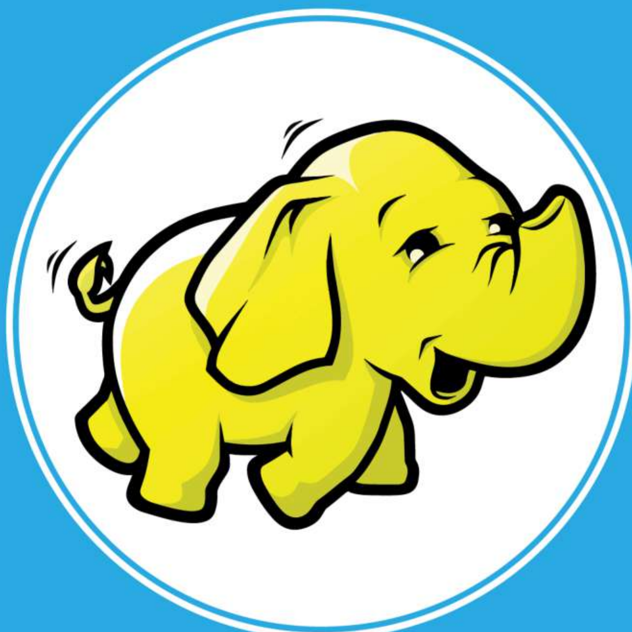
simplilearn

Big Data Hadoop Developer

Certification Examination Preparation Guide



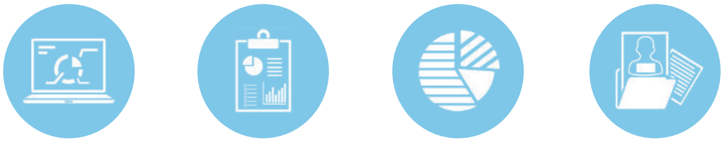
Global Leader in Professional Certification Training

www.simplilearn.com

support@simplilearn.com Live Chat 24/7 on simplilearn.com

Big Data and Hadoop Developer

Certification Training



Authored & published by

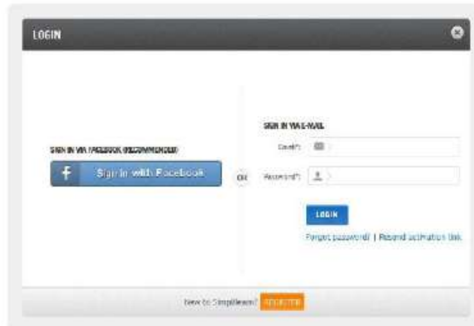
Simplilearn

Version 1.0

Notice

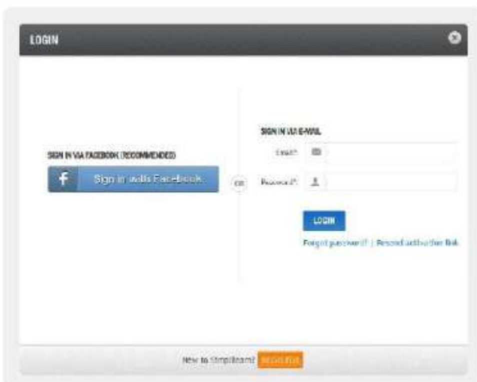
How to register Online

1



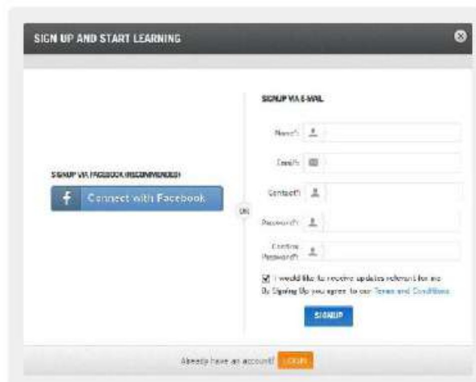
Go to www.simplilearn.com. If you have previously registered, please log in using the "Login" option at the top-right area on the screen. Please enter your registered email ID & password to access online or self-study material at any time.

2



If you haven't registered already, please register using the 'Register' button at the bottom of the "Login screen".

