the network, software, and hardware level. It can be scaled easily by adding the extended rack of the Landing Zone configuration. If the system is targeted for production or other high available workloads, the Landing Zone configuration is preferable because it has a good cost per gigabyte, features, and performance.

If the estimated cluster size is more than two racks, contact IBM for help with planning the network and services.

## 2.5.4 Rack, power, and cooling information

Here is the rack, power, and cooling information:

- ▶ Included with the rack:
  - 7014 T42 42U 19-inch Enterprise IBM Rack.
  - Rails for 19-inch rack are included. No feature code is required.
  - The rack is ballasted so that the network switches are over 32U.
  - Power S812LC servers currently are not rack-shippable (server integration happens on the customer floor).
- ▶ Environmental:
  - Server acoustics are rated at 6.1 - 9.3 dB.
  - There should be acoustic doors on rack, especially if running heavy workloads (FC #EC07, FC #EC08).
- ▶ PDU:
  - 7109 Intelligent PDU+, 1 EIA Unit, Universal, UTG024.
  - 5889 Intelligent PDU, Universal, 1-PH 24/48A, 3-PH 16/24A.



# Use case scenario for the IBM Data Engine for Hadoop and Spark

This chapter gives references about what you can do by using IBM Data Engine for Hadoop and Spark. This chapter gives information about which features are suitable for the planned workloads and how IBM Data Engine for Hadoop and Spark features can help achieve the planned objectives.

The following topics are described in this chapter:

- ▶ When to use IBM Data Engine for Hadoop and Spark
- ▶ When to use Hadoop and what workloads are suitable for it
- ▶ When to use Apache Spark and what workloads are suitable for it
- ▶ Greater resource utilization by using IBM Spectrum Symphony
- ▶ Comparing Hadoop Distributed File System and IBM Spectrum Scale
- ▶ Using the analytic capabilities of IBM Open Platform

### 3.1 When to use IBM Data Engine for Hadoop and Spark

IBM Data Engine for Hadoop and Spark is a fully integrated infrastructure solution with integrated cluster management and analytics software that delivers competitive price and performance for Hadoop-based and Apache Spark-based workloads while improving ease of deployment and cluster operational simplicity for clients deploying big data and analytics applications to support their line of business (LOB).

IBM has several offerings for big data that you can choose from depending on your needs. Figure 3-1 shows the available offerings.

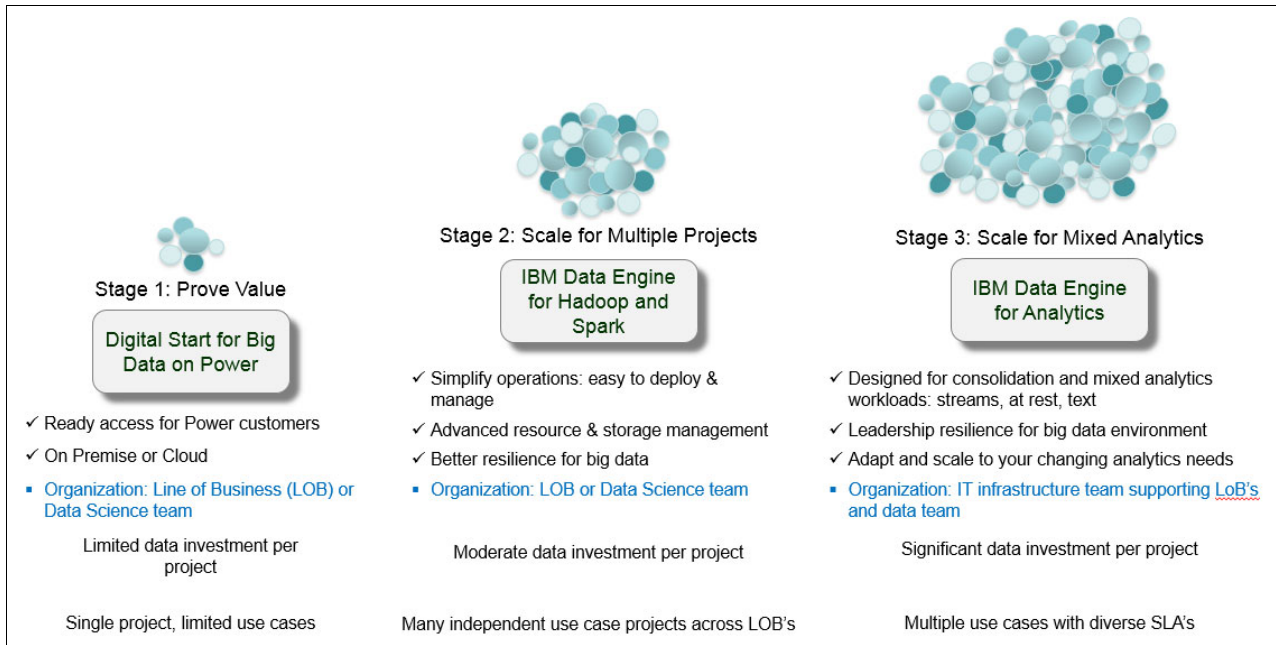


Figure 3-1 IBM Big Data on IBM Power Systems offering

### 3.2 When to use Hadoop and what workloads are suitable for it

Hadoop becomes beneficial when you must handle a large volume of data that consists of structured and unstructured data. The following section describes sample scenarios where Hadoop can be used.

#### 3.2.1 Landing Zone

As your system grows, you notice that there are many new sources of data and the volume grows. Normally, you must decide which data must be processed without discarding potentially useful data. Hadoop as a Landing Zone helps with this scenario by putting all the incoming data into Hadoop regardless of its source, format, or size. When the data is in Landing Zone, then it can be analyzed by using analytic tools or further processed according to your needs.

Figure 3-2 illustrates how Hadoop is used as a Landing Zone. The box at the lower left shows Hadoop being used to capture various data in its native format. Developers and analysts can use built-in analytic and data management technologies explore this raw data in a sandbox environment. Sources of data can include web pages, system logs, input from streaming engines (shown at upper left), and even reference data that is pulled from traditional sources, such as data warehouses (DWs) or data marts.

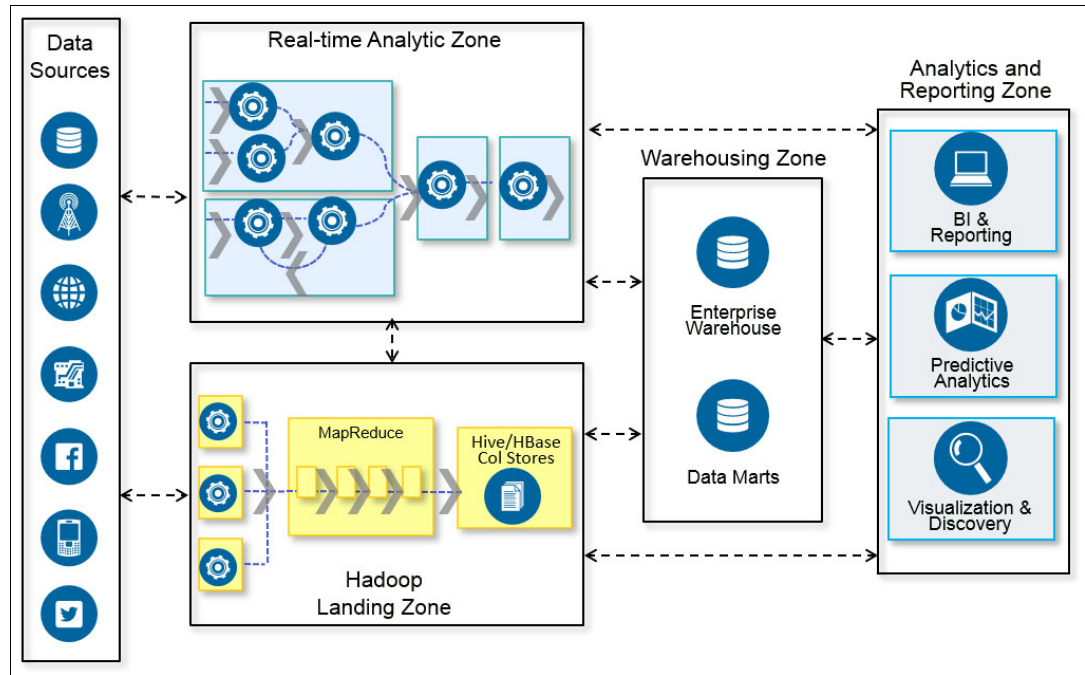


Figure 3-2 Landing Zone architecture

### 3.2.2 Data warehouse offloading

As an organization grows, so does its data, particularly data in the DW, which is used for analytics. However, keeping data in a DW is costly. But, discarding data from a DW is not an option because an analytic solution needs more data to ensure its effectiveness. Hadoop helps by providing a place to offload data from the DW, especially cold data (data that is not actively used) while providing the capability for analytic applications to access these data.

Figure 3-3 shows how Hadoop relates to the DW. This integration pattern involves offloading *cold* or infrequently accessed warehouse data to Hadoop, which turns Hadoop into a query-ready archive environment.

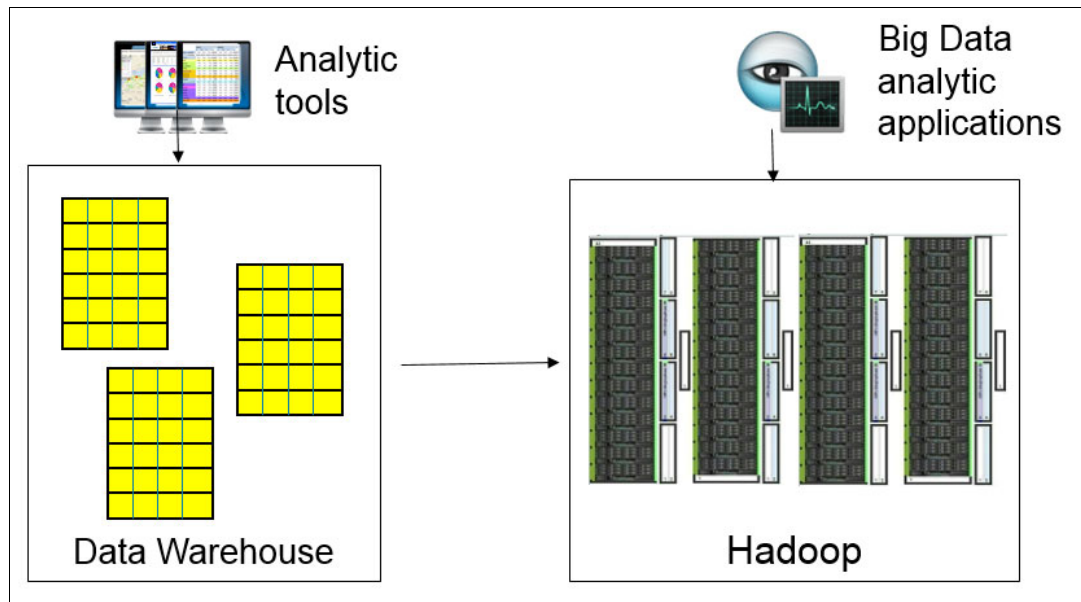


Figure 3-3 Data warehouse offloading architecture

### 3.3 When to use Apache Spark and what workloads are suitable for it

Apache Spark provides a way to access and process data. The result of this processing can then be stored in any available repository. Apache Spark shows great performance due to its in-memory capability. In addition, Apache Spark provides better productivity because it has a set of pre-built functions that help developers focus on their problem rather than building everything from scratch. Apache Spark can be integrated with Scala, Python, R, and Java. Apache Spark is also flexible in running workloads because it supports batch, interactive, iterative, and micro-batch workflows. Apache Spark can take advantage of a cluster environment, making it a good fit in a Hadoop environment.

Here are a few examples of Apache Spark use cases:

- ▶ Interactive query:
  - Enterprise-scale data volumes are accessible to interactive query for business intelligence.
  - Faster time to job completion allows analysts to ask the *next* question about their data and business.
- ▶ Large-scale batch:
  - Data cleaning to improve data quality (missing data, entity resolution, unit mismatch, and more).
  - Nightly extract, transform, and load (ETL) processing from production systems.

- ▶ Complex analytics:
  - Forecasting versus *Nowcasting*.
  - Data mining across various types of data.
- ▶ Event processing:
  - Web server log file analysis (human-readable file formats that are rarely read by humans) in near-real time.
  - Responsive monitoring of RFID-tagged devices.
- ▶ Model building:
  - Predictive modeling answers questions of *what will happen?*
  - Self-tuning machine learning, continually updating algorithms, and predictive modeling.
- ▶ Iterative analytic:
  - Build and deploy rich analytic models from iterative algorithms or programs, which must access the same set of large-scale data repeatedly.
  - Data mining and insight discovery process, which need iteratively run complex analytics and experiment with diverse data sources.

### 3.4 Greater resource utilization by using IBM Spectrum Symphony

IBM Spectrum Symphony (formerly IBM Platform Symphony) is a stand-alone product that fits into your IBM Data Engine for Hadoop and Spark solution. IBM Spectrum Symphony helps you achieve higher resource utilization so that you can get more benefits from multiple nodes inside IBM Data Engine for Hadoop and Spark, which is important for multitenancy needs. For more information about using IBM Spectrum Symphony for multitenancy, see Chapter 5, “Multitenancy” on page 77.

### 3.5 Comparing Hadoop Distributed File System and IBM Spectrum Scale

IBM Data Engine for Hadoop and Spark provides the options to use either Hadoop Distributed File System (HDFS) or IBM Spectrum Scale as the file system. HDFS is the standard distributed file system that is provided by Apache Hadoop in which Hadoop stores data. HDFS provides a distributed file system that spans all of the nodes within a Hadoop cluster, linking the file systems on many local nodes to make one large file system with a single namespace. IBM Spectrum Scale provides a fully compatible HDFS API. However, it also enhances HDFS by supplying a fully Portable Operating System Interface (POSIX)-compliant distributed file system, in addition to allowing HDFS style access.

IBM Spectrum Scale has a File Placement Optimizer (FPO) feature that extends its capability to support big data workloads by providing the following innovations:

- ▶ Locality awareness to allow compute jobs to be scheduled on nodes where the data is
- ▶ Metablocks that allow large and small block sizes to coexist in the same file system to meet the needs of different types of applications
- ▶ Write affinity that allows applications to dictate the layout of files on different nodes to maximize both write and read bandwidth

- ▶ Pipelined replication to maximize the use of network bandwidth for data replication
- ▶ Distributed recovery to minimize the effect of failures on ongoing computation

Table 3-1 shows a comparison of HDFS and IBM Spectrum Scale.

*Table 3-1 Compare HDFS and IBM Spectrum Scale*

<b>Aspects</b>	<b>HDFS</b>	<b>IBM Spectrum Scale</b>
Data locality	Supported	Supported.
HA	Supported	Provides strong fault tolerance.
Federation	Supported	Better federation capability.
Snapshot	Supported	Supported.
NFSv3	Supported	Supported.
WebHDFS	Supported	Supported.
Heterogeneous storage	Phasel	IBM Spectrum Scale can put different storage disks in to different storage pools and use a policy to control what data is placed in SSD, and what data is placed in SATA/SAS.
Memory caching	Not supported	IBM Spectrum Scale self-manages the memory cache in its pagepool.
Access control list	Supported	Supported.
Archival storage	Supported	IBM Spectrum Scale provides a better Information Lifecycle Management (ILM) policy to manage data in different storage pool.
Encryption	Supported	IBM Spectrum Scale encryption is at the file-set level and provides a better key management mechanism.
Truncate	Supported	Supported.
Quote per storage type	Not supported	Quota supports user, group, and file set.
Variable-length blocks	Not supported	If you have different storage types, you can create different storage pools from different storage types and place the file-set data over specific pools.
POSIX compliant	Not supported	Supported.



## 3.6 Using the analytic capabilities of IBM Open Platform

For more information about implementing the integration of analytic applications with IBM Open Platform with Apache Hadoop, see *Implementing an Optimized Analytics Solution on IBM Power Systems*, SG24-8291.





# Operational guidelines

This chapter explains how to install the IBM Data Engine for Hadoop and Spark infrastructure building blocks entry-level solution.

This chapter complements available documentation, and it is not intended to replace the solution manuals. The goal of this chapter is to provide you with additional details to help you implement and manage the solution.

The following topics are described in this chapter:

- ▶ Introduction
- ▶ Adding a compute node
- ▶ Configuring the Apache Spark UI
- ▶ Deployment and operation tools

## 4.1 Introduction

This chapter provides management details for the IBM Data Engine for Hadoop and Spark solution, and incorporates many topics that come from studies and experiences during the residency.

This chapter complements the available documentation for the solution. It also covers topics that are preferred practices for managing the solution.

## 4.2 Adding a compute node

This section describes how to add a compute node to an installed and configured cluster. The compute node that is added in this exercise is named *dn04*.

**Note:** This task can be performed by IBM. If you prefer that IBM performs this task, contact your account representative.

IBM Platform Cluster Manager (PCM) uses the Extreme Cluster/Cloud Administration Toolkit (xCAT). This exercise uses xCAT commands to prepare, deploy, and configure the new compute node.

**Note:** This setup assumes that the cabling and the network setup on the switches are done. If you need to perform these tasks, see the installation runbook. The runbook is part of the product documentation and a copy is available online by using your customer credentials to log in to the IBM support site, found at:

<https://www.ibm.com/support/fixcentral/>

In the window that opens, choose the following options:

- ▶ Product Group: Platform Computing
- ▶ Product: Platform Cluster Manager
- ▶ Version: 4.2.1
- ▶ Platform: Linux 64-bit pSeries

Then, in the search area, enter IDEHS. Click **IDEHS**, and the login window opens.

### 4.2.1 Identifying the networks

Complete the following steps:

1. Before you deploy the operating system (OS) on the node, identify the xCAT networks that are required for the deployment. To find the networks, run the **lsdef** command on the service management node (smn), as shown in Example 4-1.

*Example 4-1 List the xCAT service network definition*

```
[root@smn ~]# lsdef -t network service
Object name: service
dynamicrange=50.2.0.50-50.2.0.100
gateway=<xcatmaster>
mask=255.0.0.0
mgtifname=enP1p12s0f3
net=50.0.0.0
```

```
staticrange=50.2.0.10-50.2.0.100
staticrangeincrement=1
tftpserver=50.2.0.234
```

---

The service network is used by PCM and xCAT to control the physical servers through the Baseboard Management Controller (BMC) port. No operations can be done on a host before it is known to the xCAT at the BMC level.

**Note:** The physical server is also referred as the Central Electronics Complex (CEC).

2. To identify the defined data networks, run the **lsdef** command as shown in Example 4-2.

*Example 4-2 List the xCAT data network definition*

---

```
[root@smn ~]# lsdef -t network data
Object name: data
gateway=<xcatmaster>
mask=255.255.255.0
mgtifname=bond0
net=172.16.12.0
staticrange=172.16.12.50-172.16.12.100
staticrangeincrement=1
tftpserver=172.16.12.234
```

---

3. You must identify the provision network by running the **lsdef** command, as shown in Example 4-3.

*Example 4-3 List the xCAT provision network definition*

---

```
[root@smn ~]# lsdef -t network provision
Object name: provision
domain=ibm.com
dynamicrange=10.2.0.150-10.2.0.200
gateway=<xcatmaster>
mask=255.0.0.0
mgtifname=enP1p12s0f2
net=10.0.0.0
staticrange=10.2.0.50-10.2.0.100
staticrangeincrement=1
tftpserver=10.2.0.234
```

---

## 4.2.2 Defining the Central Electronics Complex group

With the BMC cabling discovered, an IP address is assigned from the range of the service network to the new hardware. However, you must identify the host on that network. There are multiple ways to identify the new physical host, such as by serial number or if you know the rest of the physical hosts' IP addresses.

Complete the following steps:

1. To list all IP addresses in use on the service and BMC networks, run the **bmcdiscover** command on the smn, as shown in Example 4-4.

*Example 4-4 Run bmcdiscover to list all the physical hosts*

---

```
[root@smn ~]# bmcdiscover -s nmap --range 50.2.0.50-100 -z -w
node-8348-21c-1038b3a:
objtype=node
groups=all
bmc=50.2.0.65
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038B3A

node-8348-21c-1038afa:
objtype=node
groups=all
bmc=50.2.0.66
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038AFA

node-8348-21c-1038aaa:
objtype=node
groups=all
bmc=50.2.0.67
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038AAA

node-8348-21c-1038aea:
objtype=node
groups=all
bmc=50.2.0.69
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038AEA

node-8348-21c-1038b2a:
objtype=node
groups=all
bmc=50.2.0.74
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038B2A

node-8348-21c-1038b5a:
objtype=node
groups=all
bmc=50.2.0.75
```

```
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038B5A
```

```
node-8348-21c-1038ada:
objtype=node
groups=all
bmc=50.2.0.76
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038ADA
```

```
node-8348-21c-1038b0a:
objtype=node
groups=all
bmc=50.2.0.77
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038B0A
```

```
node-8348-21c-1038b8a:
objtype=node
groups=all
bmc=50.2.0.78
cons=ipmi
mgt=ipmi
mtm=8348-21C
serial=1038B8A
```

- 
2. Because you know that the serial number of the server dn04 is 1038B0A, the node is node-8348-21c-1038b0a. Add that physical host to the cec xCAT group by running the **chdef** command from the smn, as shown in Example 4-5.

*Example 4-5 Add a physical host to the cec group*

---

```
[root@smn ~]# chdef node-8348-21c-1038b0a -p groups=cec
1 object definitions have been created or modified.
```

---

3. After the command runs, check that the physical host is successfully added to the cec group by running the **ldef** command from the smn node, as shown in Example 4-6.

*Example 4-6 List the cec group*

---

```
[root@smn ~]# ldef -t group cec
Object name: cec
```

```
members=node-8348-21c-1038aea,node-8348-21c-1038ada,node-8348-21c-1038aaa,node-8348-21c-1038b0a,node-8348-21c-1038b5a,node-8348-21c-1038afa,node-8348-21c-1038b2a,node-8348-21c-1038b8a,node-8348-21c-1038b3a
```

---

The information that is shown in Example 4-6 confirms that the host is part of the cec xCAT group.

**Note:** If you do not see all the servers in the cec group, contact IBM Support.

### 4.2.3 Updating the server firmware

The minimum firmware (FW) requirement for this solution is OP8 v1.7 1.17.1. Because there is BMC connectivity, perform the FW update by using the BMC GUI.