3



When you register for the first time, you will need to enter basic profile details. This information will be saved for future use and can be edited. Once you have entered all the necessary information, click "Signup" at the bottom of the page.

4

Once registered, you can edit your profile, access self-study material, take the exam, see your exam results & register for a new course by entering your login information using the "Login" option at the top-right area of the screen.

Thank you for registering. Have a great class and all the best for your exam!

Icons Used in This Book

The following icons are used across the book to highlight specific points.

eLearning

Refers to the eLearning course available on the Simplilearn.com site for a detailed explanation of the concepts. This icon will be present on each page of the handbook indicating the need to study the eLearning course content along with the classroom content.

Real-Life Examples

Highlights examples relevant to specific content. This icon will be present when a concept is explained with the help of a real life example.

Quiz

Refers to the questions at the end of every lesson. This icon will indicate the start of Quiz questions.

Tips

Indicates some important advice or point mentioned on the page that one should remember.

Notes

Use this section to add any key takeaways from the class for your future reference. This icon will be present on each page of the handbook.

IT Service Management—Best Practices

simplilearn

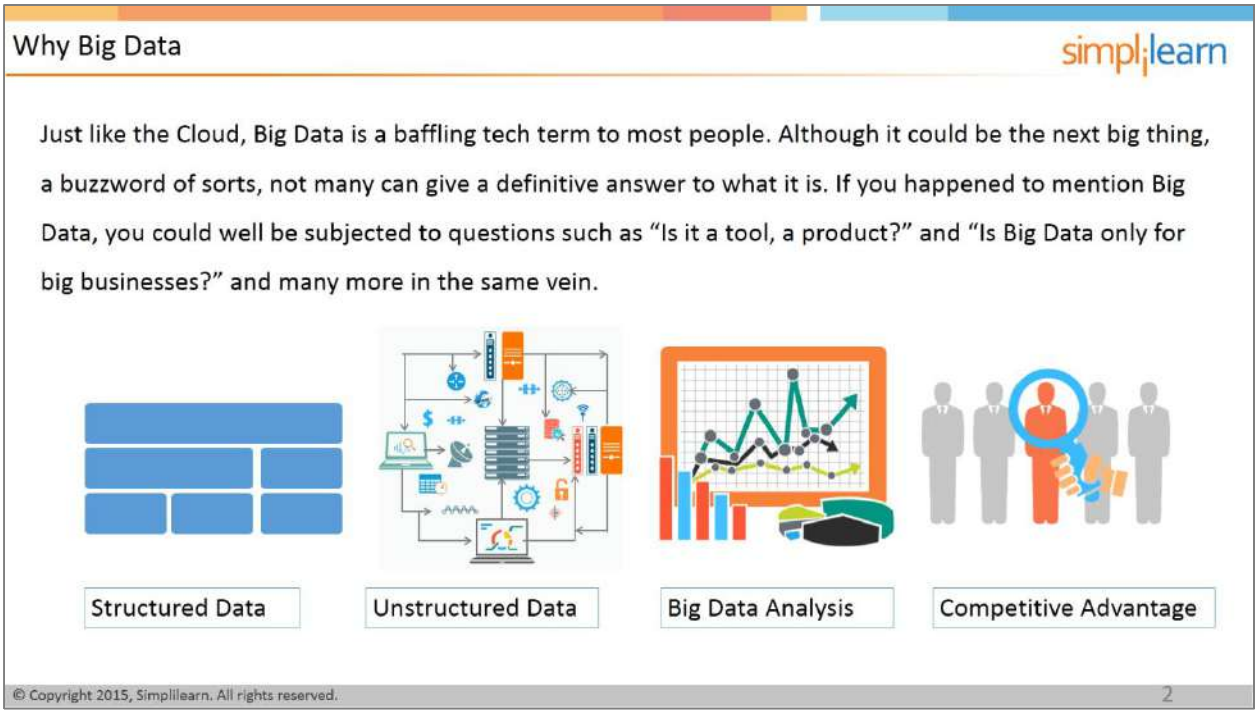
IT Service Management is the implementation and management of quality IT services that meet the needs of a business. It includes best practices followed by many organisations. The sources of best practices are:

- Existing public standards published by the International Standards Organisation or ISO;
- Industry practices that are shared among industry practitioners;
- Academic research; and
- Internal experiences or an organisation's past experiences in providing similar services.

-3-

Table of Contents

Lesson 0—Course Introduction



Why Big Data

Just like the Cloud, Big Data is a baffling tech term to most people. Although it could be the next big thing, a buzzword of sorts, not many can give a definitive answer to what it is.

If you happened to mention Big Data, you could well be subjected to questions such as “Is it a tool, a product?” and “Is Big Data only for big businesses?” and many more in the same vein.

But what is Big Data? It is just the huge amounts of data, both structured and unstructured, a business has to deal with on a daily basis.

That being said, why is it important to a business, or for that matter, you?

To use this phenomenal amount of data to its advantage, a business must fully grasp the possibilities of analyzing Big Data and thereby making informed decisions to stay ahead of the curve.

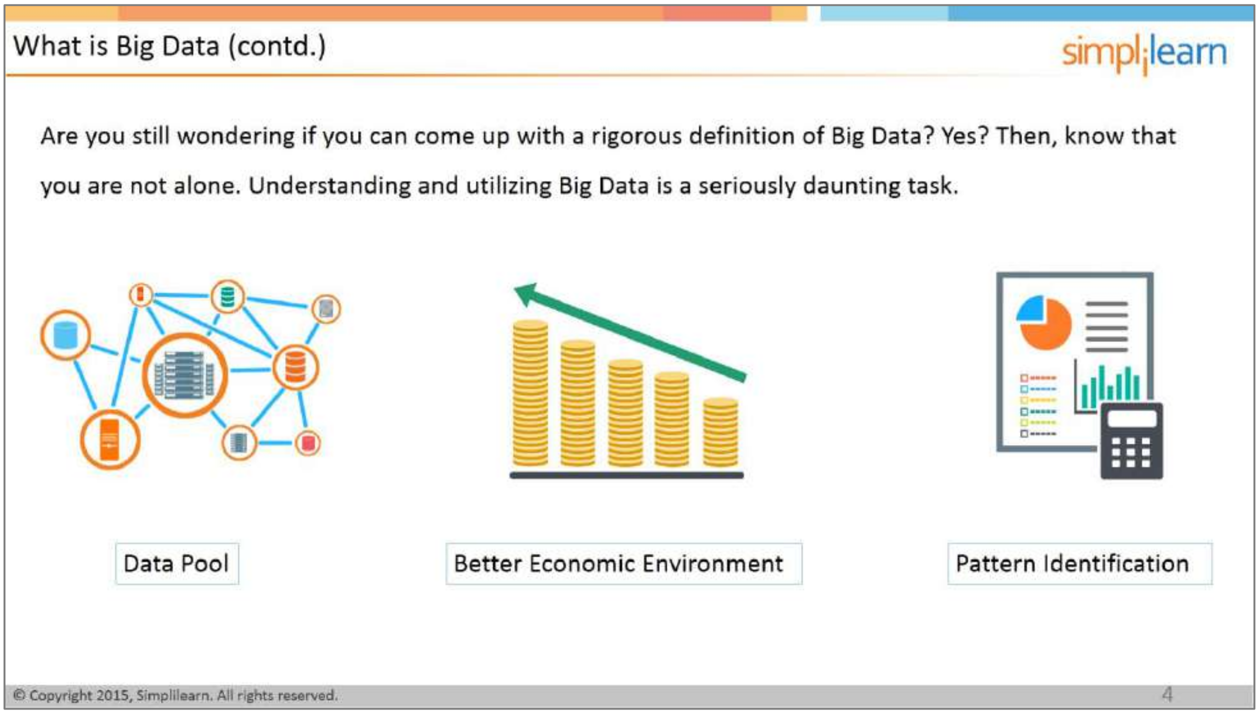
There has to be a reason why many analysts are calling Big Data a new competitive advantage, right?

What is Big Data

Let’s look at Gartner's “3Vs” model to define Big Data.

According to the American IT research and advisory firm Gartner Inc., “Big Data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.”

Today, the size (volume), complexity (variety), and the rate of growth (velocity) of the data that organizations handle have reached such unbelievable levels that traditional processing and analytical tools fail quite miserably.