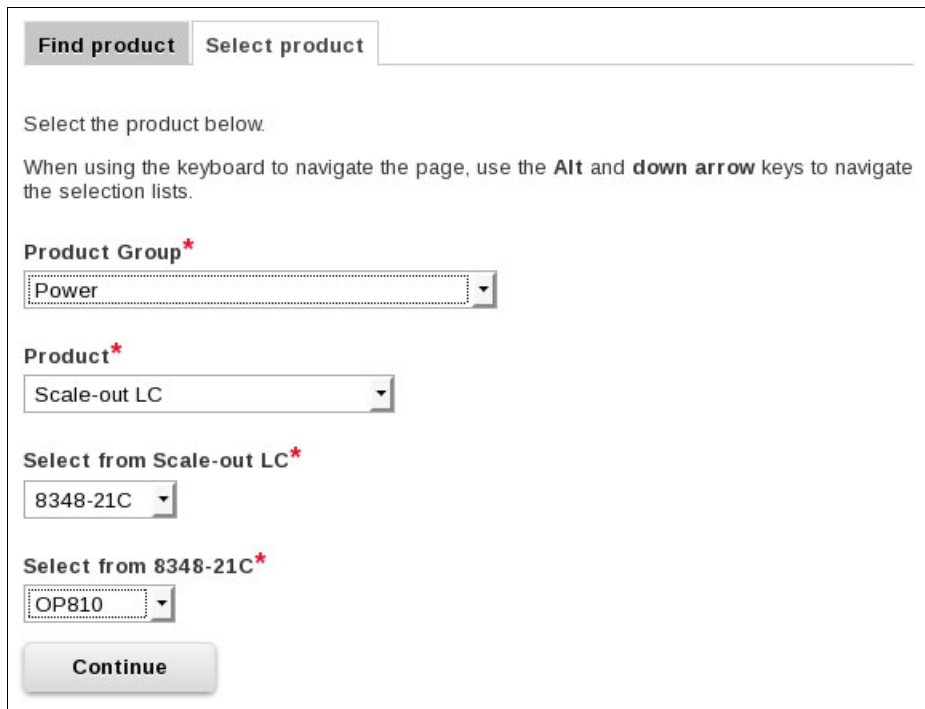
**Note:** xCAT has tools to perform FW updates as well. However, in our tests, the fastest and easiest way was to perform the update was by using the GUI.

Complete the following steps:

1. Obtain the FW level from IBM Fix Central, found at:

<https://www.ibm.com/support/fixcentral/>

2. After accessing Fix Central, select the options that are shown in Figure 4-1.



The screenshot shows a web interface for selecting a product. At the top, there are two tabs: "Find product" (selected) and "Select product". Below the tabs, there is a search bar. The main content area contains the following elements:

- Text: "Select the product below."
- Text: "When using the keyboard to navigate the page, use the **Alt** and **down arrow** keys to navigate the selection lists."
- Section: "Product Group\*" with a dropdown menu showing "Power".
- Section: "Product\*" with a dropdown menu showing "Scale-out LC".
- Section: "Select from Scale-out LC\*" with a dropdown menu showing "8348-21C".
- Section: "Select from 8348-21C\*" with a dropdown menu showing "OP810".
- A "Continue" button at the bottom.

Figure 4-1 Firmware selection menu for 8348-21C

At the time of writing, the list of available FW levels are the ones that are shown in Figure 4-2 on page 51.



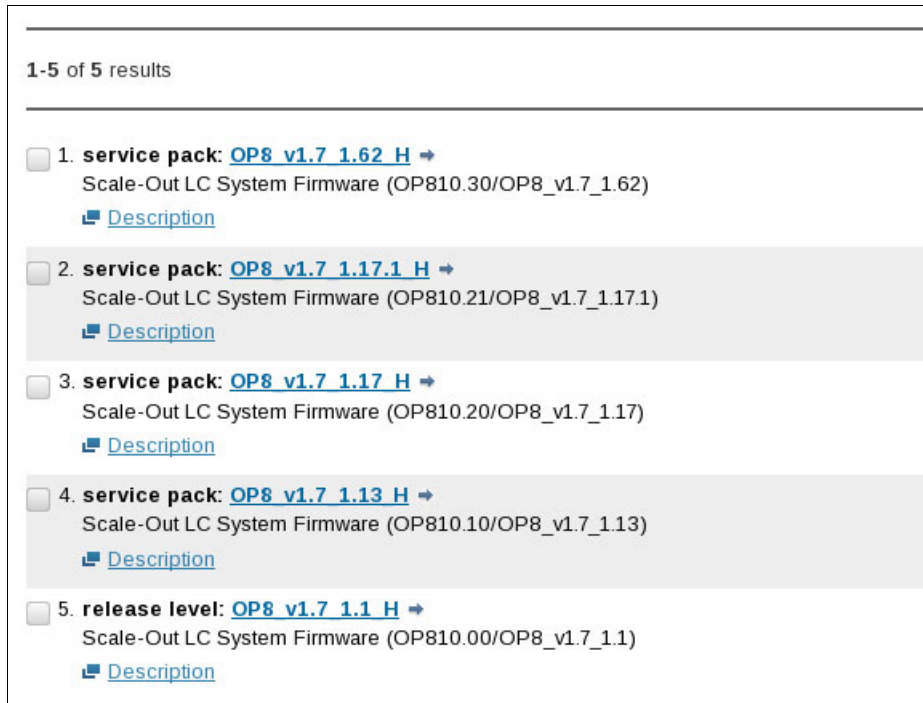


Figure 4-2 List of available FW for 8348-21C

3. For this exercise, update the FW to the latest available level OP8 v1.17 1.52 by selecting the FW and choosing the download method. When you are ready to download, you see the window that is shown in Figure 4-3.



Figure 4-3 OP8 v1.17 1.52 files

The HTML file contains the change log that must be read before you perform the upgrade. The HPM file must be placed on a computer that has access to the BMC network and an Internet browser.

4. Log in to the computer that has the HPM file, and go to the Firmware Update tab at the top. A window similar to Figure 4-4 opens.

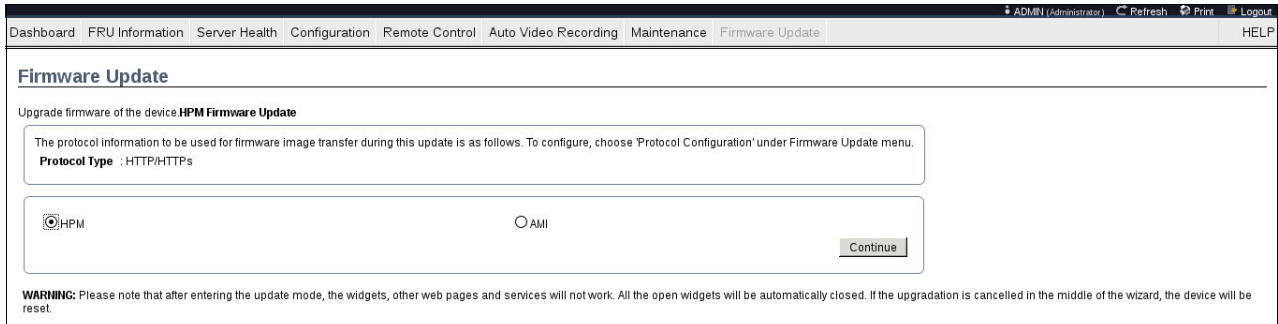


Figure 4-4 Firmware Update tab view on the BMC GUI

5. Select the **HPM** radio button and click **Continue**. A window opens that shows the HPM file that you downloaded from IBM Fix Central. Select the file and click **OK**. The window that is shown in Figure 4-5 opens.

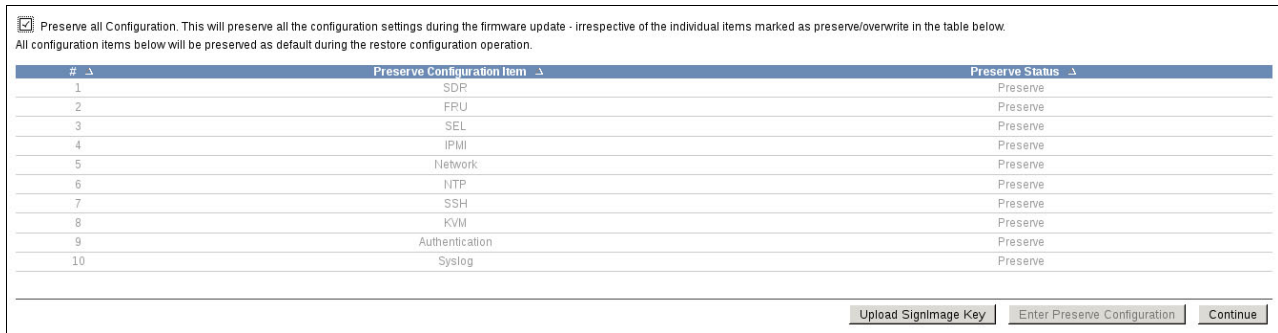


Figure 4-5 Firmware Update components view

6. Select the **Preserve all Configuration** check box and click **Continue**.
7. A window with a warning that you cannot do other operations until the FW upgrade is completed opens. Click **OK** to continue.
8. Figure 4-6 on page 53 shows the last window where you can halt the FW update. After you check that it is safe to continue and that the list of components meets your requirements, click **Proceed**. This action initiates the BIOS and then the BOOT and APP updates. The progress view looks similar to the one that is shown in Figure 4-7 on page 53.

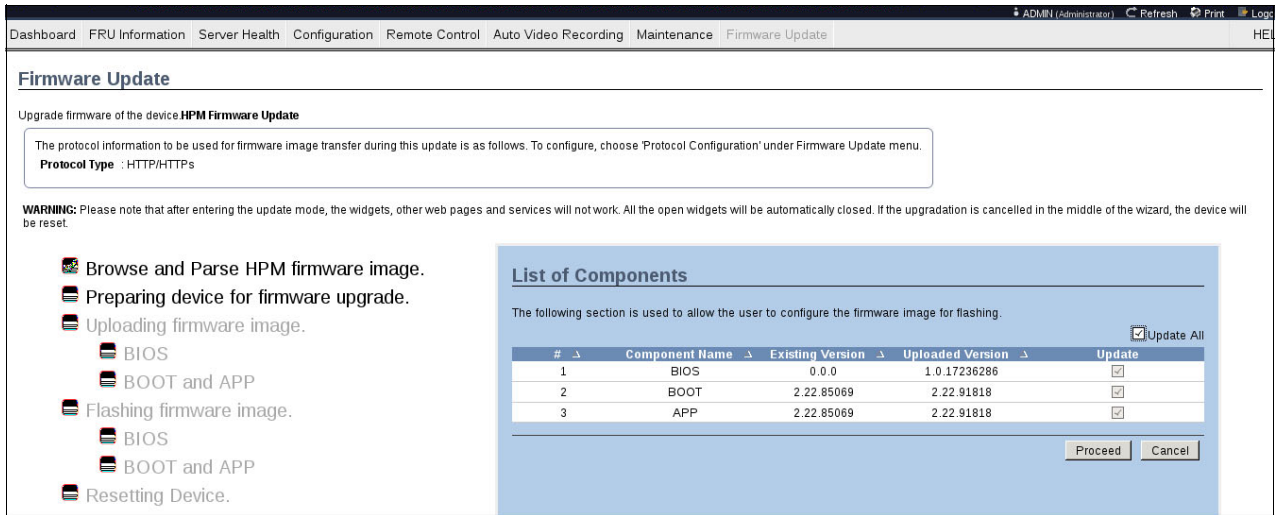


Figure 4-6 Firmware Update start step window

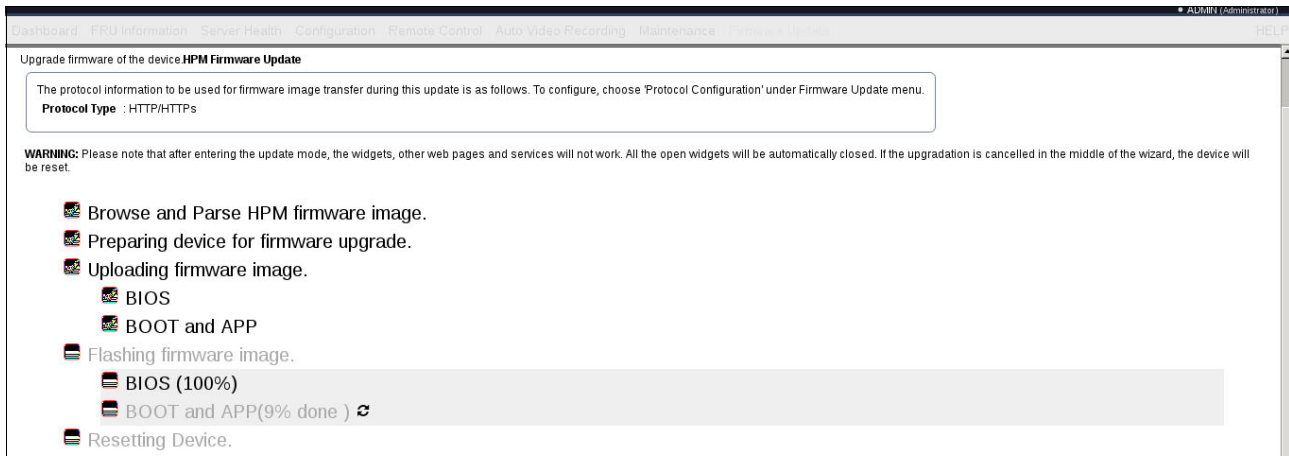


Figure 4-7 Firmware Upgrade progress step

- Wait (do not perform any other operations) until you see the message that is shown in Figure 4-8.

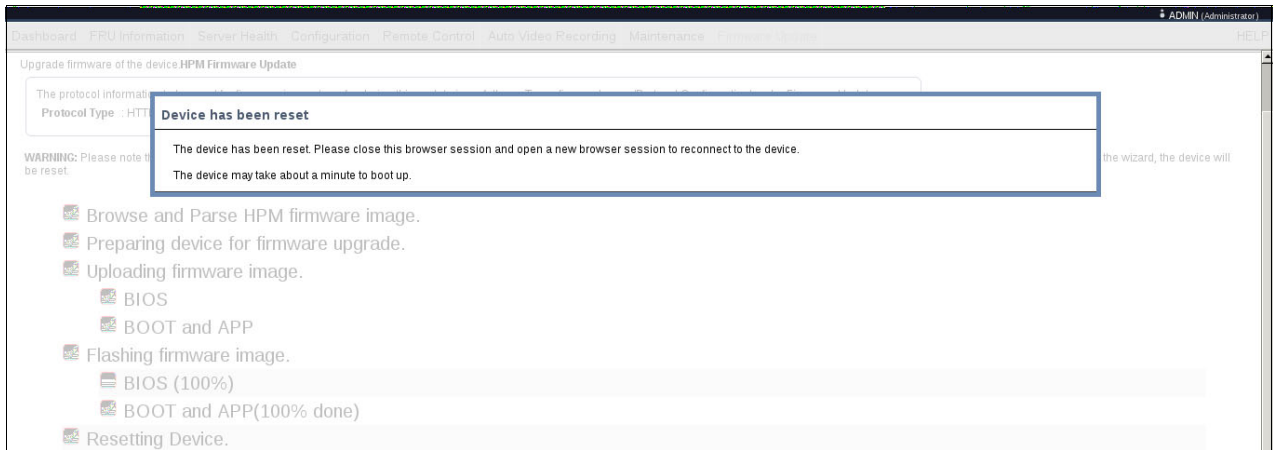


Figure 4-8 Firmware Update success message

10. Now, the server and BMC are restarting. It takes a few minutes for the BMC GUI to be operational again. After it is operational, you can log in and check that the FW level that is shown on the Dashboard tab is correct, as shown in Figure 4-9.

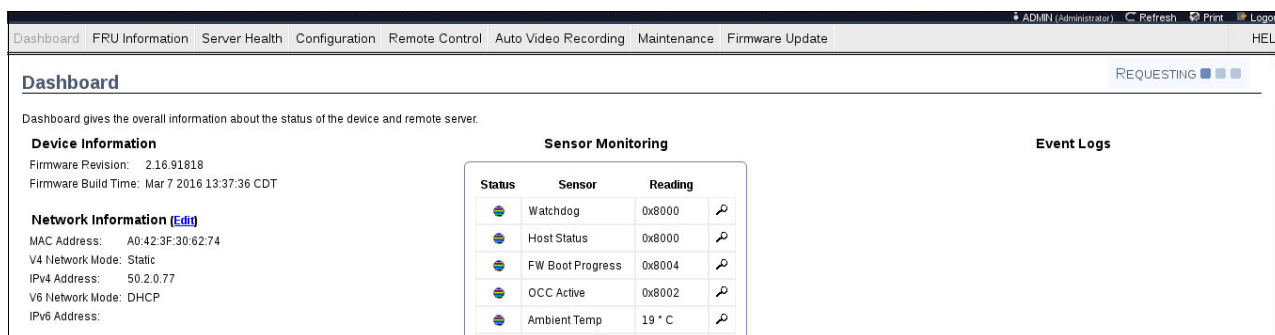


Figure 4-9 Firmware Upgrade level window

Figure 4-9 shows that the current FW level is 2.16.91818.

**Note:** The current FW level can be checked by running either the `cat /proc/ractrends/Helper/FwInfo` or `impitool fru` command.

However, the FW version that is shown is not directly mapped to the version that is shown in IBM Fix Central. The preferred way to identify the installed FW version is to compare the build date with the release date of the FW on IBM Fix Central.

After the FW is at the required level of 1.17.1 or newer, it is possible to install the base OS on the server.

## 4.2.4 Installing the base operating system

Now you can install the base OS by completing the following steps:

1. Configure the xCAT database to be aware of the installation method and the source for the new server.
2. Identify the Media Access Control (MAC) address of the interface that is used by provisioning. To do so, start the system and open a remote console on PetiBoot. From the smn node, run the `rpower` command, as shown in Example 4-7.

*Example 4-7 rpower resetnode for MAC identification*

```
[root@smn ~]# rpower dn04 reset
dn04: reset
```

3. From the same smn node, open an interactive serial console to the dn04 server, as shown in Example 4-8.

*Example 4-8 Open rcons for MAC identification*

```
[root@smn ~]# rcons dn04
```

An interactive serial console opens as though a serial cable is directly connected to the node. Then, the PetitBoot menu opens, as shown in Example 4-9.

*Example 4-9 PetitBoot menu*

---

```
Petitboot (dev.20160310)                8348-21C                1038B0A
.....
  [Disk: sdm2 / 50ae0ef3-78f5-4891-80e0-bea151d0bd39]
Red Hat Enterprise Linux Server (0-rescue-b02cebb6813e4413ac61eb56667dd6a4)
Red Hat Enterprise Linux Server (3.10.0-327.el7.ppc64le) 7.2 (Maipo)
  [Disk: sdn2 / e0ffd45e-cd30-432a-adac-a809563abe3b]
Red Hat Enterprise Linux Server (0-rescue-b02cebb6813e4413ac61eb56667dd6a4)
Red Hat Enterprise Linux Server (3.10.0-327.el7.ppc64le) 7.2 (Maipo)

*System information
System configuration
Language
Rescan devices
Retrieve config from URL
Exit to shell
```

---

4. Select the System information menu entry and press the Enter key. Then, identify the interface MAC address for the network interface section, as shown in Example 4-10.

*Example 4-10 Network interfaces section of the System information screen of PetitBoot*

---

```
Network interfaces
enP1p12s0f0:
  MAC: 98:be:94:58:36:f8
  link: up

enP1p12s0f1:
  MAC: 98:be:94:58:36:f9
  link: up

enP1p12s0f2:
  MAC: 98:be:94:58:36:fa
  link: up

enP1p12s0f3:
  MAC: 98:be:94:58:36:fb
  link: down
```

---

Because the cabling must be identical, the deployment network is connected to the enP1p12s0f2 interface, which is the upper 1 GbE interface.

5. Add the MAC address to the xCAT database as the preferred MAC address for the new node by running the **chdef** command, as shown in Example 4-11.

*Example 4-11 Define the MAC address for deployment on node dn04*

---

```
[root@smn ~]# chdef dn04 mac=98:be:94:58:36:fa
1 object definitions have been created or modified.
```

---

6. You now create two stanza files with the information about the new node. These files are the `dn04.imp` file, where the network information is defined, and `dn04_location.stz`, where the physical location information is defined. The contents of these files are shown in Example 4-12. In this example, select the IP addresses from the networks that are identified in 4.2.1, “Identifying the networks” on page 46.

**Note:** In this example, we manually add the IP address for the provisioning and data networks for this node. You can decide whether xCAT assigns an IP address for those networks instead.

*Example 4-12 Update the dn04 node definitions*

---

```
[root@smn ~]# cat dn04.imp
dn04:
ip=10.2.0.69
mac=98:be:94:58:36:fa
nicips=bmc!50.2.0.77,bond0!172.16.12.69

[root@smn ~]# cat dn04_location.stz
node-8348-21c-1038b0a:
objtype=node
height=2
mtm=8348-21C
rack=rack1
serial=1038B0A
unit=27
```

---

7. Import this data into the xCAT database by running the `nodeimport` command on the `smn` node, as shown in Example 4-13.

*Example 4-13 nodeimport dn04 into xCAT*

---

```
[root@smn ~]# nodeimport dn04.imp
[root@smn ~]#
```

---

8. Confirm the IP addresses on the node that is defined in the xCAT database by running the `lsdef` command, as shown in Example 4-14.

*Example 4-14 List the dn04 xCAT defined IP addresses*

---

```
[root@smn ~]# lsdef -t node dn04 | grep ips
nicips.bmc=50.2.0.77
nicips.bond0=172.16.12.71
nicips.eth90=10.2.0.91
```

---

9. xCAT generates the Domain Name Service (DNS) entries for the provision network by using the `/etc/hosts` file. You must add the entry on to the `smn` node. If the DNS is not generated, the deployment of the base system fails. The entry is shown in Example 4-15.

*Example 4-15 /etc/hosts entry for dn04 on provision network*

---

```
10.2.0.91 dn04.ibm.com dn04 dn04-eth90
```

---

10. After the entry is created in the `/etc/hosts` file, update the DNS by running the `makedns` command, as shown in Example 4-16 on page 57.

*Example 4-16 Update the DNS entries*

---

```
[root@smn ~]# makedns -n
```

---

11. Import the location information in to the xCAT database by running the **chdef** command, as shown in Example 4-17.

*Example 4-17 Import location information for the dn04 into xCAT database*

---

```
[root@smn ~]# cat dn04_location.stz | chdef -z
1 object definitions have been created or modified.
```

---

12. For PCM to discover the changes in the xCAT database, restart PCM by completing the following steps:

- a. Stop PCM.
- b. Check that PCM is down.
- c. Start PCM.
- d. Check that PCM is up.

These steps are shown in Example 4-18.