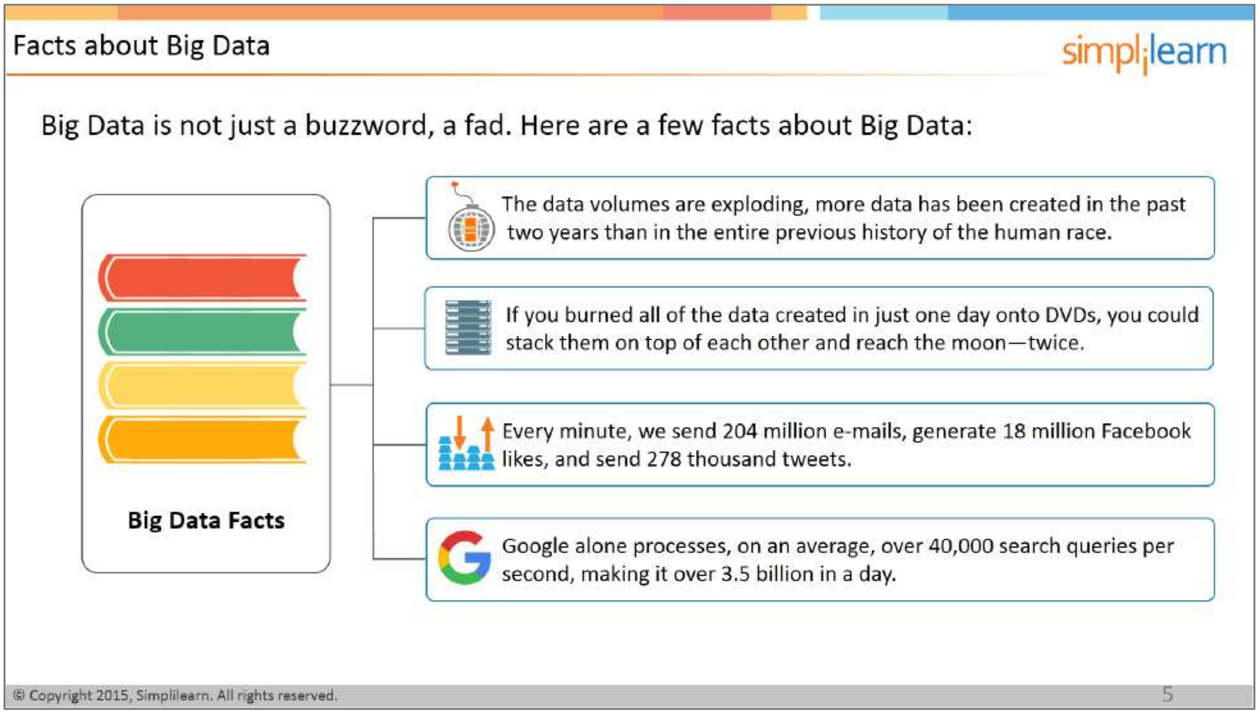


What is Big Data (contd.)

Are you still wondering if you can come up with a rigorous definition of Big Data? Yes? Then, know that you are not alone. Though people may be divided on how best to define Big Data, few are likely to dispute that these large pools of data are integral to improved future prospects.

But what is certain is that understanding and utilizing Big Data is a seriously daunting task.

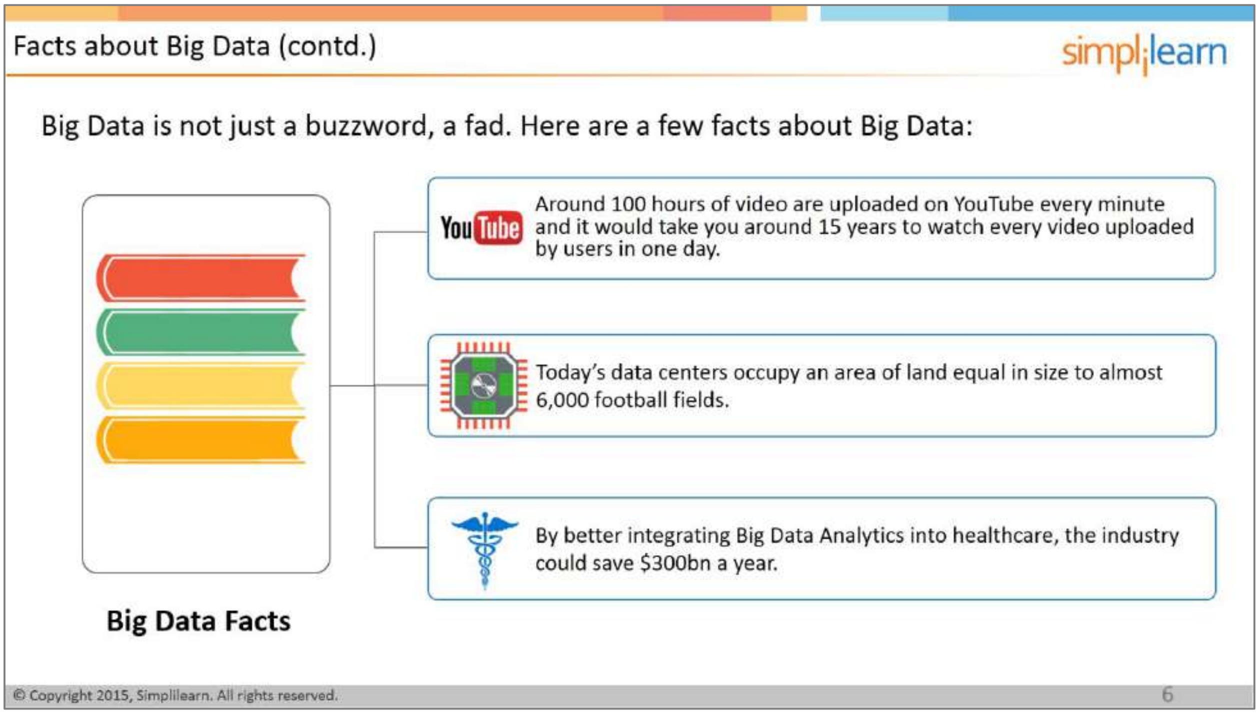
The sheer volume of data that is captured and analyzed will pave way for a better economic environment only if it is effectively used to discern patterns that enable better decision making.



Facts about Big Data

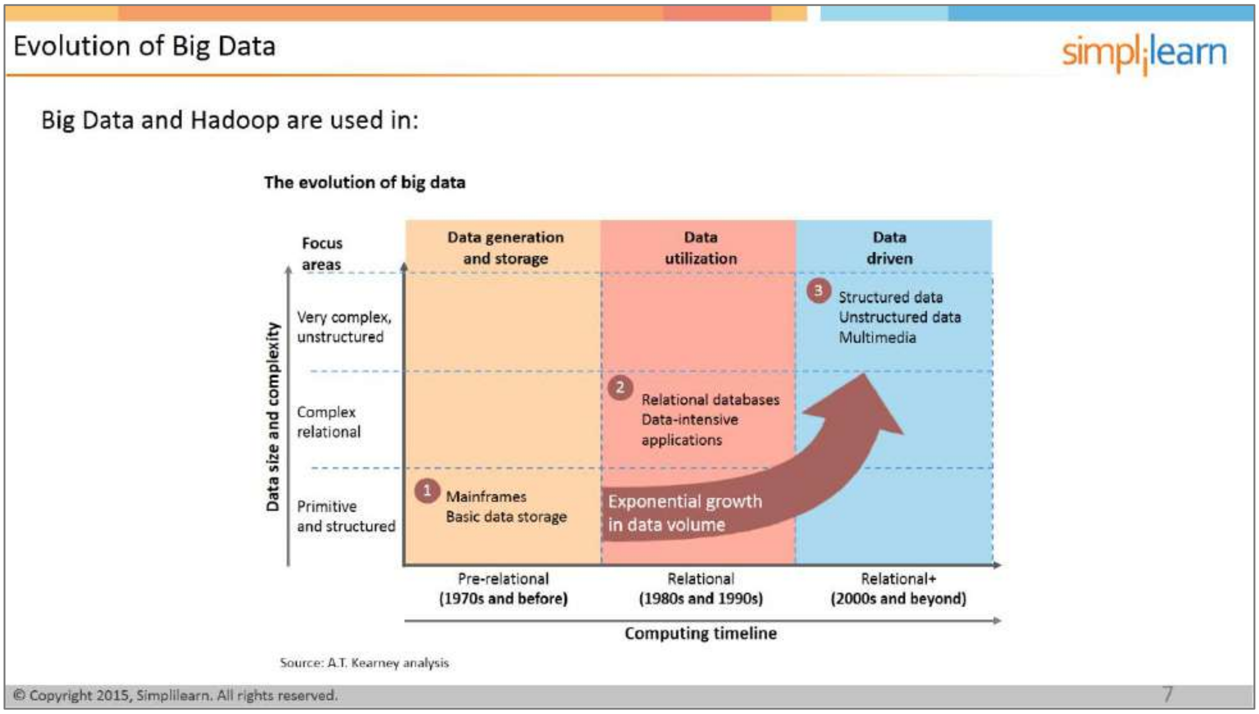
Big Data is not just a buzzword, a fad. It is something that will undoubtedly touch businesses and people's lives world over. For people who think that this is just another term that only the tech industry seems to care about, here are a few facts about Big Data that will likely convince them that Big Data is a revolution in the making! Read on, skeptics!