*Example 4-18 Restart PCM*

---

```
#STOP
[root@smn ~]# cd /etc/rc.d/init.d; ./pcm stop; cd - >/dev/null
Stopping Web Portal services           [ OK ]
Stopping PERF services                 [ OK ]
Stopping Rule Engine service          [ OK ]
Stopping PCMD service                 [ OK ]
Stopping Message broker               [ OK ]
Shut down LIM on <smn.ibm.com> ..... done           [ OK ]
Stopping Platform Cluster Manager Services: [ OK ]

#CHECK is DOWN
[root@smn ~]# cd /etc/rc.d/init.d; ./pcm status; cd - >/dev/null
EGO service is not running

#START
[root@smn ~]# cd /etc/rc.d/init.d; ./pcm start; cd - >/dev/null
Checking for xcatd service started     [ OK ]
Start up LIM on <smn.ibm.com> ..... done           [ OK ]
- Waiting for PCM EGO service started ... [ OK ]
Cluster name : PCM EGO master host name : smn.ibm.com EGO master version : 1.2.10
- Waiting for PCM master node online ..... [ OK ]
Starting PERF services                 [ OK ]
Starting Message broker               [ OK ]
Starting PCMD service                 [ OK ]
Starting Rule Engine service          [ OK ]
Starting Web Portal services          [ OK ]
Starting Platform Cluster Manager Services: [ OK ]

#CHECK is UP
[root@smn ~]# cd /etc/rc.d/init.d; ./pcm status; cd - >/dev/null
Cluster name           : PCM
EGO master host name   : smn.ibm.com
EGO master version     : 1.2.10
```

SERVICE	STATE	ALLOC	CONSUMER	RGROUP	RESOURCE	SLOTS	SEQ_NO	INST_STATE	ACTI
PURGER	STARTED	46	/Manage*	Manag*	smn.ibm*	1	1	RUN	56
PTC	STARTED	47	/Manage*	Manag*	smn.ibm*	1	1	RUN	57
PLC	STARTED	48	/Manage*	Manag*	smn.ibm*	1	1	RUN	58
WEBGUI	STARTED	52	/Manage*	Manag*	smn.ibm*	1	1	RUN	62
RULE-EN*	STARTED	51	/Manage*	Manag*	smn.ibm*	1	1	RUN	61
PCMD	STARTED	50	/Manage*	Manag*	smn.ibm*	1	1	RUN	60
ACTIVEMQ	STARTED	49	/Manage*	Manag*	smn.ibm*	1	1	RUN	59

**Note:** You can now check the output of the `lstrree` command to confirm that you see the new server and the physical host that is assigned to it is the correct one.

13. Define the deployment interface by running the `chdef` command, as shown in Example 4-19.

*Example 4-19 Run the `chdef` command define the deployment interface for `dn04`*

```
[root@smn ~]# chdef dn04 installnic=mac
1 object definitions have been created or modified.
```

14. Because this is an already configured IBM Data Engine for Hadoop and Spark clustered solution, it already has defined install resources. To see the available resources, run the `lsdef` command, as shown in Example 4-20.

*Example 4-20 List the installation `osimages` available on `xCAT`*

```
[root@smn ~]# lsdef -t osimage
rhels7.2-ppc64le-stateful-compute_ibm (osimage)
rhels7.2-ppc64le-stateful-mgmtnode (osimage)
rhels7.2-ppc64le-stateless-compute (osimage)
```

15. You set the image to install on node `dn04` on the next start by running the `nodeset` command, as shown in Example 4-21.

*Example 4-21 Set `osimage` on `dn04` for the next start*

```
[root@smn ~]# nodeset dn04 osimage=rhels7.2-ppc64le-stateful-compute_ibm
```

16. Set the boot mode to Network in node `dn04` on the next start by running the `rsetboot` command, as shown in Example 4-22.

*Example 4-22 Set the `dn04` boot mode to network*

```
[root@smn ~]# rsetboot dn04 net
dn04: Network
```

17. You are ready to deploy the base OS in an automated way. The next time that the server starts, it contacts the `smn` node and installs the base OS without any user interaction. The installation configures the network, and does some post-configuration steps that are needed for this setup. To kick off the restart, run the `rpower` command, as shown in Example 4-23.

*Example 4-23 Restart node `dn04`*

```
[root@smn ~]# rpower dn04 reset
dn04: reset
```



To monitor the installation, you can either open an interactive serial console by running the `rcons` command on dn04 or use the non-interactive `tail -f /var/log/console/dn04` command. You can also wait until the server starts and responds to `ping` commands.

When the installation is complete, you can `ssh` from the snm node to the dn04 node as the root user. The SSH keys are already exchanged between the nodes, so no password is asked.

## 4.2.5 Configuring the host name, users, and groups

After the base OS installation is complete, install and configure the middleware in the cluster to manage and run workloads on the new data node by completing the following steps:

1. Change the host name so that it uses the name that belongs to the data network by running the `change_hostname.sh` script, as shown in Example 4-24.

**Note:** For this example, we have the IBM Data Engine for Hadoop and Spark packages in the `/root/IDEHS_Inst_1.0/packages/idehsv1` directory. You must adapt the path to your installation.

*Example 4-24 Change the host name of dn04 to dn04-dat*

```
[root@smn idehsv1]# /root/IDEHS_Inst_1.0/packages/idehsv1/change_hostname.sh -m dn04 -d dn04-dat
```

```
#####Change Hostnames to data network based Hostname#####
```

```
##### hostname src: dn04
```

```
##### hostname dst: dn04-dat
```

```
[D]: new static hostname is dn04-dat
```

```
[I]: Apply PCM EGO patch.
```

```
xCAT: [I]: Wrap "KIT_PCM_setupego " and run as "TMP_wIPmU"
```

```
Tue May 24 09:01:48 EDT 2016 Running postscript: KIT_PCM_setupego
```

```
Postscript: KIT_PCM_setupego exited with code 0
```

```
dn04: dn04-dat
```

2. Manage the users and groups for the node just added.

- If your setup uses the Lightweight Directory Access Protocol (LDAP), configure the LDAP client now by following whichever process you already use.

**Note:** The reason for configuring LDAP at the beginning is that the current version of Ambari defines the users and groups with different UIDs and GIDs than the rest of the cluster, which makes the node unusable as the shared file system. The applications expect all nodes to have the same UID and GID.

- If your setup does not use LDAP, copy the `/etc/passwd` and `/etc/groups` files from another compute node to the new node to ensure that you have the same UID and GID across nodes.

## 4.2.6 Installing and configuring IBM Spectrum Scale

After the UID and GID are set by either configuring the LDAP client or copying the files from another working data node, you can install and update the IBM Spectrum Scale client in the new node by using the `inst_gpfs` script that is shown in Example 4-25.

*Example 4-25 Install the IBM Spectrum Scale client on dn04*

---

```
[root@smn idehsv1]# /root/IDEHS_Inst_1.0/packages/idehsv1/inst_gpfs
```

---

The installation takes a few minutes and it generates a large amount of output. You must wait until the installation process indicates that it reached a successful installation and update.

After the installation completes, the process has installed and updated IBM Spectrum Scale to the same version as the rest of the nodes in the cluster. Now, you must configure and add storage to the new node that is added to the IBM Spectrum Scale-File Place Optimizer (IBM Spectrum Scale-FPO) cluster.

**Note:** Although this example offers only the IBM Spectrum Scale-FPO solution, it is possible due to the flexibility of IBM Spectrum Scale to have a colder tier/pool of IBM Spectrum Scale for data not being use for the workload, and move the data transparently by using IBM Spectrum Scale Information Lifecycle Management (ILM) capabilities.

To read more about IBM Spectrum Scale and its ILM capabilities, see *IBM Spectrum Scale (formerly GPFS)*, SG24-8254.

Complete the following steps:

1. Add the new node to the existing IBM Spectrum Scale cluster by running the `mmaddnode` command from a member of the cluster node. This example uses the `dn01` node, as shown in Example 4-26.

*Example 4-26 Add the dn04 node to the IBM Spectrum Scale cluster*

---

```
[root@dn01-dat ~]# mmaddnode -N dn04-dat
Wed May 25 15:29:13 EDT 2016: mmaddnode: Processing node dn04-dat.ibm.com
mmaddnode: Command successfully completed
mmaddnode: Warning: Not all nodes have proper GPFS license designations.
Use the mmchlicense command to designate licenses as needed.
mmaddnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

---

2. Accept the IBM Spectrum Scale-FPO license for the new node by running the `mmchlicense` command. You can run this command from any node that belongs to the cluster that includes the `dn04` node, as shown in Example 4-27.

*Example 4-27 Set the IBM Spectrum Scale-FPO license on dn04 node*

---

```
[root@dn01-dat ~]# mmchlicense fpo --accept -N dn04-dat
The following nodes will be designated as possessing FPO licenses:
dn04-dat.ibm.com
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

---

3. Create the partition disk table on the data disks by using any method that you choose if the result is a partition scheme that is identical to the rest of the data nodes. For convenience, and without any IBM warranty, you can run the script to clone partitions that is shown in Appendix B, “Script to clone partitions” on page 103.
4. After the partition scheme is created, create a stanza file that generates the Network Shared Disk (NSD). Use the same naming convention that is used in the cluster in this version, but you can use any names that makes sense to you. However, it is important to use failure groups that are not in use at the moment in the IBM Spectrum cluster. You can see the `nsd_disks.txt` file that we used for the stanza file in Example 4-28.

*Example 4-28 The nsd\_disks.txt file that is used on the dn04 node*

---

```
%nsd: nsd=gpfs165nsd device=/dev/sd1 servers=dn04-dat usage=metadataOnly failureGroup=404 pool=system
%nsd: nsd=gpfs166nsd device=/dev/sdm servers=dn04-dat usage=metadataOnly failureGroup=404 pool=system
%nsd: nsd=gpfs167nsd device=/dev/sdn servers=dn04-dat usage=metadataOnly failureGroup=404 pool=system
%nsd: nsd=gpfs168nsd device=/dev/sdo servers=dn04-dat usage=metadataOnly failureGroup=404 pool=system
%nsd: nsd=gpfs169nsd device=/dev/sdp2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs170nsd device=/dev/sdq2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs171nsd device=/dev/sdr2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs172nsd device=/dev/sds2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs173nsd device=/dev/sdt2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs174nsd device=/dev/sdu2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs175nsd device=/dev/sdv2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
%nsd: nsd=gpfs176nsd device=/dev/sdw2 servers=dn04-dat usage=dataOnly failureGroup=4,0,4 pool=datapool
```

---

**Note:** In this sample, we choose `gpfsXXXnsd` names for the NSD. The only reason to do so is that the current IBM Data Engine for Hadoop and Spark uses that schema. However, NSD names that start with `gpfs` are reserved and cannot be used because they cause the creation of the NSD to fail. Do not use `gpfsXXXnsd` names. If your cluster has `gpfsXXXnsd` names already, contact IBM Support.

5. Start the IBM Spectrum Scale client on `dn04` by running the `mmstartup` command from any node that is part of the cluster, as shown in Example 4-29.

*Example 4-29 Start IBM Spectrum Scale on the dn04 node*

---

```
[root@dn04-dat ~]# mmstartup -N dn04-dat
```

---

6. Create the NSD by running the `mmcrnsd` command. The command can run from any node that is part of the cluster, and has local access to the stanza file. In this example, we use node `dn04`, as shown in Example 4-30.

*Example 4-30 The mmcrnsd command running on the dn04 node*

---

```
[root@dn04-dat ~]# mmcrnsd -F nsd_disks.txt -v no
mmcrnsd: Processing disk sd1
mmcrnsd: Processing disk sdm
mmcrnsd: Processing disk sdn
mmcrnsd: Processing disk sdo
mmcrnsd: Processing disk sdp2
mmcrnsd: Processing disk sdq2
mmcrnsd: Processing disk sdr2
mmcrnsd: Processing disk sds2
mmcrnsd: Processing disk sdt2
mmcrnsd: Processing disk sdu2
mmcrnsd: Processing disk sdv2
mmcrnsd: Processing disk sdw2
```

---

mmcrnsd: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.

---

**Important:** In this example, we use the “skip verification of the disks” option with the command, as shown in Example 4-30. This can be a dangerous approach because it unconditionally formats the disks. We did so because we had run other tests in those disks. If you are *absolutely* sure that those disks have no data, you can use this approach as well. Otherwise, remove the **-v no** flag of the **mmcrnsd** command.

7. The NSDs are not known to the IBM Spectrum Scale cluster, so you must add them to the bigpfs shared parallel file system, as shown in Example 4-31.

*Example 4-31 Add the NSDs*

---

```
[root@dn04-dat ~]# mmadddisk bigpfs -F nsd_disks.txt
The following disks of bigpfs will be formatted on node mn02-dat:
gpfs165nsd: size 5723166 MB
gpfs166nsd: size 5723166 MB
gpfs167nsd: size 5723166 MB
gpfs168nsd: size 5723166 MB
gpfs169nsd: size 4578533 MB
gpfs170nsd: size 4578533 MB
gpfs171nsd: size 4578533 MB
gpfs172nsd: size 4578533 MB
gpfs173nsd: size 4578533 MB
gpfs174nsd: size 4578533 MB
gpfs175nsd: size 4578533 MB
gpfs176nsd: size 4578533 MB
Extending Allocation Map
Checking Allocation Map for storage pool system
  7 % complete on Wed May 25 15:54:06 2016
 15 % complete on Wed May 25 15:54:11 2016
 32 % complete on Wed May 25 15:54:16 2016
 53 % complete on Wed May 25 15:54:21 2016
 73 % complete on Wed May 25 15:54:26 2016
 94 % complete on Wed May 25 15:54:31 2016
100 % complete on Wed May 25 15:54:32 2016
Checking Allocation Map for storage pool datapool
 86 % complete on Wed May 25 15:54:37 2016
100 % complete on Wed May 25 15:54:38 2016
Completed adding disks to file system bigpfs.
mmadddisk: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

---

8. The disks are part of the shared file system, but the data must be restriped to comply with the locality of the data, the number of copies, and the failure groups. To do so, run the **mmrestripefs** and the **mmapplypolicy** commands, as shown in Example 4-32 on page 63.

**Note:** The commands in Example 4-32 on page 63 must be run also when restarting a node because the IBM Spectrum Scale-FPO characteristics as data is in local drives, not in main storage.

*Example 4-32 The mmrestripefs and mmapplypolicy commands to balance the file system*