- The data volumes are exploding, more data has been created in the past two years than in the entire previous history of the human race.
- If you burned all of the data created in just one day onto DVDs, you could stack them on top of each other and reach the moon—twice.
- Every minute, we send 204 million e-mails, generate 18 million Facebook likes, and send 278 thousand tweets.
- Google alone processes, on an average, over 40,000 search queries per second, making it over 3.5 billion in a day.



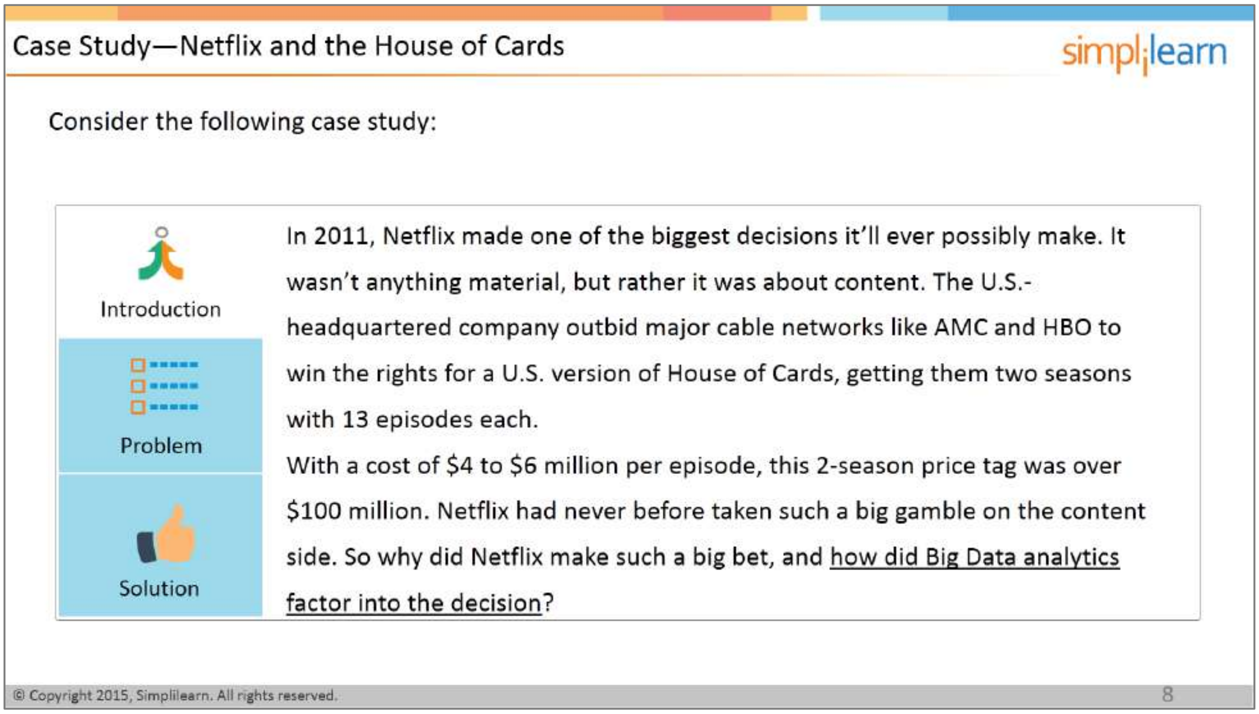
Facts about Big Data (contd.)