```
[root@dn04-dat ~]# mmrestripefs bigpfs -R  
[root@dn04-dat ~]# mmrestripefs bigpfs -b  
[root@dn04-dat ~]# mmapplypolicy bigpfs
```

**Note:** The process that is shown in Example 4-32 can take a fair amount of time depending on the data in the file system. Run it in a window session.

The node has the shared file system up and balanced.

## 4.2.7 Installing software with Ambari

You can use the Ambari GUI to install the software on the dn04 node. To do so, complete the following steps:

1. Use a browser and go the Uniform Resource Locator (URL) of the Ambari management console. You see the Ambari dashboard, as shown in Figure 4-10.

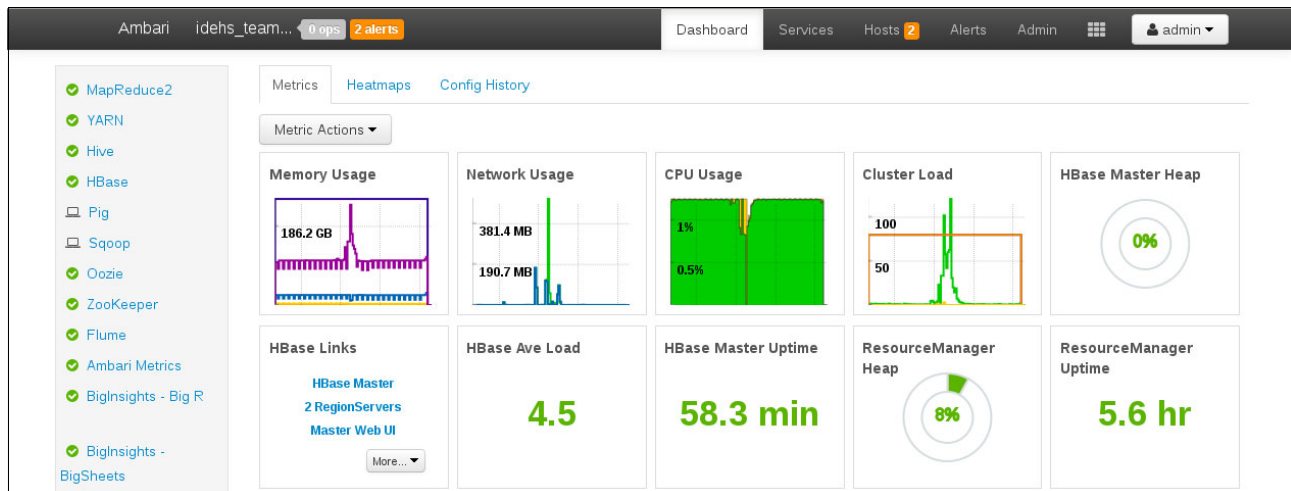


Figure 4-10 Ambari dashboard before you add the dn04 node

2. Click the **Hosts** tab. The window that is shown in Figure 4-11 opens.

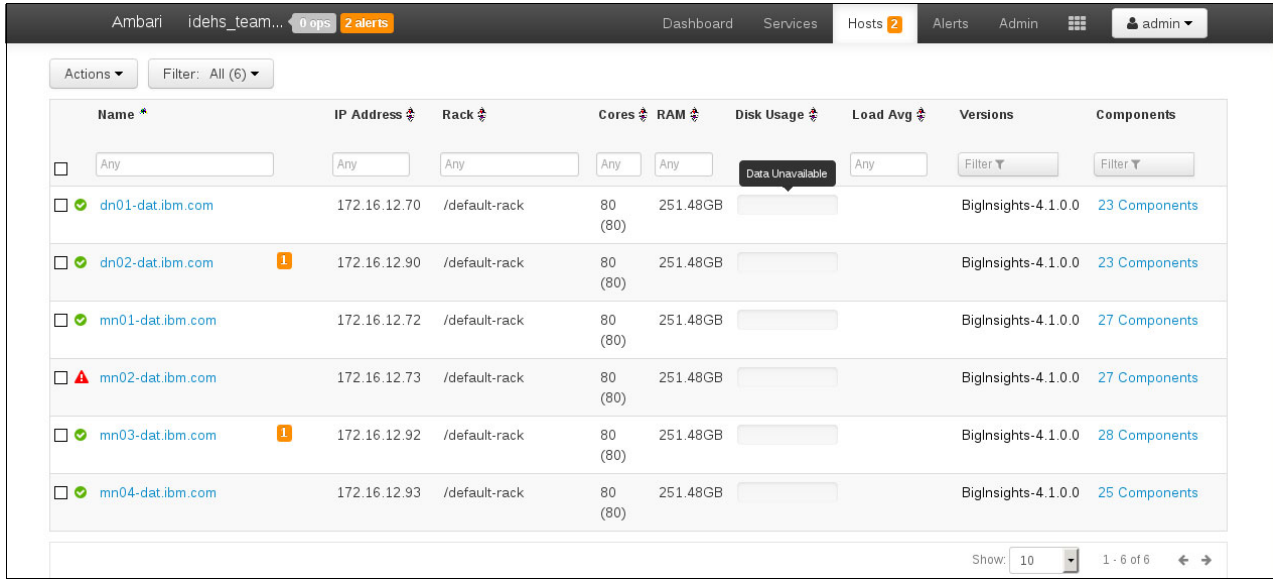


Figure 4-11 Ambari host view before adding dn04

Click **Actions** → **Add host**. The window that is shown in Figure 4-12 opens.

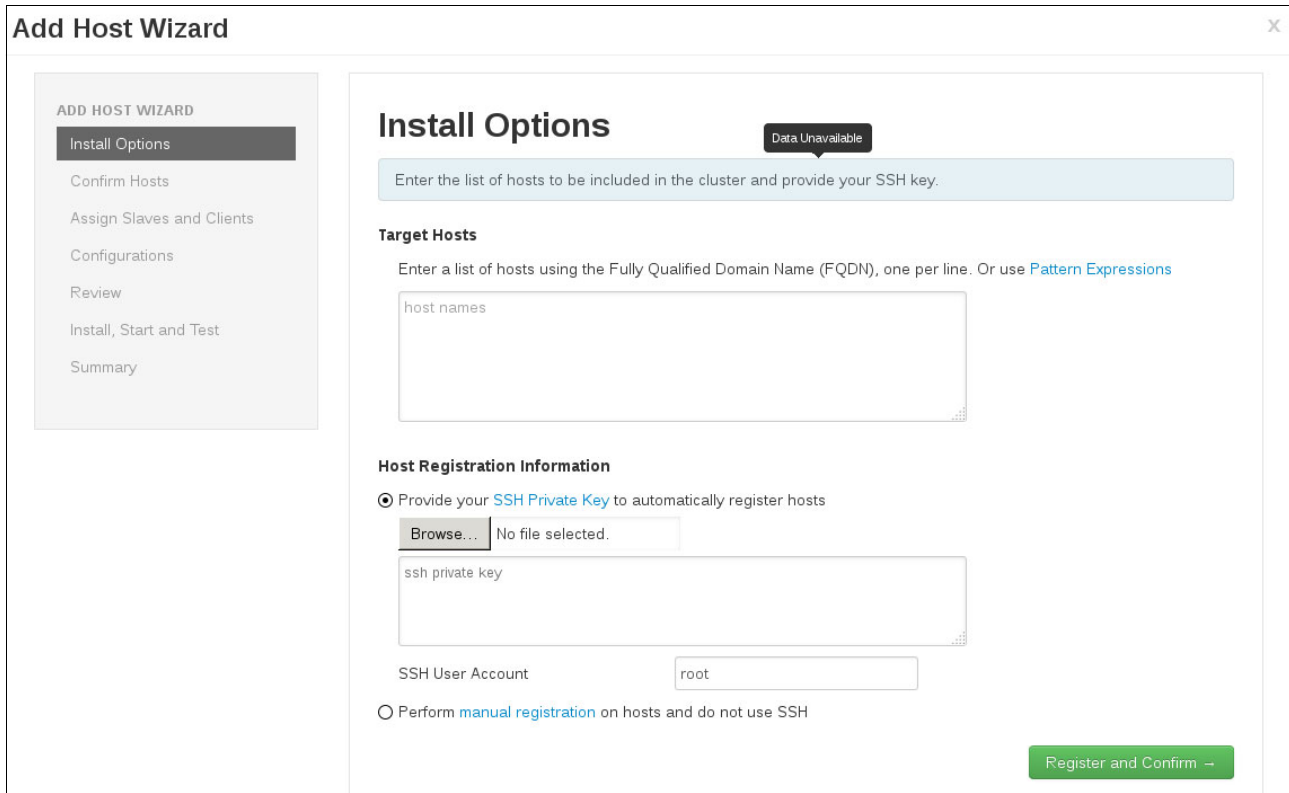


Figure 4-12 Empty Ambari add host window

- For the host name, it is important to add the data network and the Fully Qualified Domain Name (FQDN) host name or the deployment does not work. In this example, the FQDN is the dn04-dat.ibm.com host name. For the root user SSH private key, you can either copy and paste it from the dn04 node or browse for it. If you browse, be sure that you selected to view hidden files because the directory where the keys are is hidden (the .ssh directory on the root user home). The window should now look like the one that is shown in Figure 4-13.

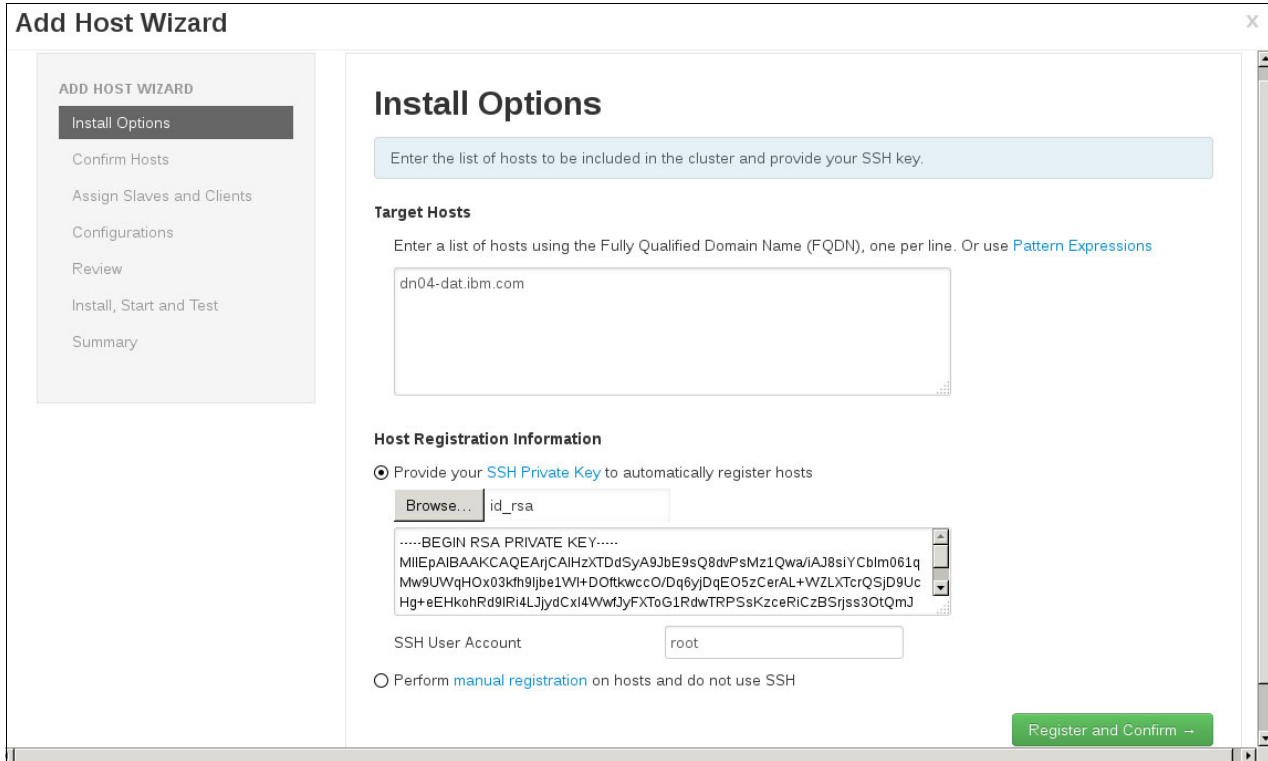


Figure 4-13 Filled Ambari add host window

- Click the **Register and Confirm**, which starts the installation of the Ambari agent. During the installation, you see a window similar to the one that is shown in Figure 4-14.

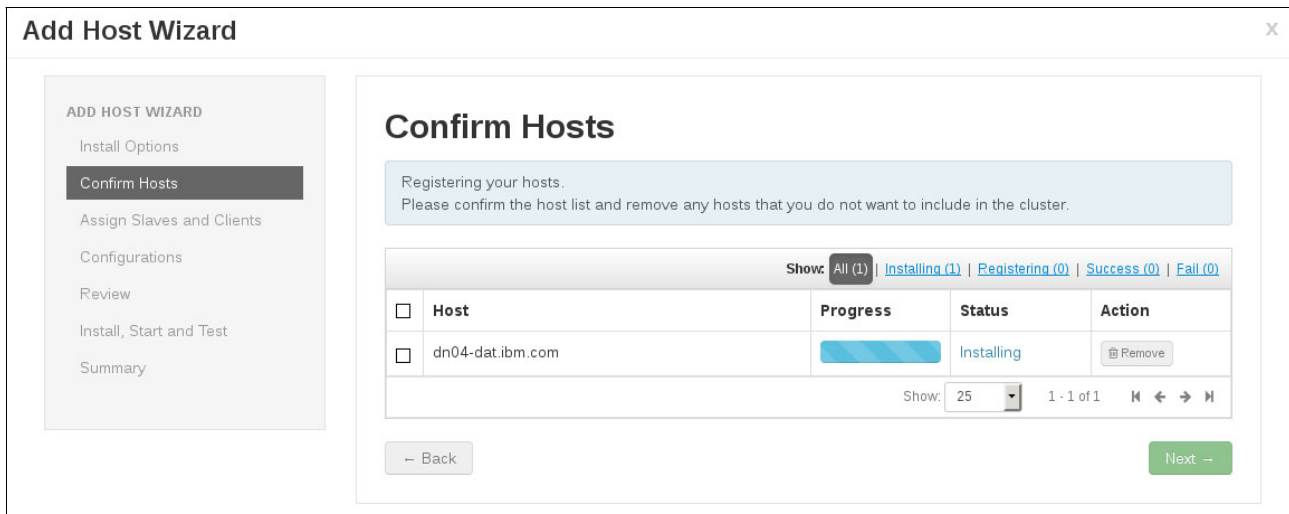


Figure 4-14 Install the Ambari agent

Wait until the installation completes. A successful installation shows a window that is similar to Figure 4-15.

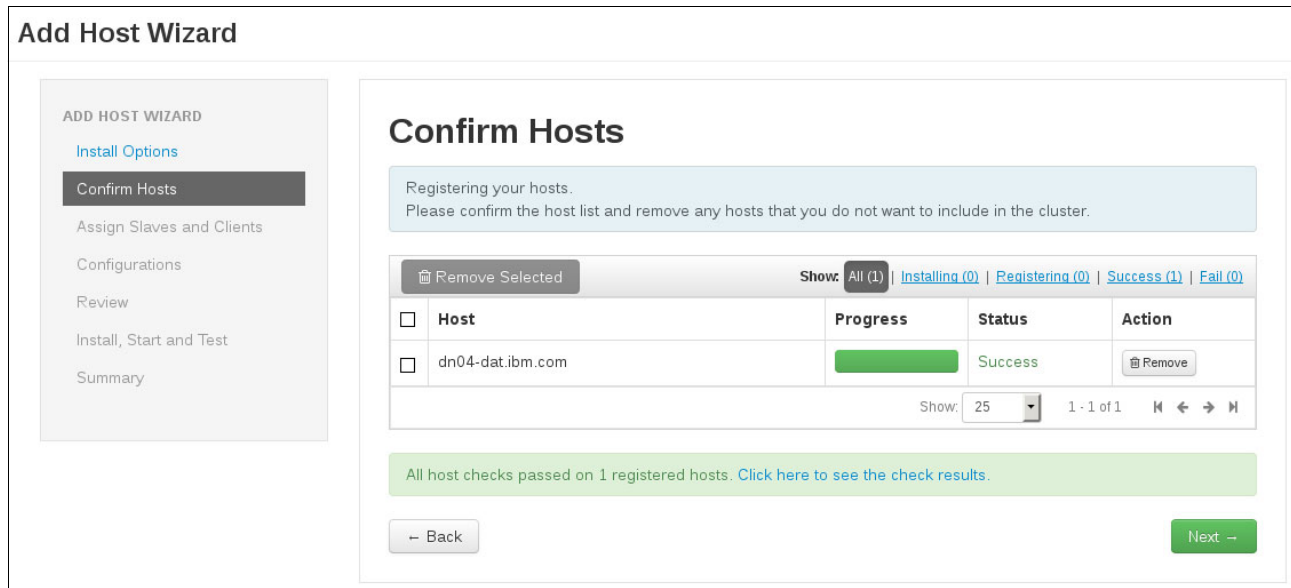


Figure 4-15 Success with installing the Ambari agent

5. Ambari now can control the dn04 node and install and configure software on it. Click **Next** to select the software to install.



6. If your setup uses the default IBM Data Engine for Hadoop and Spark software stack, then you must select the following groups in the window that opens, as shown in Figure 4-16 and in Figure 4-17:
- NodeManager
  - RegionServer
  - Flume
  - GPFS Hadoop Connector
  - Symphony Compute
  - Client

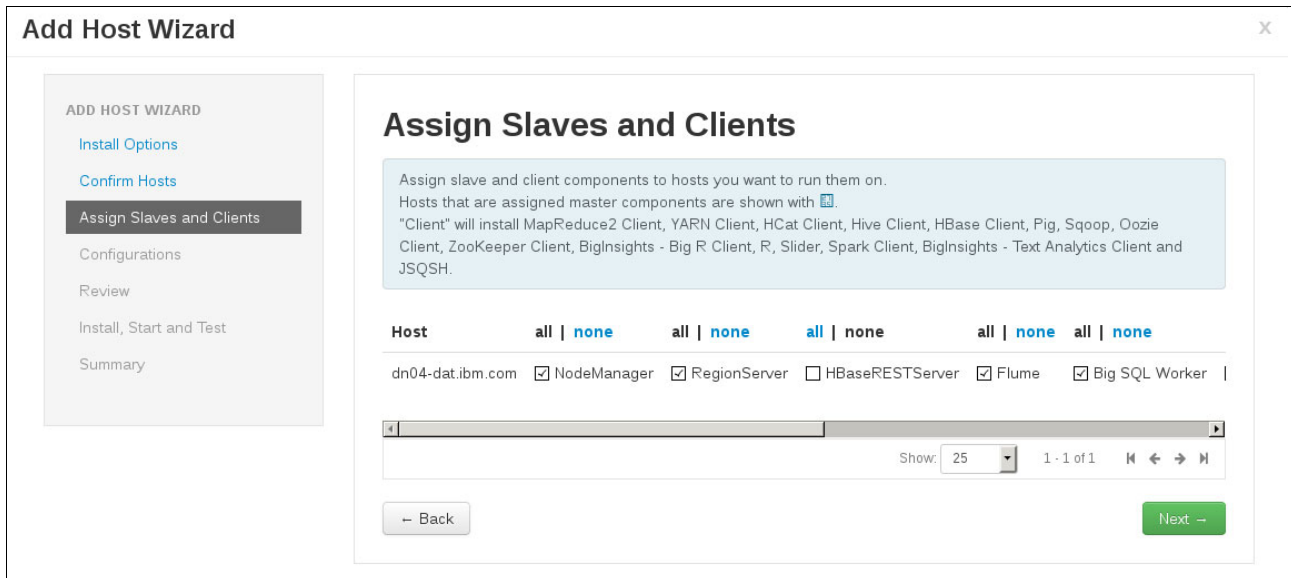


Figure 4-16 Ambari software selection window 1

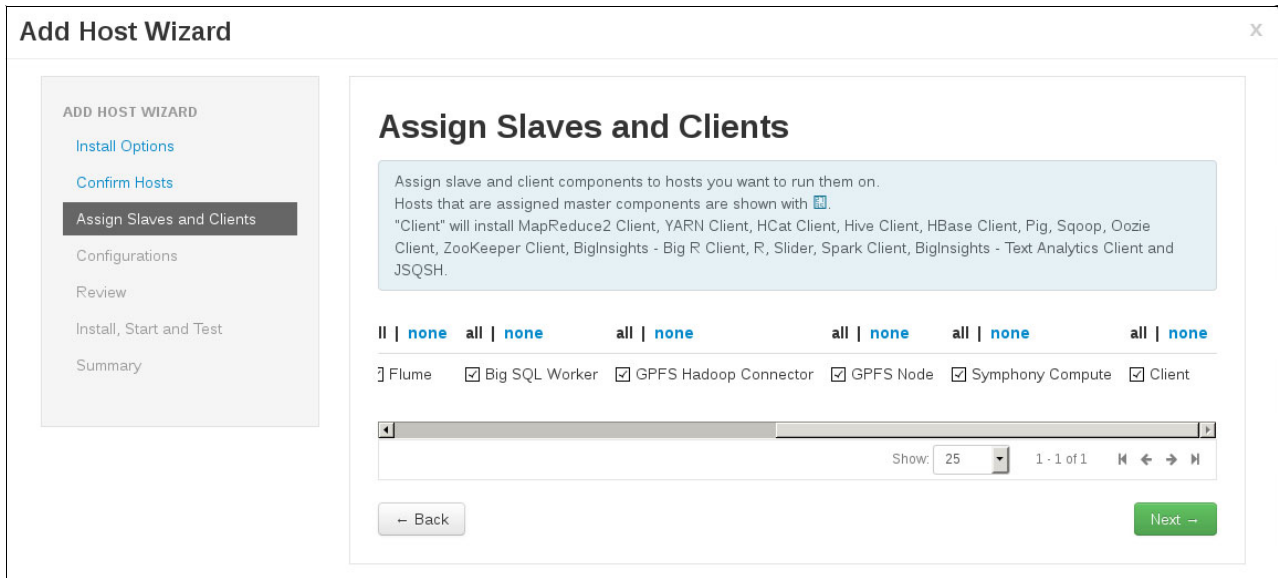


Figure 4-17 Ambari software selection window 2

7. Click **Next**. The Configuration window opens, as shown in Figure 4-18.

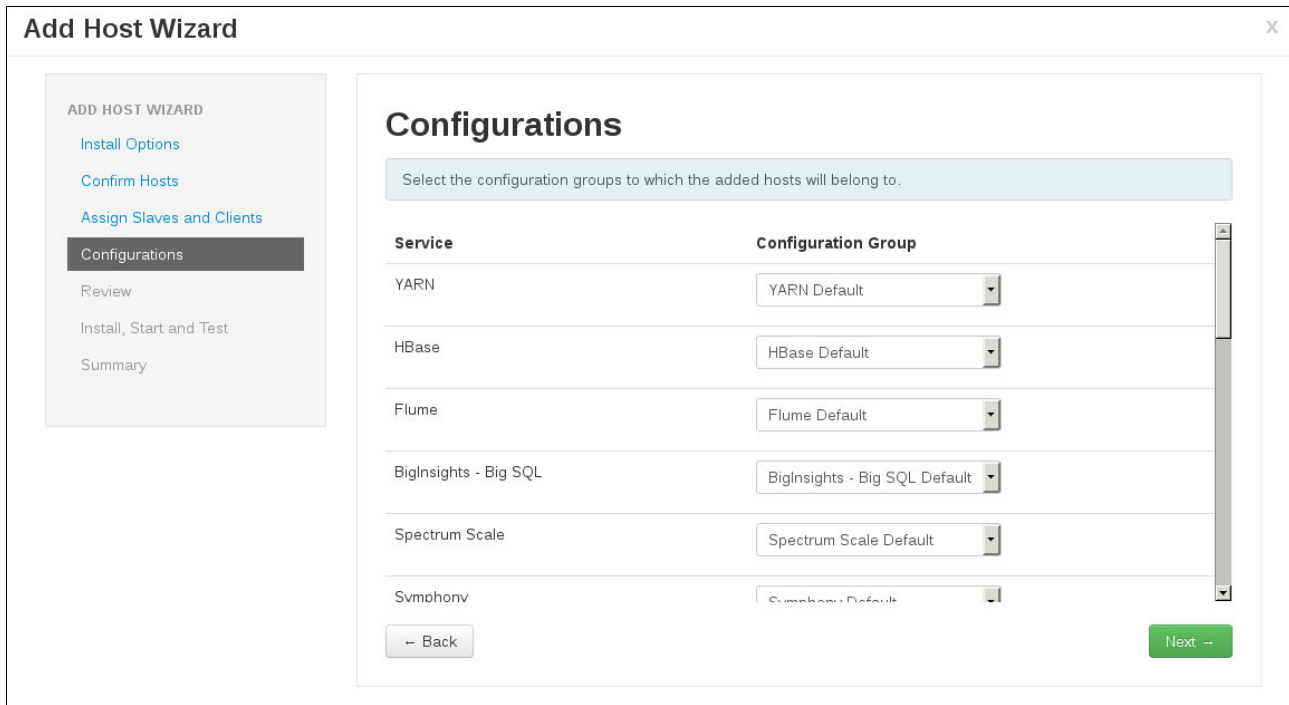


Figure 4-18 Ambari configuration new host window

8. Click **Next**. The Review window opens, as shown in Figure 4-19.

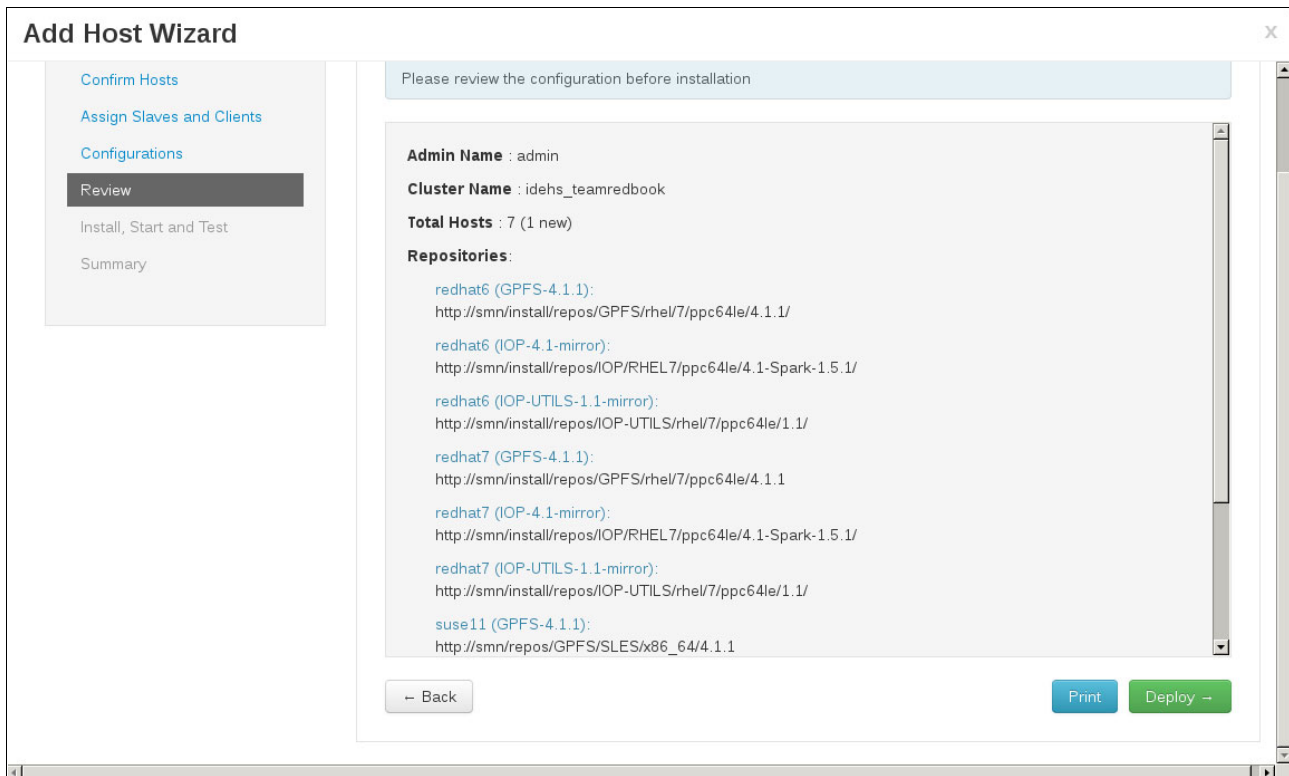


Figure 4-19 Ambari review add host window

9. Click **Deploy**, which initiates the deployment of all the components that are selected during the Ambari add host process. The window that shows the process is shown in Figure 4-20.

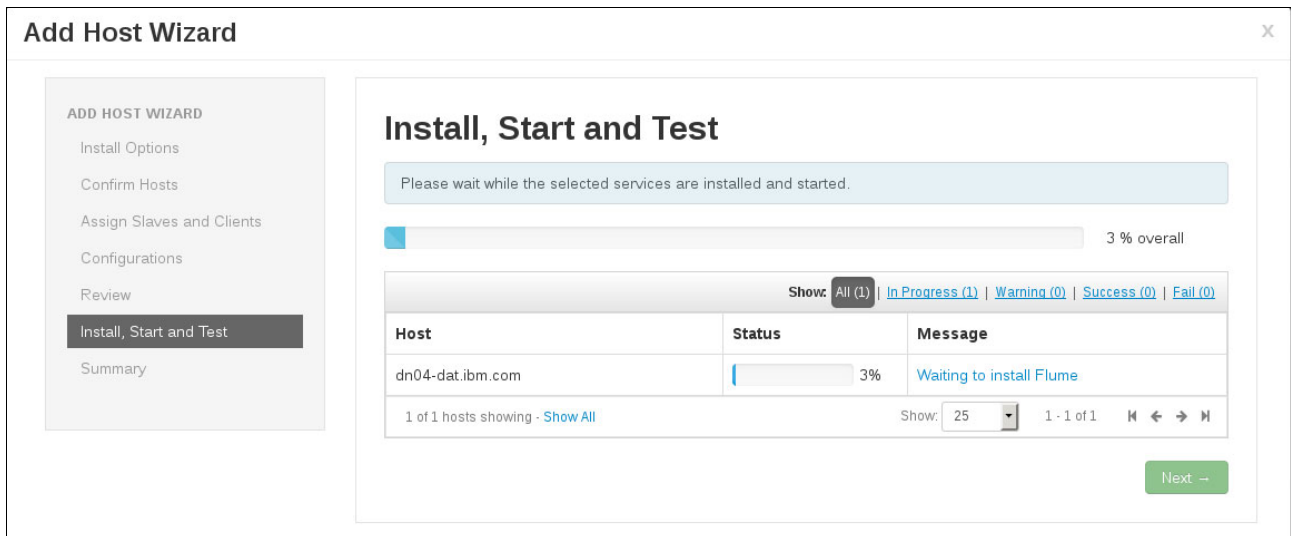


Figure 4-20 Ambari deployment installation window

10. This action continues the deployment and starts the rest of the remaining components, as shown in Figure 4-21 and Figure 4-22.

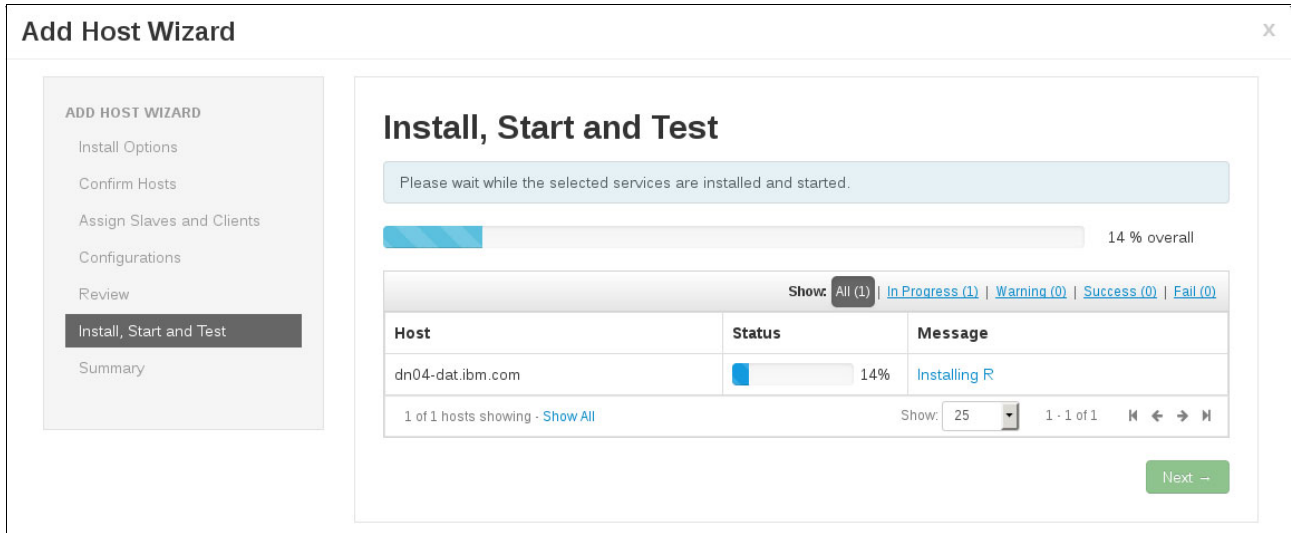


Figure 4-21 Ambari install the remaining components window

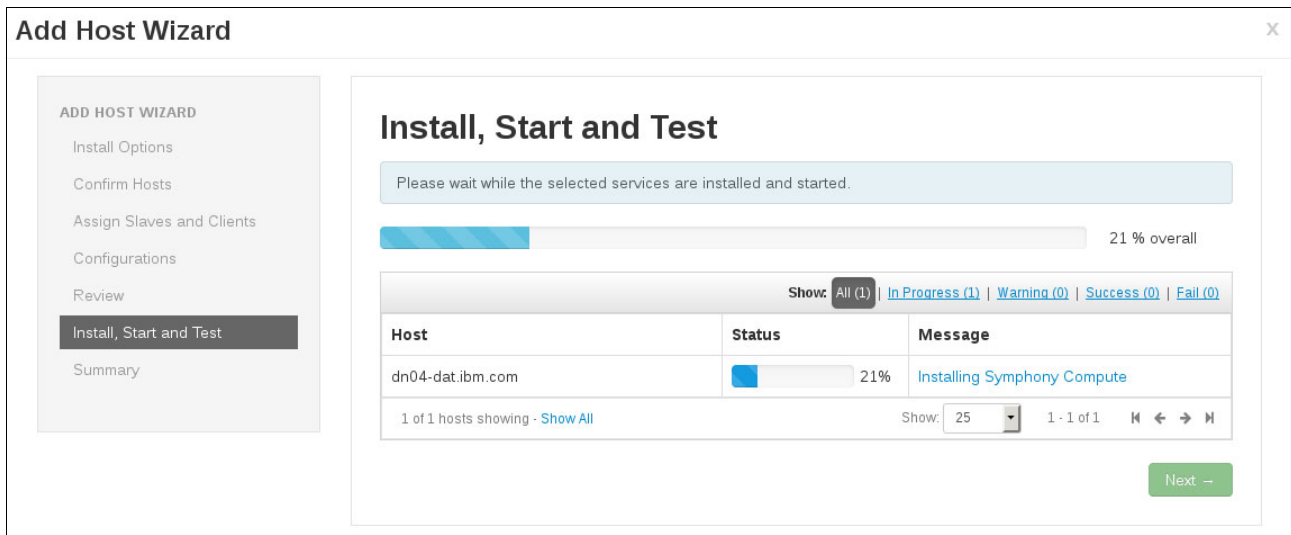


Figure 4-22 Ambari installing Symphony Compute window

11. Click **Next** to move to open the Summary window, as shown in Figure 4-23 on page 71.

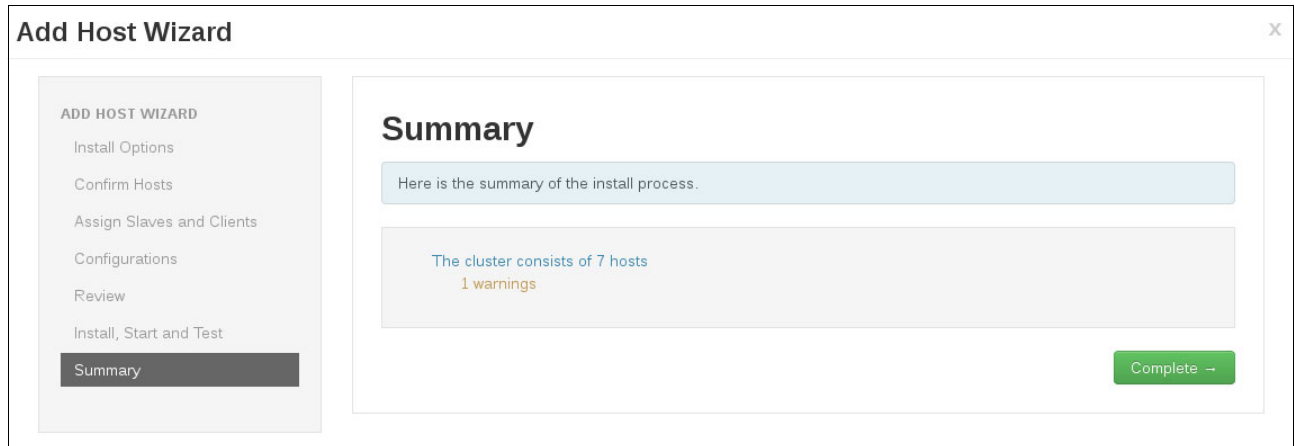


Figure 4-23 Ambari add host summary installation window

This is the last window of the installation of all the required software to run workloads. If you did not see the warning window, your node is ready now to run workloads.

### 4.3 Configuring the Apache Spark UI

Apache Spark can be configured to run on IBM Spectrum Symphony. For the steps to enable Apache Spark on IBM Spectrum Symphony, see the installation runbook. The runbook is part of the product documentation and a copy is available online by using your customer credentials to log in to the IBM support site, found at:

<https://www.ibm.com/support/fixcentral/>

In the window that opens, choose the following options:

- ▶ Product Group: Platform Computing
- ▶ Product: Platform Cluster Manager
- ▶ Version: 4.2.1
- ▶ Platform: Linux 64-bit pSeries

Then, in the search area, enter IDEHS. Click **IDEHS**, and the login window opens.

When Apache Spark is already running on IBM Spectrum Symphony, the Apache Spark service in Ambari shows “Spark is running on Symphony Resource Manager”, as shown in Figure 4-24. When configured in this way, the Apache Spark History Server and the Apache Spark Thrift Server appear as Stopped.

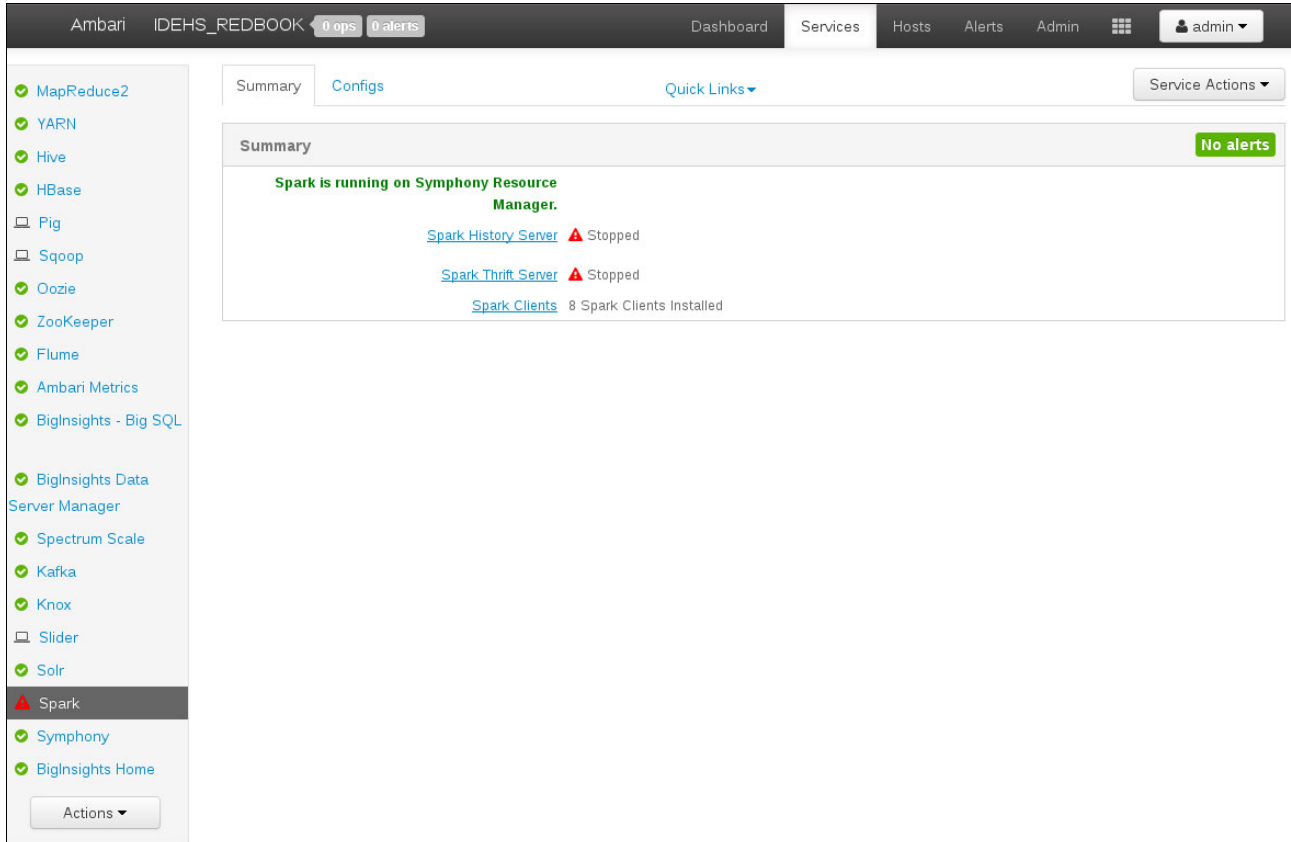


Figure 4-24 Ambari Dashboard showing Apache Spark is running on Symphony Resource Manager

To enable the Apache Spark History Server, complete the following steps:

1. From command line, edit the `start-history-server.sh` file in the `sbin` directory of Apache Spark, as shown in Figure 4-25 on page 73. By default, this file is in `/usr/iop/4.1.0.0/spark/sbin`.

```

root@mn02-dat:/usr/iop/4.1.0.0/spark/sbin
[root@mn02-dat sbin]# pwd
/usr/iop/4.1.0.0/spark/sbin
[root@mn02-dat sbin]# ls
slaves.sh                start-slaves.sh
spark-config.sh          start-thriftserver.sh
spark-daemon.sh          stop-all.sh
spark-daemons.sh         stop-history-server.sh
start-all.sh             stop-master.sh
start-history-server.sh   stop-mesos-dispatcher.sh
start-master.sh           stop-mesos-shuffle-service.sh
start-mesos-dispatcher.sh stop-shuffle-service.sh
start-mesos-shuffle-service.sh stop-slave.sh
start-shuffle-service.sh stop-slaves.sh
start-slave.sh            stop-thriftserver.sh
[root@mn02-dat sbin]# vi start-history-server.sh

```

Figure 4-25 Edit `start-history-server.sh` in the `sbin` directory of Apache Spark installation

2. Add the GPFS library to the `SPARK_CLASSPATH` and `LD_LIBRARY_PATH` lines inside this file, as shown in Figure 4-26 and Figure 4-27.

```

export SPARK_CLASSPATH=/usr/lpp/mmfs/hadoop/*
export LD_LIBRARY_PATH=/usr/lpp/mmfs/hadoop/:$LD_LIBRARY_PATH

```

Figure 4-26 Add the GPFS library to `SPARK_CLASSPATH` and `LD_LIBRARY_PATH`

```

root@mn02-dat:/usr/iop/4.1.0.0/spark/sbin

sbin=`dirname "$0"`
sbin=`cd "$sbin"; pwd`

. "$sbin/spark-config.sh"
. "$SPARK_PREFIX/bin/load-spark-env.sh"

#ADD TO EXPORT GPFS LIBRARY
export SPARK_CLASSPATH=/usr/lpp/mmfs/hadoop/*
export LD_LIBRARY_PATH=/usr/lpp/mmfs/hadoop/:$LD_LIBRARY_PATH