- Around 100 hours of video are uploaded on YouTube every minute and it would take you around 15 years to watch every video uploaded by users in one day.
- Today's data centers occupy an area of land equal in size to almost 6,000 football fields.
- By better integrating Big Data Analytics into healthcare, the industry could save $300bn a year—That is equal to reducing the healthcare costs of every man, woman, and child by $1,000 a year. (Source: https://newpush.com/category/big-data/)



Evolution of Big Data

The image gives a very real picture of the evolution of Big Data. Here, the data size and complexity have been mapped against computing timeline. In early 1970s, in fact even before that, the focus was on generation and storage of data. Mainframes were used for basic data storage. During the 1980s and 90s, the focus was on data utilization, for which relational databases and other data-intensive applications were brought into the picture. In the 2000s the spotlight was on data-driven technology, for which structured data, unstructured data, and multimedia were used. Throughout this evolution phase, we saw an exponential growth in data volume.

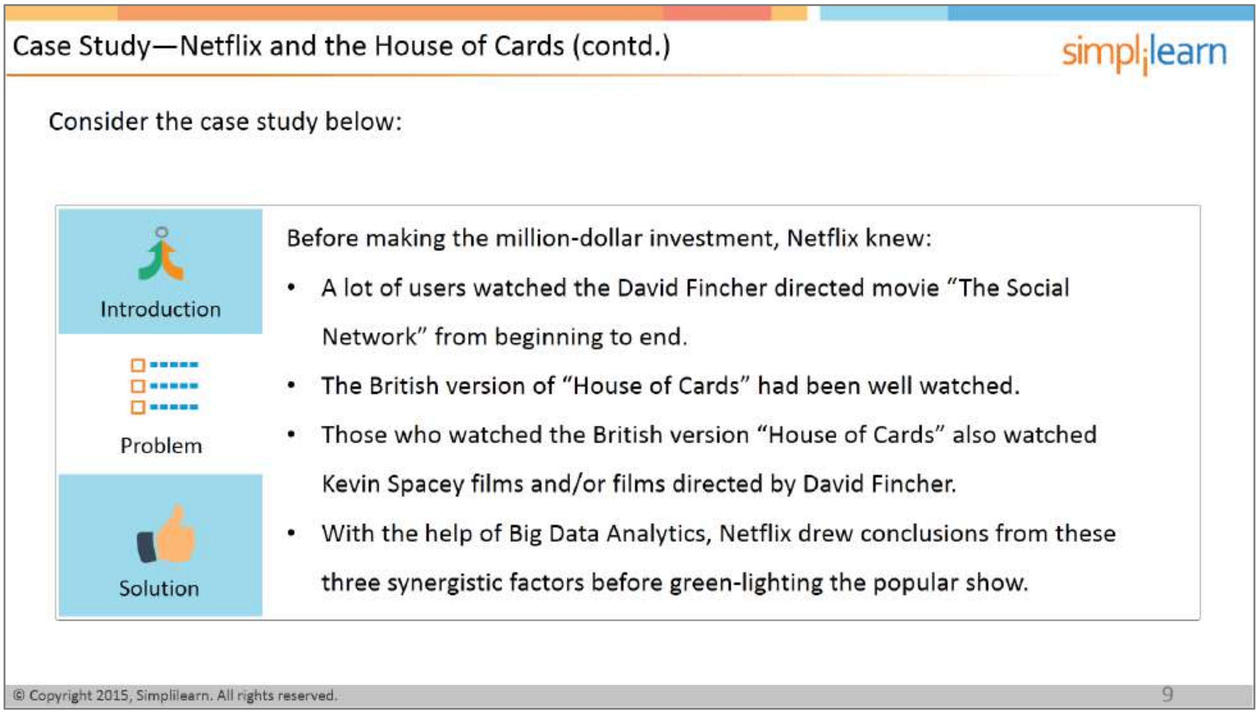


Case Study – Netflix and the House of Cards

Consider the following case study:

In 2011, Netflix made one of the biggest decisions it'll ever possibly make. It wasn't anything material, but rather it was about content. The U.S.-headquartered company outbid major cable networks like AMC and HBO to win the rights for a U.S. version of House of Cards, getting them two seasons with 13 episodes each.