if [ $# != 0 ]; then
    echo "Using command line arguments for setting the log directory is deprecated. Please "
    echo "set the spark.history.fs.logDirectory configuration option instead."
    export SPARK_HISTORY_OPTS="$SPARK_HISTORY_OPTS -Dspark.history.fs.logDirectory=$1"
fi

```

Figure 4-27 Add the GPFS library to `SPARK_CLASSPATH` and `LD_LIBRARY_PATH`

- Open Ambari, then open the Apache Spark service. Go to the Configs tab and check the `spark.eventLog.enabled` parameter. Ensure that it is set to true, as shown in Figure 4-28.

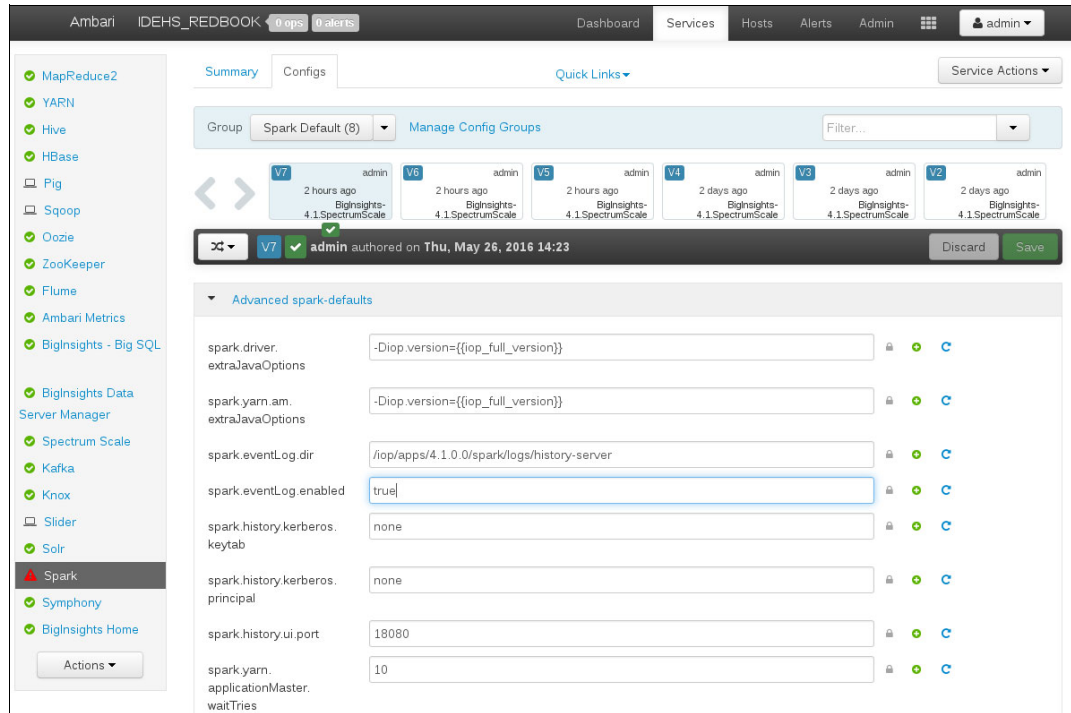


Figure 4-28 Ambari Dashboard: Apache Spark service window

- Click the **Summary** tab, and then click the **Spark History Server**, as shown in Figure 4-29.

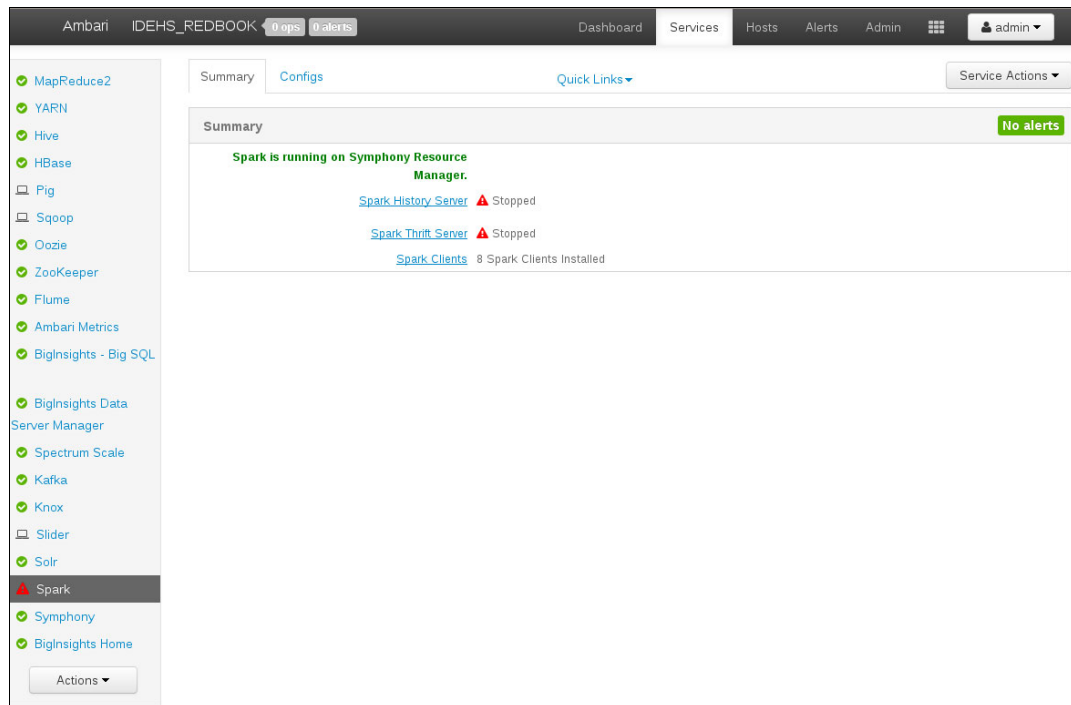


Figure 4-29 Ambari Dashboard - Apache Spark service - Summary tab



- Click the button next to the Apache Spark History Server, and then click **Start**, as shown in Figure 4-30.

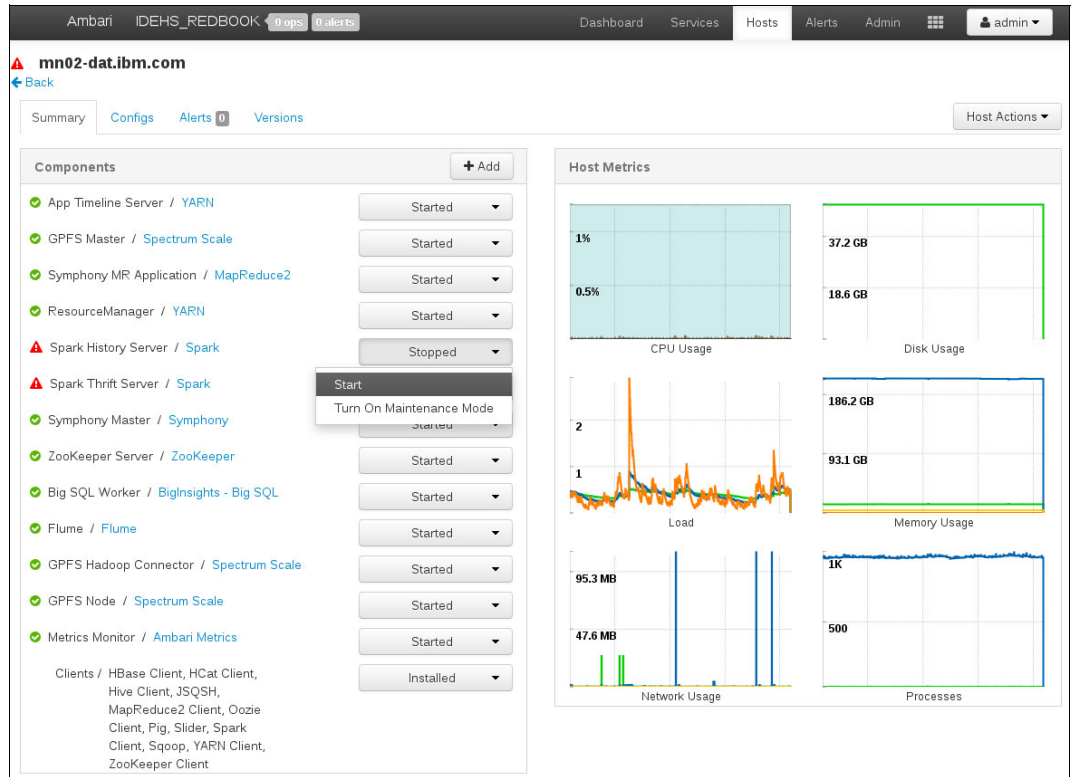


Figure 4-30 Ambari Dashboard: Apache Spark service - Summary tab - Apache Spark History Server - Start

- After the service starts, click **Back** to return to the Apache Spark service. The Apache Spark History Server is already running. To open the UI, click **Quick links** and then click **Spark History Server**, as shown in Figure 4-31.

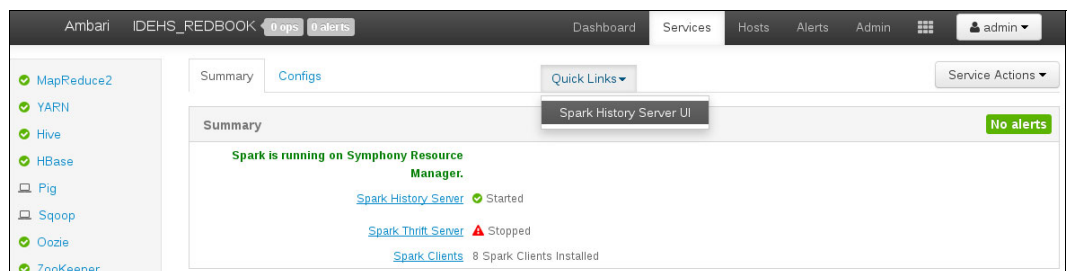


Figure 4-31 Apache Spark History Server started

## 4.4 Deployment and operation tools

Daily, the IT staff is monitoring systems to ensure that the service level agreement (SLA) is met and also administering the deployment of new packages that eventually demand more hardware resources. The volume of data is constantly growing due to overwhelming amount of log and trace entries that are generated per transaction, along with data to support core business operations. Thus, the complexity of system administration is gradually and often dramatically increasing in a short period.

Deployment and operational tools, including mechanisms of monitoring and administering systems, are vital to business operation as much as an unexpected systems outage can affect revenue. In addition, a highly available and reliable analytics system can help position your business ahead of the competition, providing valuable insights to support your business decisions in a faster manner, in areas of marketing, customized campaign, and discounts to maximize your profit.

#### 4.4.1 List of tools

The available tools for the operations team to monitor and administer the IBM Data Engine for Hadoop and Spark solution are the following ones:

- ▶ *Ambari Dashboard* is a web interface to monitor and administer available services in the cluster.
- ▶ *IBM Platform Management Console* is a web interface to IBM Spectrum Symphony and the Application Service Controller for IBM Spectrum Symphony (formerly Platform Symphony).
- ▶ The *EGOSH* command line is an administrative command interface to Enterprise Grid Orchestrator (EGO) for cluster management and control commands that are available in the IBM Spectrum Symphony grid.



# Multitenancy

This chapter provides additional details about using IBM Data Engine for Hadoop and Spark for multitenancy requirements, and what configuration can be done by using IBM Spectrum Symphony (formerly IBM Platform Symphony) to support the multitenancy objectives.

The following topics are described in this chapter:

- ▶ Introduction to multitenancy
- ▶ IBM Spectrum Computing resource manager
- ▶ Configuring multitenancy for MapReduce workloads

## 5.1 Introduction to multitenancy

*Multitenancy* is a reference to the operation mode of software where multiple independent instances of one or multiple applications operate in a shared environment.<sup>1</sup> The instances (tenants) are logically isolated, but physically integrated. The degree of logical isolation must be complete, but the degree of physical integration varies. The more physical integration, the harder it is to preserve the logical isolation. The tenants (application instances) can be representations of organizations that obtained access to the multitenant application (this is the scenario of an ISV offering services of an application to multiple customer organizations). The tenants can also be multiple applications competing for shared underlying resources (this is the scenario of a private or public cloud where multiple applications are offered in a common cloud environment).

For example, an organization has many lines of business (LOBs). Each LOB can have many groups, and each group can have access to several applications. All the applications run on the same platform, which is IBM Data Engine for Hadoop and Spark. In this case, the organization must set up multitenancy to ensure that all the applications can work according to their service-level agreement (SLA). Figure 5-1 shows an example of multitenancy in an organization.

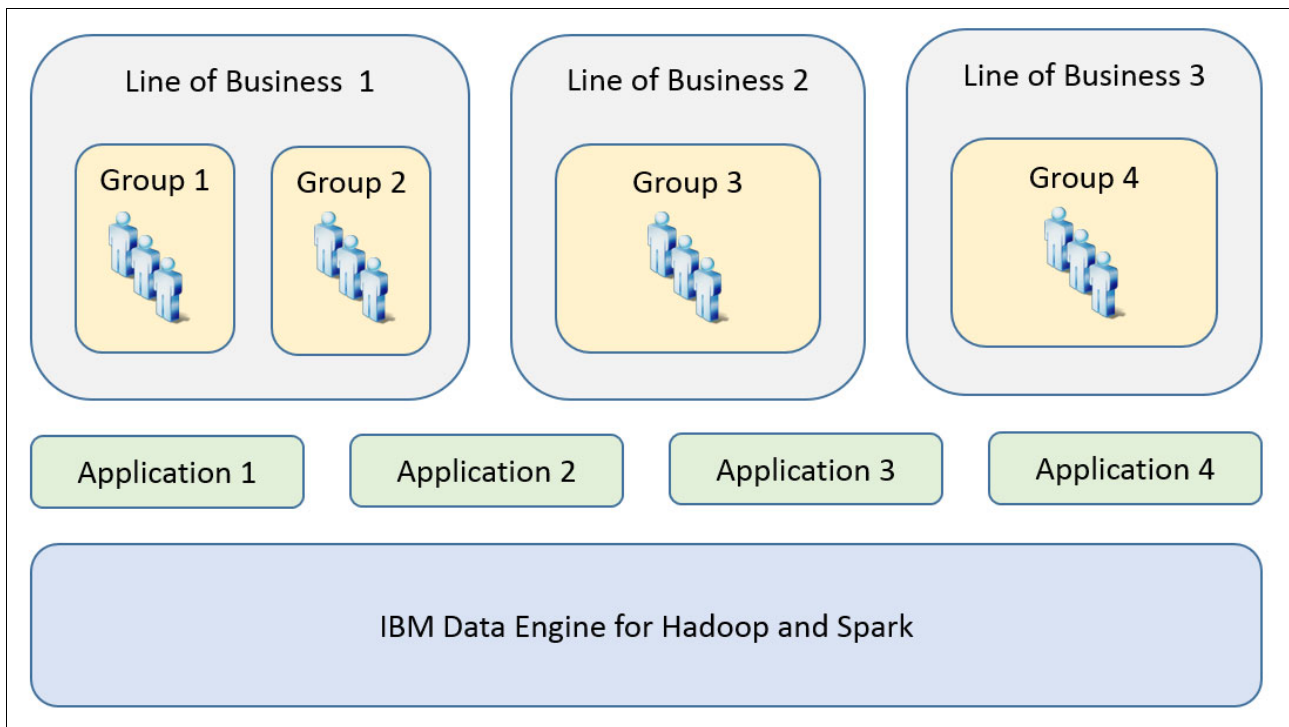


Figure 5-1 Multitenancy architecture example

<sup>1</sup> <http://www.gartner.com/it-glossary/multitenancy/>

## 5.2 IBM Spectrum Computing resource manager

IBM Spectrum Symphony multitenancy capability is possible because of the resource manager that is available as a core component of the solution: The IBM Spectrum Computing resource manager.

This resource orchestrator, also known as the Enterprise Grid Orchestrator (EGO), manages the supply and distribution of resources, making them available to applications. It provides a full suite of services to support and manage resource orchestration in a cluster, including cluster management, configuration and auditing of service-level plans, failover capabilities, monitoring, and data distribution. Only the resource requirements are considered when allocating resources, thus letting the business services use the resources with no interference.

EGO uses resource groups to organize and manage the supply of resources, which are then allocated to different workloads according to policies. Resource groups can be static or dynamic, and use different attributes of hosts to define membership, or simply tags. Resources can also be logical entities that are independent of nodes (bandwidth capacity and software licenses).

Those resources are used through consumers. A consumer is a logical structure that creates the association between the workload demand and the resource supply. Consumers are organized hierarchically into a tree structure to reflect the structure of a business unit, department, projects, and more.

Figure 5-2 shows different consumers that are mapped to different resource pools.

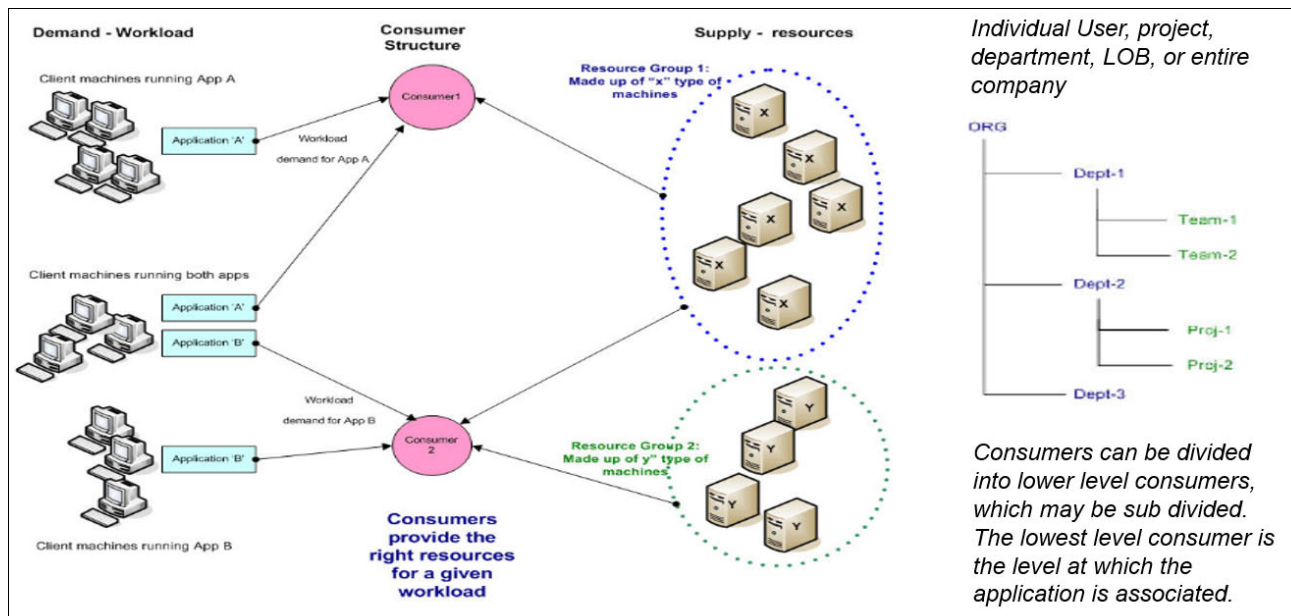


Figure 5-2 Consumers that are defined to use supplies that are organized in resource groups

Then, policies that assign different amounts of resources to different consumers are defined in the resource plan, as shown in Figure 5-3.

- Sharing while preserving ownership
- Change the plan 'on the fly' while workload is running
- Allocations flex during runtime to reflect business priorities – Dynamic Allocation
- Enables application level SLA management

The screenshot displays the 'Resource Plan' configuration for 'ComputeHosts'. It features a table with columns for 'Consumer', 'Owned Slots', 'Consumer Rank', 'Lend | Limit', 'Borrow | Limit', and 'Model | Share | Ratio | Limit'. The table lists various consumers such as 'SymTesting', 'SampleApplications', 'SymExec', and 'MapReduceConsumer'. To the right, there are two detailed panels: 'Lend Details' for 'MapReduce61' and 'Borrow Details' for 'MapReduceDefault'. Both panels show 'Total' limits and a list of consumers to lend from or borrow from, with checkboxes and input fields for each.

Figure 5-3 Resource plan definition

The resource plan allows an SLA to be created so that each LOB can meet its objectives, while sharing a common set of resources. The resource requirements can accommodate multi-dimensional resource allocations where each allocation can request different amounts of physical resource types, including but not limited to CPU, cores, memory, and number of disks.

**Note:** For more information about the technology behind IBM Spectrum Computing resource scheduler, see the following website:

<http://ibm.co/1TKU1Mg>

**Note:** You must create an IBM ID to access this website and retrieve the information.

## 5.3 Configuring multitenancy for MapReduce workloads

This section shows how to configure multitenancy for MapReduce workloads.

### 5.3.1 Monitoring MapReduce jobs by using IBM Spectrum Symphony

When IBM Spectrum Symphony is installed and set up by using Ambari, IBM Spectrum Symphony creates a MapReduce 7.1 workload by default. To test the default workload, complete the following steps:

1. Log in to the command line and run the Hadoop workload, as shown in Example 5-1 on page 81.

### Example 5-1 Sample MapReduce job

```
hadoop jar
/usr/iop/4.1.0.0/hadoop-mapreduce/hadoop-mapreduce-client-jobclient.jar sleep
-r 10 -m 20 -mt 1000
```

2. After running the job, you see that the job is running in IBM Spectrum Symphony Session Manager (SSM), which marks that the job is managed by IBM Spectrum Symphony. Figure 5-4 shows the output when you run the job.

```
[root@mn01-dat ~]# hadoop jar /usr/iop/4.1.0.0/hadoop-mapreduce/hadoop-mapreduce
-client-jobclient.jar sleep -r 10 -m 20 -mt 1000
WARNING: Use "yarn jar" to launch YARN applications.
OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=512m; support was
removed in 8.0
16/05/18 17:12:04 INFO Configuration.deprecation: mapred.map.tasks is deprecated
. Instead, use mapreduce.job.maps
16/05/18 17:12:04 INFO internal.MRJobSubmitter: Connected to JobTracker(SSM)
16/05/18 17:12:04 INFO Configuration.deprecation: mapred.output.key.comparator.c
lass is deprecated. Instead, use mapreduce.job.output.key.comparator.class
16/05/18 17:12:04 INFO Configuration.deprecation: mapred.compress.map.output is
deprecated. Instead, use mapreduce.map.output.compress
16/05/18 17:12:04 INFO internal.MRJobSubmitter: Job <Sleep job> submitted, job i
d <424>
16/05/18 17:12:04 INFO internal.MRJobSubmitter: Job will not verify intermediate
data integrity using checksum.
16/05/18 17:12:04 INFO mapreduce.Job: Running job: job_ssm_0424
```

Figure 5-4 Running the MapReduce example

3. Open the IBM Spectrum Symphony web interface and log in, as shown in Figure 5-5. By default, the URL is `http://<Symphony_Master_hostname>:58089/platform/`.

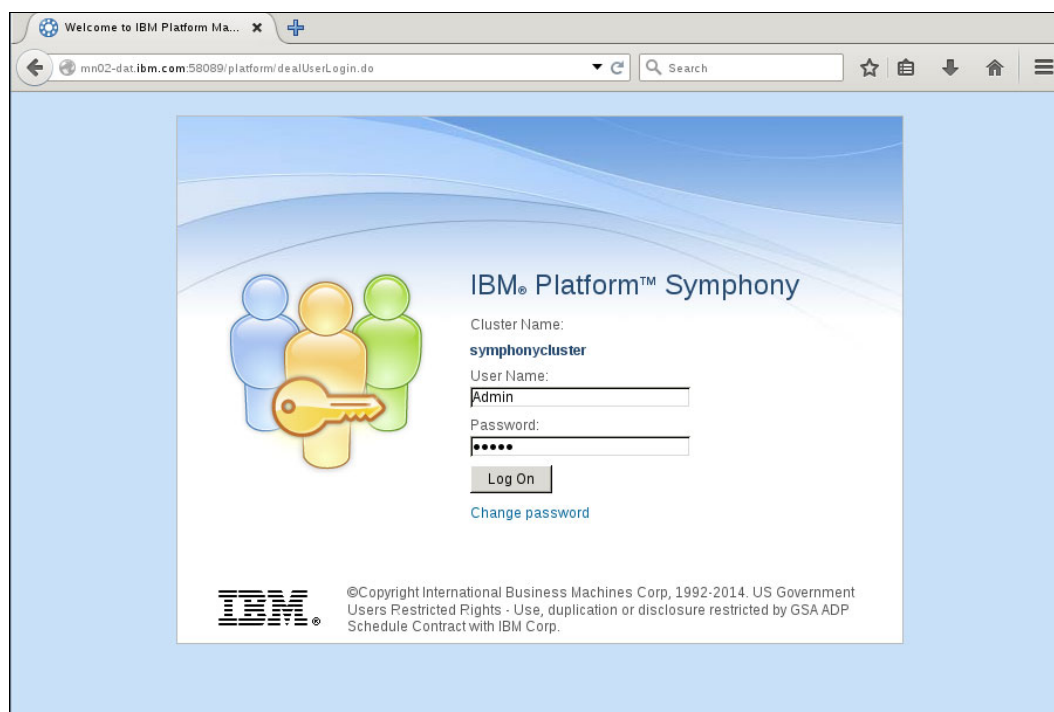


Figure 5-5 IBM Spectrum Symphony web interface

4. Click **Workload** → **MapReduce** → **Jobs**, as shown in Figure 5-6.



Figure 5-6 Open MapReduce Jobs

5. The job is displayed as the MapReduce 7.1 application, as shown in Figure 5-7.

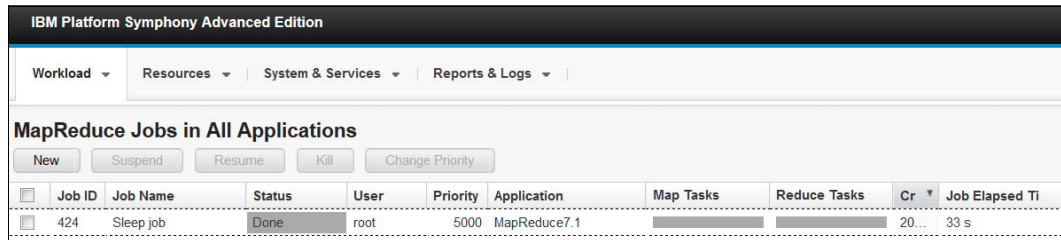


Figure 5-7 Display MapReduce sample job

By using the default MapReduce 7.1 application profile, IBM Spectrum Symphony balances all MapReduce jobs that are running in the system. When running one job, the job takes 160 slots, as shown in Figure 5-8.

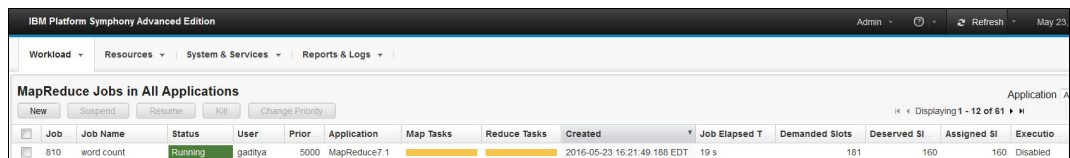


Figure 5-8 MapReduce job taking 160 slots

When running three jobs, each job gets around 53 slots, which means the 160 slots are distributed between each job, as shown in Figure 5-9 on page 83.



Job	Job Name	Status	User	Prior	Application	Map Tasks	Reduce Tasks	Created	Job Elapsed T	Demanded Slots	Deserved SI	Assigned SI	Executio
817	word count	Running	ngoly	5000	MapReduce7.1			2016-05-23 17:25:52.342 EDT	119 s	161	53.33	53	Disabled
816	word count	Running	rtkatah	5000	MapReduce7.1			2016-05-23 17:24:50.748 EDT	181 s	102	53.33	54	Disabled
815	word count	Running	gadlya	5000	MapReduce7.1			2016-05-23 17:24:46.846 EDT	184 s	101	53.33	53	Disabled

Figure 5-9 MapReduce jobs sharing the available 160 slots

**Note:** To specify the default application name of all MapReduce jobs, open the `pmr-site.xml` file. By default, the file is in the following directory:

`/opt/ibm/platformsymphony/soam/mapreduce/conf/pmr-site.xml`.

Find the following string inside the file and change the value as needed:

```
<property>
<name>mapreduce.application.name</name>
<value>MapReduce7.1</value>
<description>The mapreduce application name.</description>
</property>
```

### 5.3.2 Creating an application profile

To specify multiple workloads in IBM Spectrum Symphony, you must create an application profile for each type of workload. To create an application profile, complete the following steps:

1. Open the IBM Spectrum Symphony UI and click **Workload** → **MapReduce** → **Application Profile**, as shown in Figure 5-10.

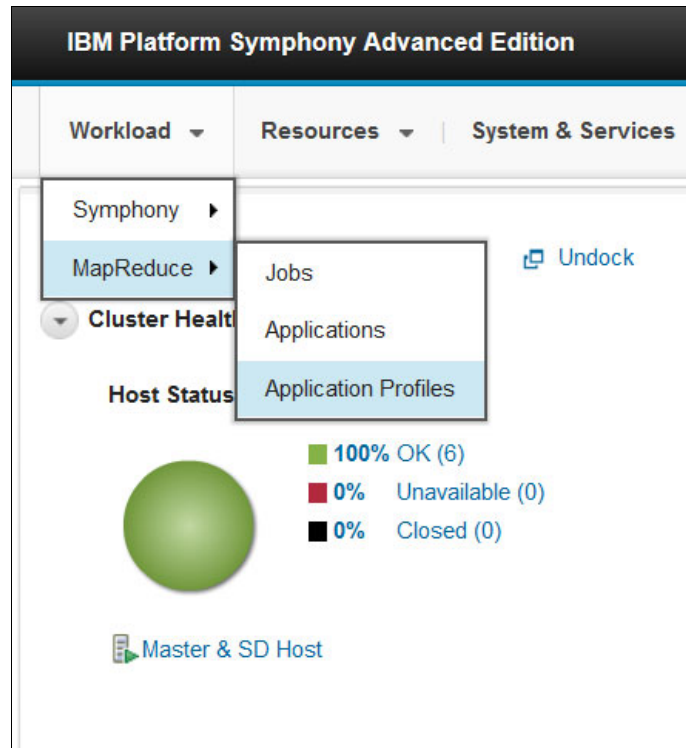


Figure 5-10 Open MapReduce Application Profiles

2. Click **Add** in the MapReduce application profiles window, as shown in Figure 5-11.

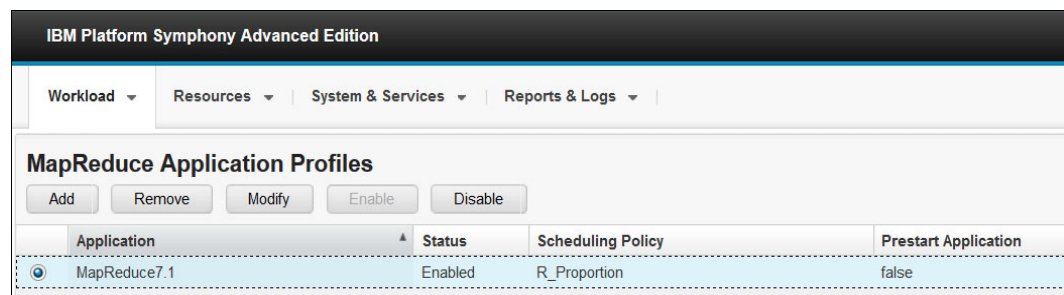


Figure 5-11 MapReduce Application Profiles window

3. Provide an application name and job priority. Leave the field “User who starts job tracker and runs job” empty. Then, assign users/groups for the consumer user of the new application profile. In this example, it builds an application profile for an application called “App1”, and the priority is default. Click **Add** to create the application profile, as shown in Figure 5-12 on page 85.

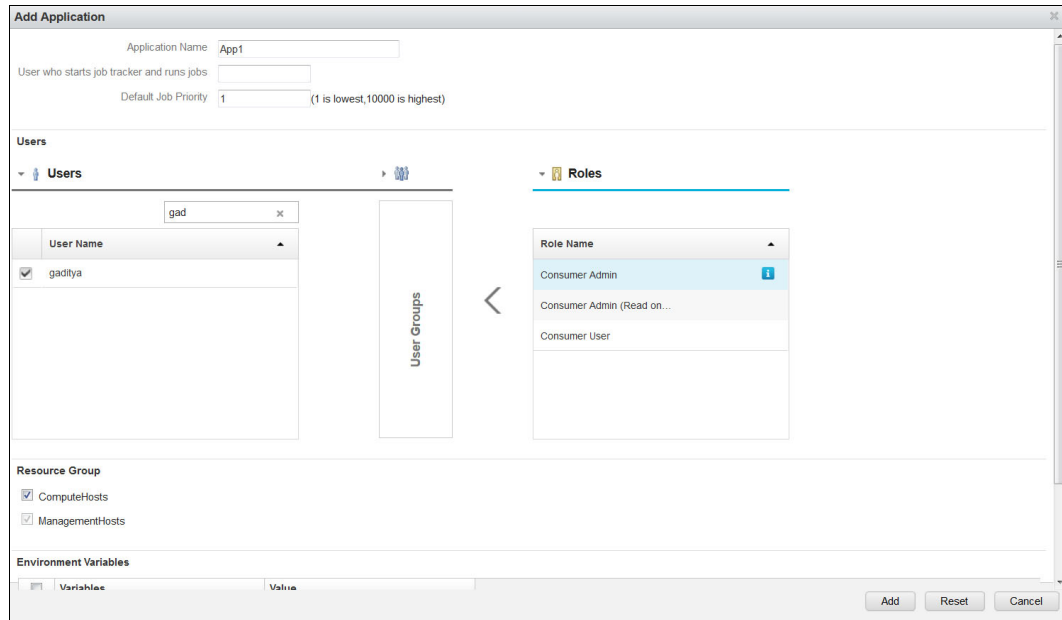


Figure 5-12 Configuration for new application profile

**Note:** If you do not see the user or groups in the list of users, restart the IBM Spectrum Symphony service by using Ambari to update the list.

4. Click **Yes** to confirm that the job tracker user is blank, as shown in Figure 5-13.

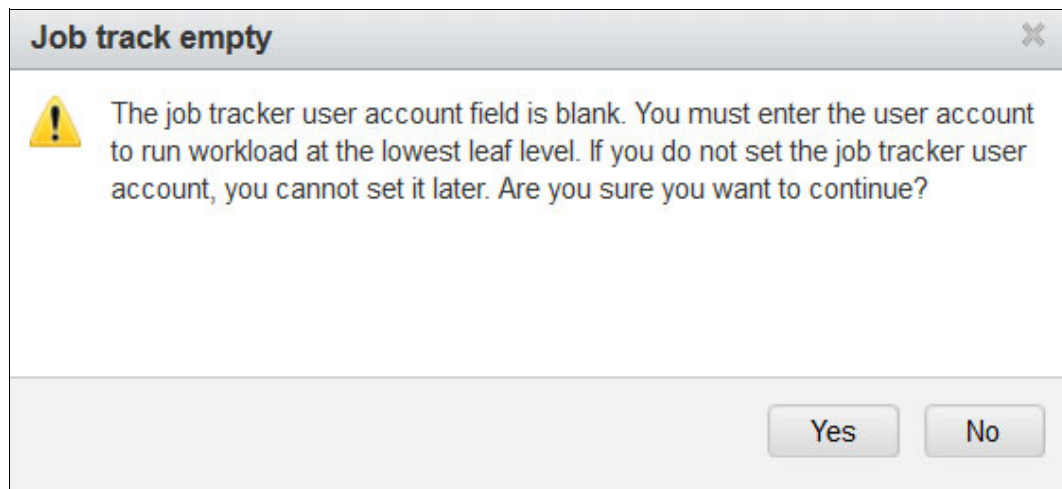


Figure 5-13 Confirm that the job tracker user is blank

- You can see the new application profile now. If it is not displayed yet. Click **Refresh**, as shown in Figure 5-14.

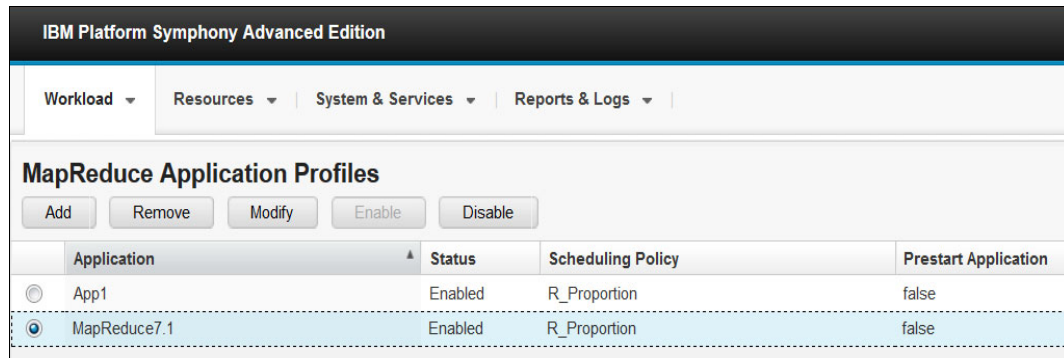


Figure 5-14 New application profile in the MapReduce Application Profiles window

- Click **Resource** → **Resource Planning** → **Resource Plan (Multi-dimensional)**, as shown in Figure 5-15.

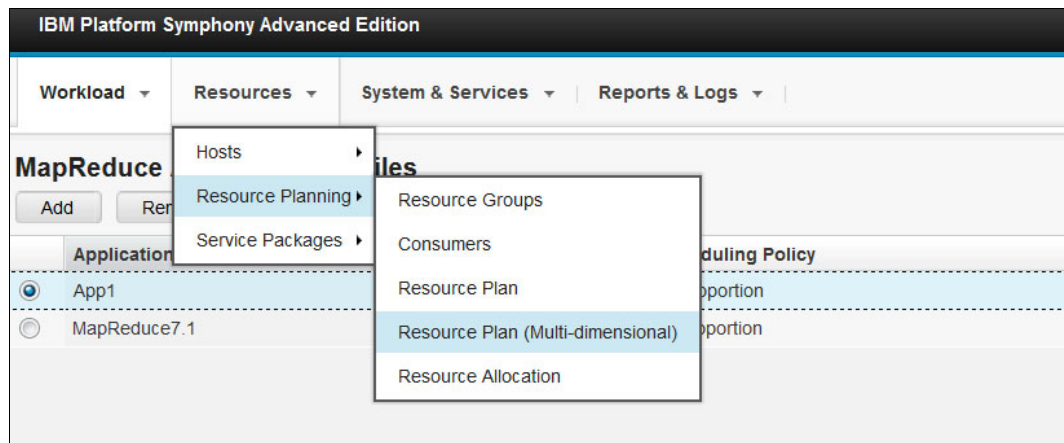


Figure 5-15 Open Resource Plan (Multi-dimensional)

- Click the **Consumers** tab, and then click **Add Consumer**, as shown in Figure 5-16 on page 87.

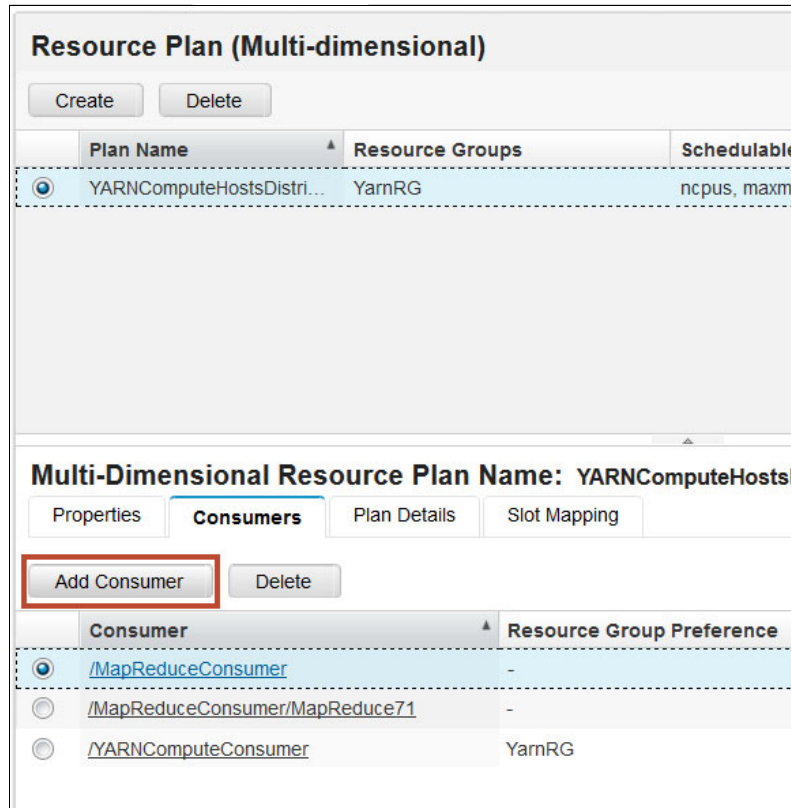


Figure 5-16 Add Consumer in the Resource Plan (Multi-dimensional) window

- Check the recently created application profile name, and then click **Apply**. Figure 5-17 shows adding the App1 Consumer.

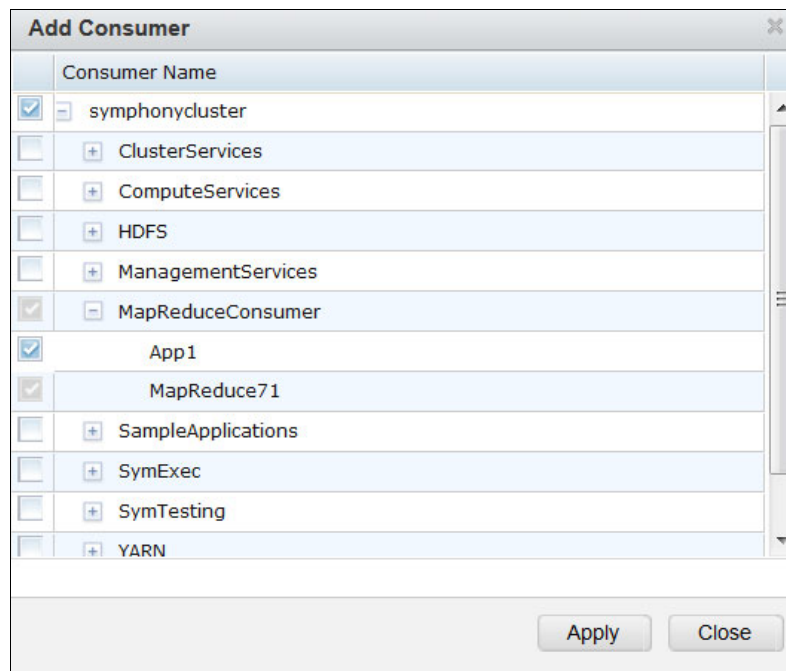


Figure 5-17 Add Consumer window

9. Click **Workload** → **MapReduce** → **Application Profiles**, as shown in Figure 5-18.

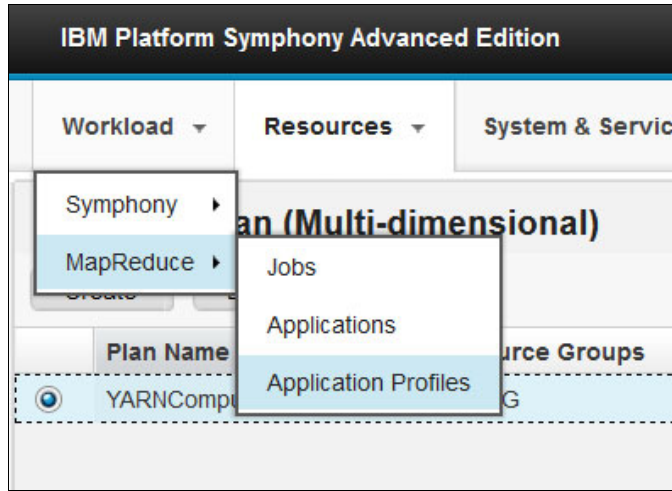


Figure 5-18 Open Application Profiles

10. Next, copy the definition from the existing MapReduce 7.1 application profile. Click the MapReduce7.1 application profile, and then click **Modify**, as shown in Figure 5-19.

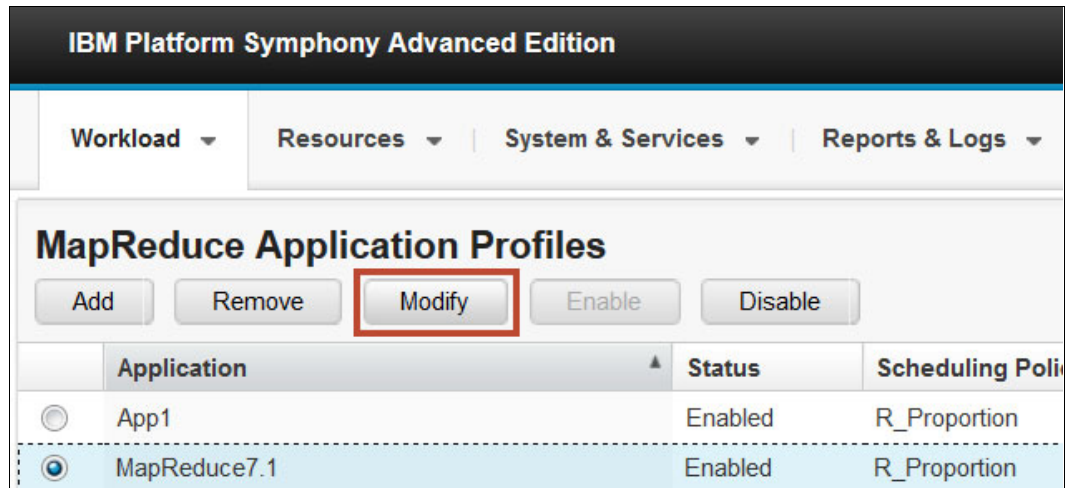


Figure 5-19 Open Modify Application Profiles window

11. Click **Export**, as shown in Figure 5-20.



Figure 5-20 Panel showing the Export button

12. Open the MapReduce7.1.xml file, then make the changes that are shown in Example 5-2 on page 89.

Example 5-2 Changes to the MapReduce7.1.xml file

```
<Consumer applicationName="App1" consumerId="/MapReduceConsumer/App1"
numOfSlotsForPreloadedServices="10000" policy="R_Proportion"
preStartApplication="false" taskHighWaterMark="1.0" taskLowWaterMark="1.0"
workloadType="MapReduce" preemptionCriteria="PolicyDefault"
enableSelectiveReclaim="false" preemptionScope="LowerOrEqualRankedSessions"
schedulingAffinity="None"/>

...

<osType fileNamePattern="s" logDirectory="${SOAM_HOME}/mapreduce/logs/tasklogs"
name="all"
startCmd="${PMR_HOME}/${PMR_VERSION}/${EGO_MACHINE_TYPE}/etc/RunMapReduceService.sh"
subDirectoryPattern="%applicationName%/%sessionId%/task_%taskId%"
workDir="${PMR_HOME}/work/App1/${SUB_WORK_DIR}">
```

App1 is the name of the new application profile. Save the file with another name.

13. Go back to IBM Spectrum Symphony web interface, click the App1 application profile, and then click **Modify**, as shown in Figure 5-21.

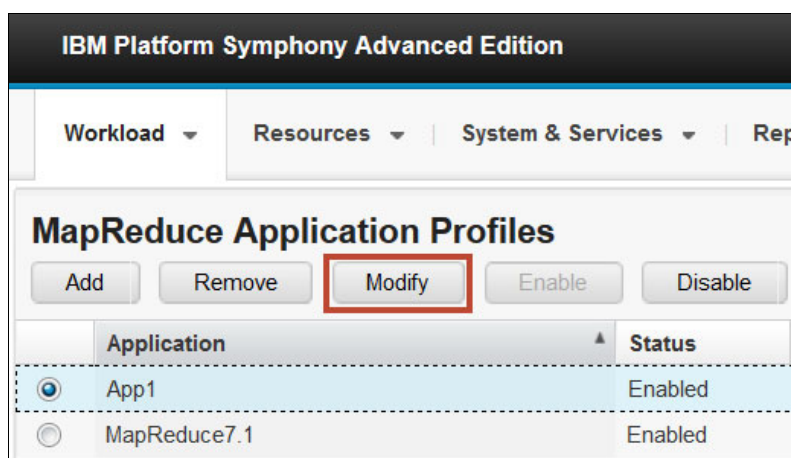


Figure 5-21 Modify the newly created Application Profile

- Click **Import** (Figure 5-22). Select the recently created XML file and click Import. Then, click **Save**.

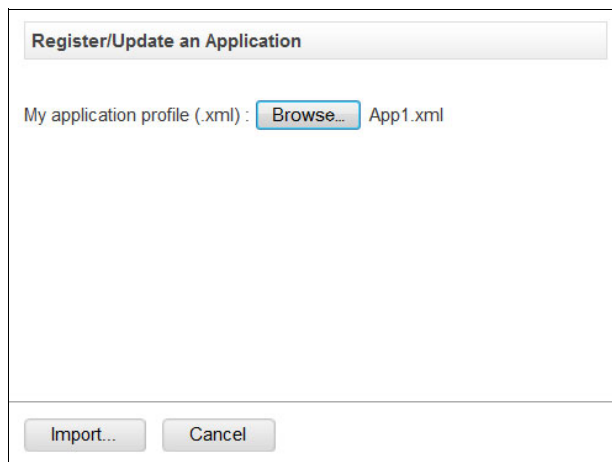


Figure 5-22 Application Profile Import window

- Now, there is a new MapReduce application profile that is ready to use, as shown in Figure 5-23. Repeat step 2 on page 84 through step 10 on page 88 to add another application profile.

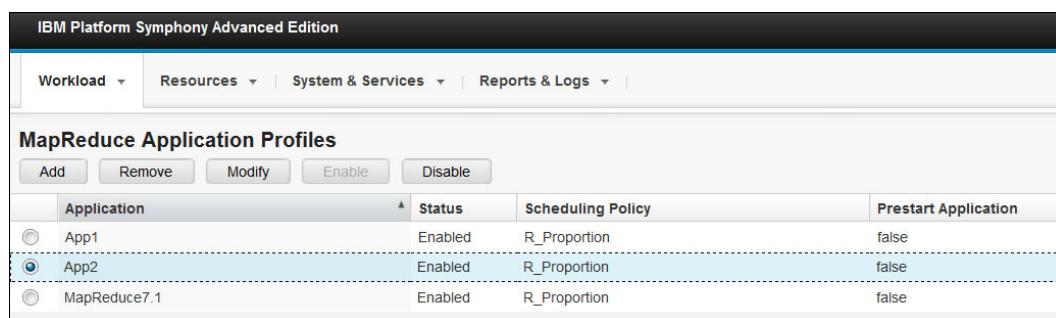


Figure 5-23 New Application Profiles

- Submit a new workload and specify the application name so it uses the newly created application profile. Check that you are using the new application profile. To specify the application name, use `-Dmapreduce.application.name=<application_name>`, as shown in Example 5-3.

*Example 5-3 Map Reduce sample workload with the application name*

---

```

hadoop jar
/usr/iop/4.1.0.0/hadoop-mapreduce/hadoop-mapreduce-examples-2.7.1-IBM-11.jar
wordcount -Dmapreduce.application.name=App1 -Dmapreduce.job.reduces=100
/tmp/output3 wc/output

#-Dmapreduce.application.name must be specified before other parameters.

```

---

- Open the IBM Spectrum Symphony web interface, and click **Workload** → **MapReduce** → **Jobs**. The recently run job uses the new application profile, as shown in Figure 5-24 on page 91.



Job ID	Job Name	Status	User	Priority	Application	Map Tasks	Reduce Tasks	Created	Job Elapsed	Demanded Slots	Deserved Slots	Assigned Slots	Execution
104	word count	Running	njoly	5000	App2			2016-05-27 12:56:21.560	69 s	242	124	124	Disabled
104	word count	Running	gadiya	5000	App1			2016-05-27 12:56:20.876	70 s	228	124	124	Disabled

Figure 5-24 MapReduce Jobs that uses the new application profile

### 5.3.3 Adding users or groups to an existing application profile

You can add additional users or groups into an existing application profile to grant permission for new users or groups to use the application profile. Complete the following steps:

1. Ensure that you already created users or groups in all operating system (OS) of each node. To create users or groups in all of the nodes, run the `xdsh` command, as shown in Example 5-4.

*Example 5-4 The xdsh command for creating groups and adding users*

```
xdsh teamredbook 'groupadd -g 30600 redbookgroup'
xdsh teamredbook 'usermod -a -G redbookgroup gadiya'
```

In this case, `teamredbook` is the node group that is defined in Extreme Cluster/Cloud Administration Toolkit (xCAT).

**Note:** To list existing node groups, run the following command in the system management node:

```
lsdef -t group
```

2. Open the IBM Spectrum Symphony web interface, and then open the Application Profile window. Click **Modify** on the application profile that you want to add users or groups, as shown in Figure 5-25.

Application	Status	Scheduling Policy	Prestart Application
App1	Enabled	R_Proportion	false
MapReduce7.1	Enabled	R_Proportion	false

Figure 5-25 Modify Application profile

3. Click the **Users** tab, as shown in Figure 5-26.

The screenshot shows the 'Application Profile' configuration page for 'App1'. At the top, there are two tabs: 'Application Profile' (selected) and 'Users'. Below the tabs is a dropdown menu for 'Advanced Configuration'. Underneath, there are two radio buttons for 'SOAM Version': 'Use specific version' (selected) and 'Always use the latest available version'. The 'Use specific version' option has a dropdown menu showing '7.1'. A warning icon and text state: 'The update will terminate all workload for this application.' Below this is a section titled 'General Settings' with a downward arrow. It contains three fields: 'Application Name(\*)' with the value 'App1', 'Shared file system location for this application' with the value '\$\${EGO\_SHARED\_TOP}/soam/work', and 'Default Service Definition' with a dropdown menu showing 'MapReduceService'.

Figure 5-26 Application profile tabs

4. Click **Roles** and choose the roles for the new users and groups.

5. Choose the users or groups that you want to add by selecting the check boxes next to their names. Click **Save** when finished, as shown in Figure 5-27.

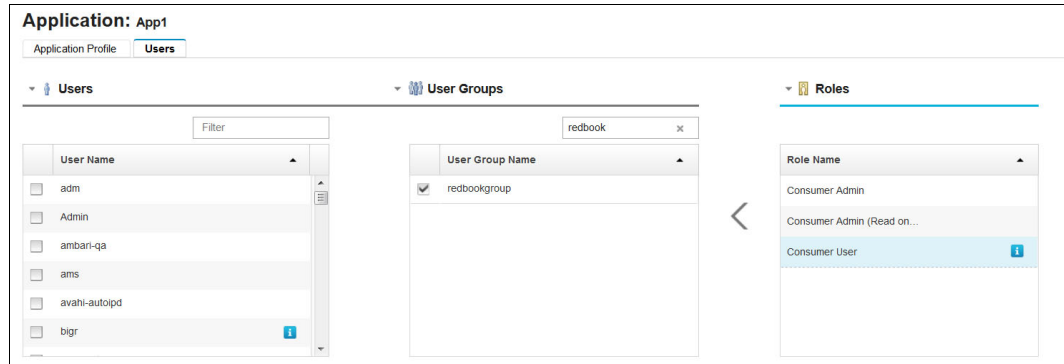


Figure 5-27 Add groups to an existing application profile

### 5.3.4 Configuring the share ratio between application profiles

You can configure which application profile has more share ratio compared to other application profiles. Share ratio refers to how many slots can be shared with an application profile. A higher share ratio means that the application profile gets a higher number of slots.

To configure the share ratio, complete the following steps:

1. From the IBM Spectrum Symphony web interface, click **Resources** → **Resource Planning** → **Resource Plan (Multi-dimensional)**. Click the **Plan Details** tab, then in the Consumer pane, select **/MapReduceConsumer**, as shown in Figure 5-28.

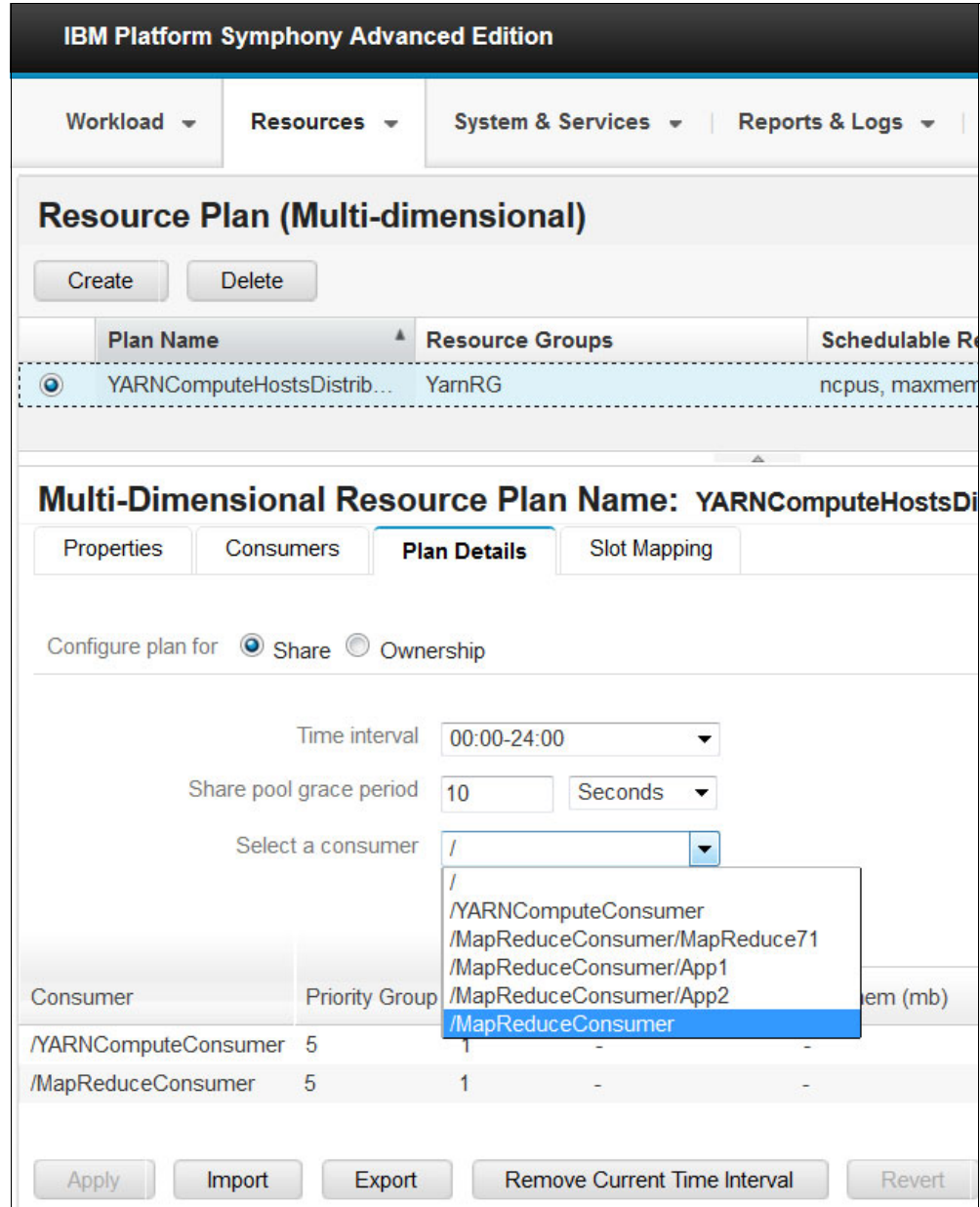


Figure 5-28 Select Consumer in Multi-dimensional resource plan

- Modify the share ratio of the application number by clicking the number. Click **Apply**, as shown in Figure 5-29.

Consumer	Priority Group	Share Ratio	Limit(Absolute)	
			ncpus	maxmem (mb)
/MapReduceConsumer/MapReduce71	5	1	-	-
/MapReduceConsumer/App1	5	1	-	-
/MapReduceConsumer/App2	5	<input type="text" value="2"/>	-	-

Figure 5-29 Configure the share ratio

- Run the jobs. The application profile with the higher share ratio has the higher number of slots, as shown in Figure 5-30.

MapReduce Jobs in All Applications											Application All	
Job ID	Job Name	Status	User	Priority	Application	Map Tasks	Reduce Tasks	Created	Job Elapsed	Demanded Slots	Deserved Slots	Assigned Slots
105	word count	Running	njoly	5000	App2	<div style="width: 100%; height: 10px; background-color: green;"></div>	<div style="width: 100%; height: 10px; background-color: orange;"></div>	2016-05-27 13:05:30.710	51 s	150	150	150
105	word count	Running	gadiya	5000	App1	<div style="width: 100%; height: 10px; background-color: green;"></div>	<div style="width: 100%; height: 10px; background-color: orange;"></div>	2016-05-27 13:05:28.787	53 s	150	98	98

Figure 5-30 MapReduce jobs after configuring the share ratio

### 5.3.5 Configuring slot mapping

You can define how much CPU and memory can be used per slot in IBM Spectrum Symphony so that you can configure how much of the hardware resources can be used by all the applications and workloads that is running on IBM Spectrum Symphony.

To configure the mapping, complete the following steps:

1. From the IBM Spectrum Symphony web interface, click **Resources** → **Resource Planning** → **Resource Plan (Multi-dimensional)**, as shown in Figure 5-31.

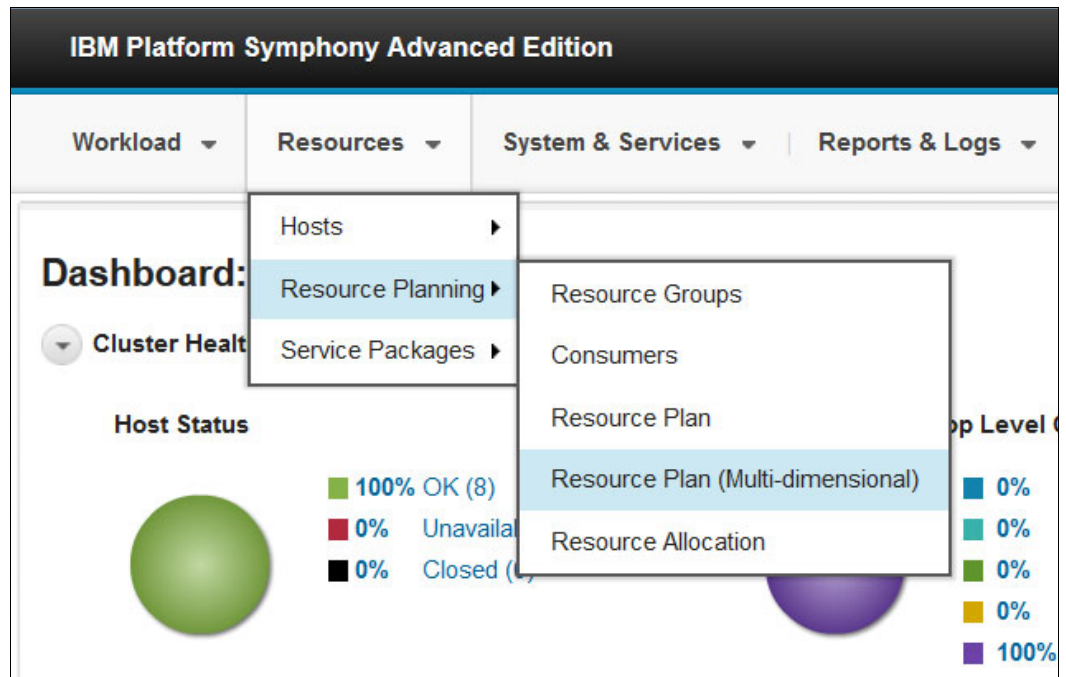


Figure 5-31 Open Resource Plan (Multi-dimensional)

- Click the **Slot Mapping** tab, then change ncpus and maxmem. In Figure 5-32, you define one slot with 1 ncpus and 4096 MB maxmem for /MapReduceConsumer.

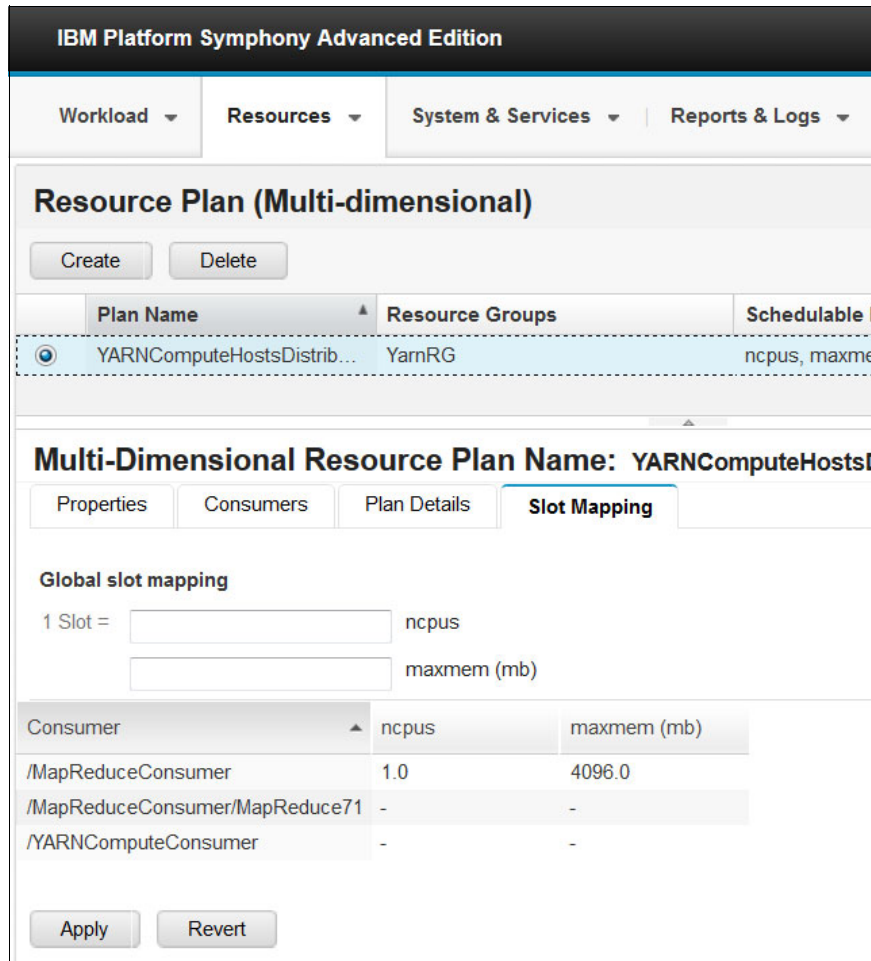


Figure 5-32 Configure slot mapping

**Note:** You can also configure the global slot mapping, which affects all the consumers. Ensure that there are no workloads running that can use the consumer to configure the slot mapping.

### 5.3.6 Configuring the priority for running jobs

You can configure the priority for jobs that are already running, which can be useful when there are multiple long running jobs and you must prioritize the slots allocations for each job. To configure the priority for running jobs, complete the following steps:

1. From the IBM Spectrum Symphony web interface, click **Workload** → **MapReduce** → **Jobs**. Select the check box next to the job that you want to modify, and then click **Change Priority**, as shown in Figure 5-33.

Job ID	Job Name	Status	User	Priority	Application	Map Tasks	Reduce Tasks	Created	Job Elapsed	Demanded Slots	Deserved Slots	Assigned Slots	
<input checked="" type="checkbox"/>	1624	word count	Running	gaditya	5000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:54.890 ...	5 s	150	82.67	82
<input type="checkbox"/>	1623	word count	Running	npjoly	5000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:53.493 ...	6 s	150	82.67	82
<input type="checkbox"/>	1622	word count	Running	rikatahi	5000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:51.909 ...	8 s	150	82.67	84

Figure 5-33 MapReduce jobs before changing priority

2. Provide a priority number, as shown in Figure 5-34.

#### Change Job Priority

Change priority of job <1624> of MapReduce7.1 to

1 is lowest, 10000 highest.

Yes
No

Figure 5-34 Change the job priority

3. Do the same for other jobs. After some time, the slots are reconfigured according to the assigned priority, as shown in Figure 5-35.

Job ID	Job Name	Status	User	Priority	Application	Map Tasks	Reduce Tasks	Created	Job Elapsed	Demanded Slots	Deserved Slots	Assigned Slots	
<input type="checkbox"/>	1624	word count	Running	gaditya	1000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:54.890 ...	268 s	168	41.33	44
<input type="checkbox"/>	1623	word count	Running	npjoly	2000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:53.493 ...	270 s	148	82.67	84
<input type="checkbox"/>	1622	word count	Running	rikatahi	3000	MapReduce7.1	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	<div style="width: 100%; height: 10px; background-color: #f08000;"></div>	2016-05-27 01:27:51.909 ...	271 s	136	124	120

Figure 5-35 MapReduce jobs after changing the priority





# A

## Ordering the solution

This appendix describes how to order the solution and how to obtain IBM Lab Services to perform the initial setup of the solution.

The following topics are described in this appendix:

- ▶ Predefined configuration
- ▶ How to use the IBM Configurator for e-business (e-config)
- ▶ Services

## Predefined configuration

As described in Chapter 2, “Solution reference architecture” on page 15, there are two predefined configurations for this solution: *Starter* and *Landing Zone*. Each of these configurations has differences in capacity and resilience. To decide which size is more appropriate for your organization, you can do the sizing by using your own tools or ask IBM for assistance.

If you chose to use the IBM sizing services, you must provide the following information:

- ▶ Is this environment going to be for production or not?
- ▶ Primarily Hadoop or Apache Spark analytic nodes?
- ▶ Raw data sizes?
- ▶ Compressions rates?
- ▶ Shuffle sort storage percentage?
- ▶ Anticipated data growth rate?
- ▶ Preferred drive size?
- ▶ Overrides?

With this information, IBM can recommend a solution with a size that best fits your requirements.

## How to use the IBM Configurator for e-business (e-config)

IBM Configurator for e-business (e-config) is a tool that is available for the following actions:

- ▶ Configuring and upgrading IBM systems and subsystems
- ▶ Configuring multiple product lines with just one tool
- ▶ Checking only the panels that you need, rather than all product categories and configuration options
- ▶ Viewing all your selections from a high level without moving backward
- ▶ Viewing list prices as the configuration is being constructed
- ▶ Using system diagrams to review expansion options, explore configuration alternatives, and know immediately whether the alternatives all work together for an optimal solution

The e-config tool can be found at the following website:

<http://www.ibm.com/services/econfig/announce/index.htm>

**Note:** This section is not intended to be training for the e-config tool or a detailed step by step configuration guide of for the IBM Data Engine for Hadoop and Spark solution. This section is just a guide to find the information about the solution and the services that are associated with it.

This solution has a preconfigured solution under the Power Systems product base in e-config. You can see the proposed solutions at the time of writing in Figure A-1 on page 101.

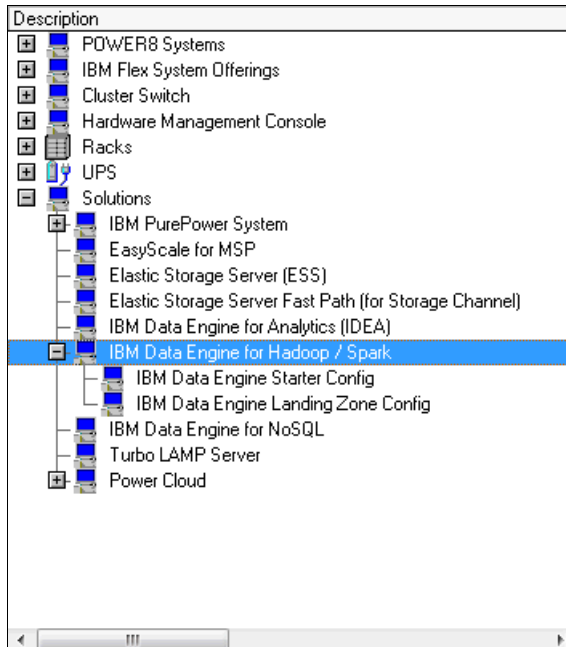


Figure A-1 IBM Data Engine for Hadoop and Spark e-config menu selection

Select the option that applies to your requirements and size it as agreed during the sizing part of the engagement. The selection comes with a rack and the needed nodes, switches, and internal cables for the solution to work.

The solution comes with onsite services from IBM System Lab Services, as shown in Example A-1.

Example A-1 IBM System Lab Services consultation that is included with an IBM Data Engine for Hadoop and Spark order

6911-300	IBM Systems Lab Services for 1 day for Power Systems	1		N/C
0003	Standard Power Systems ServiceUnit for 1-day of onsite consultation	10	29 180,00	OTC

## Services

IBM Systems Lab Services can help with the preparation, setup, and post-installation of the IBM Data Engine for Hadoop and Spark solution. Here is the basic setup list of services for IBM Data Engine for Hadoop and Spark that IBM Systems Lab Services can provide:

- ▶ Conduct project planning and prep work sessions.
- ▶ Perform remote cluster validation before the solution is shipped.
- ▶ Perform onsite cluster start.
- ▶ Perform onsite cluster health check.
- ▶ Perform onsite cluster network integration.

- ▶ Skills mentoring occurs throughout the engagement. This mentoring requires that the client dedicate staff that is responsible for managing the system during the engagement.
- ▶ Create and deliver to the client's project manager an IBM Data Engine for Hadoop and Spark implementation record document that is defined in the deliverable materials section.

**Note:** For more information about IBM Systems Lab Services, see the following website:

<http://www.ibm.com/systems/services/labservices/>



# B

## Script to clone partitions

This appendix provides a script to clone partitions from a source server into a destination server. The script is provided as-is with no warranty of any kind from IBM.

The following topic is described in this appendix:

- ▶ Clone partitions script

## Clone partitions script

To add a node, it is necessary to have a partition layout. As this script uses the IBM Spectrum Scale-File Placement Optimizer (IBM Spectrum Scale-FPO) setup, internal disks are used in the implementation. Also, because all the nodes are homogeneous, the script takes advantage of this homogeneity to clone from the server that runs the script of the partition layout to a defined server.

The script must be able to SSH passwordless to the destination server from the source server as the *root* user. The script ignores sda and sdb disks because they are reserved for the operating system (OS).

Example B-1 shows the clone partitions script.

*Example B-1 The clone\_partitions.sh script*

---

```
#!/bin/ksh
#
# ABSOLUTELY NO WARRANTY OF ANY KIND. USE AT YOUR OWN RISK
#
# Clone partitions for adding new node to IBM Data Engine for Hadoop and Spark
solution
# SSH between nodes must work passwordless for root user
# Nodes MUST be equal
# v0.1 May 2016
#
#set -x

DST_SERVER=$1
SRC_SERVER=`hostname -s`
SGDISK_BIN=`which sgdisk`
SSH_BIN=`which ssh`
SCP_BIN=`which scp`
#Anyone that wants to do this smarter, please do. Will be appreciated.
DISK_LIST=`lsblk | grep disk | grep -v sda | grep -v sdb | awk '{print $1}'`

check_parameters () {
if [[ -z "$DST_SERVER" ]] ; then
    echo "ERROR 10: Must provide the following 1 parameter:
        destination_server"
    exit 10
fi
return
}

check_needed_sw () {
if [[ -e $SGDISK_BIN ]] ; then
    echo "sgdisk is installed."
    echo
else
    echo "ERROR 11: This script needs sgdisk installed"
    echo
    exit 11
fi
return
}
```

```

welcome_note () {
echo
echo "This will clone partitions of

    $DISK_LIST

    from $SRC_SERVER to $DST_SERVER"
echo
echo "You have 3 seconds to cancel the run with Ctrl-C ..."
echo
sleep 3
return
}

read_src_server_partitions () {
for disk in $DISK_LIST
do
    $SGDISK_BIN --backup=/tmp/$SRC_SERVER.$disk.partitions.sgdisk /dev/$disk
done
}

delete_dst_server_partitions () {
for disk in $DISK_LIST
do
    $SSH_BIN $DST_SERVER $SGDISK_BIN -o /dev/$disk
done
return
}

create_dst_server_partitions () {
for disk in $DISK_LIST
do
    $SCP_BIN /tmp/$SRC_SERVER.$disk.partitions.sgdisk
    $DST_SERVER:/tmp/$SRC_SERVER.$disk.partitions.sgdisk
    $SSH_BIN $DST_SERVER $SGDISK_BIN
    --load-backup=/tmp/$SRC_SERVER.$disk.partitions.sgdisk /dev/$disk
    $SSH_BIN $DST_SERVER $SGDISK_BIN -G /dev/$disk
done
return
}

#MAIN
check_needed_sw
check_parameters
welcome_note
read_src_server_partitions
delete_dst_server_partitions
create_dst_server_partitions
echo "Done"
echo
exit 0

```

---





# Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

## IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Some publications that are referenced in this list might be available in softcopy only.

- ▶ *Analytics in a Big Data Environment*, REDP-4877
- ▶ *Apache Spark for the Enterprise: Setting the Business Free*, REDP-5336
- ▶ *Building Big Data and Analytics Solutions in the Cloud*, REDP-5085
- ▶ *Governing and Managing Big Data for Analytics and Decision Makers*, REDP-5120
- ▶ *Implementing an Optimized Analytics Solution on IBM Power Systems*, SG24-8291

You can search for, view, download, or order these documents and other Redbooks, Redpapers, web docs, draft and additional materials, at the following website:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Online resources

These websites are also relevant as further information sources:

- ▶ e-config tool  
<http://www.ibm.com/services/econfig/announce/index.htm>
- ▶ IBM Big Data infrastructure  
<https://www.ibm.com/marketplace/cloud/big-data-infrastructure/us/en-us>
- ▶ IBM Data Engine for Hadoop and Spark - Power Systems Edition  
<http://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POL03246USEN>
- ▶ IBM Fix Central  
<https://www.ibm.com/support/fixcentral/>
- ▶ IBM Spectrum Computing resource scheduler  
<http://ibm.co/1TKU1Mg>
- ▶ IBM Systems Lab Services  
<http://www.ibm.com/systems/services/labservices/>

## Help from IBM

IBM Support and downloads

[ibm.com/support](https://ibm.com/support)

IBM Global Services

[ibm.com/services](https://ibm.com/services)

**Redbooks**

**IBM Data Engine for Hadoop and Spark**

(0.2"spine)  
0.17"->0.473"  
90->249 pages







SG24-8359-00

ISBN 0738441937

Printed in U.S.A.

Get connected