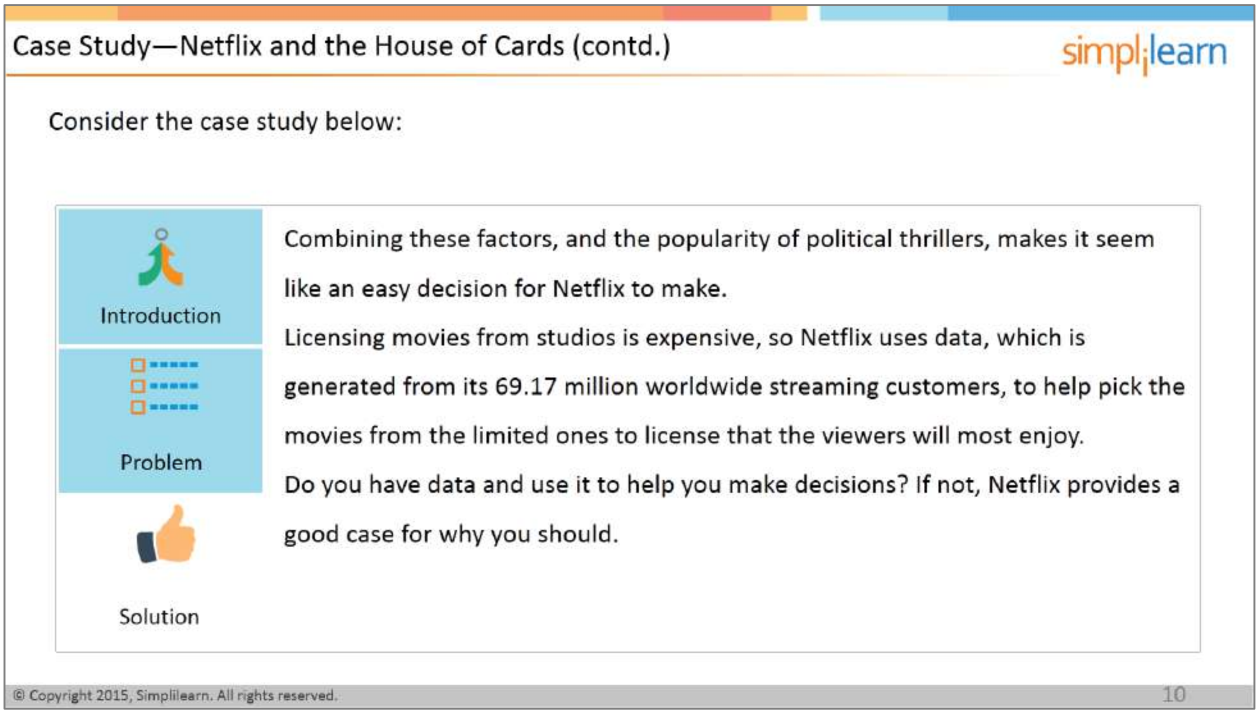
With a cost of $4 to $6 million per episode, this 2-season price tag was over $100 million. Netflix had never before taken such a big gamble on the content side. So why did Netflix make such a big bet, and how did Big Data analytics factor into the decision? Let's find out.



Case Study—Netflix and the House of Cards (contd.)

Before making the million-dollar investment, Netflix knew:

- A lot of users watched the David Fincher directed movie "The Social Network" from beginning to end.
- The British version of "House of Cards" had been well watched.
- Those who watched the British version "House of Cards" also watched Kevin Spacey films and/or films directed by David Fincher.
- With the help of Big Data Analytics, Netflix drew conclusions from these three synergistic factors before green-lighting the popular show.

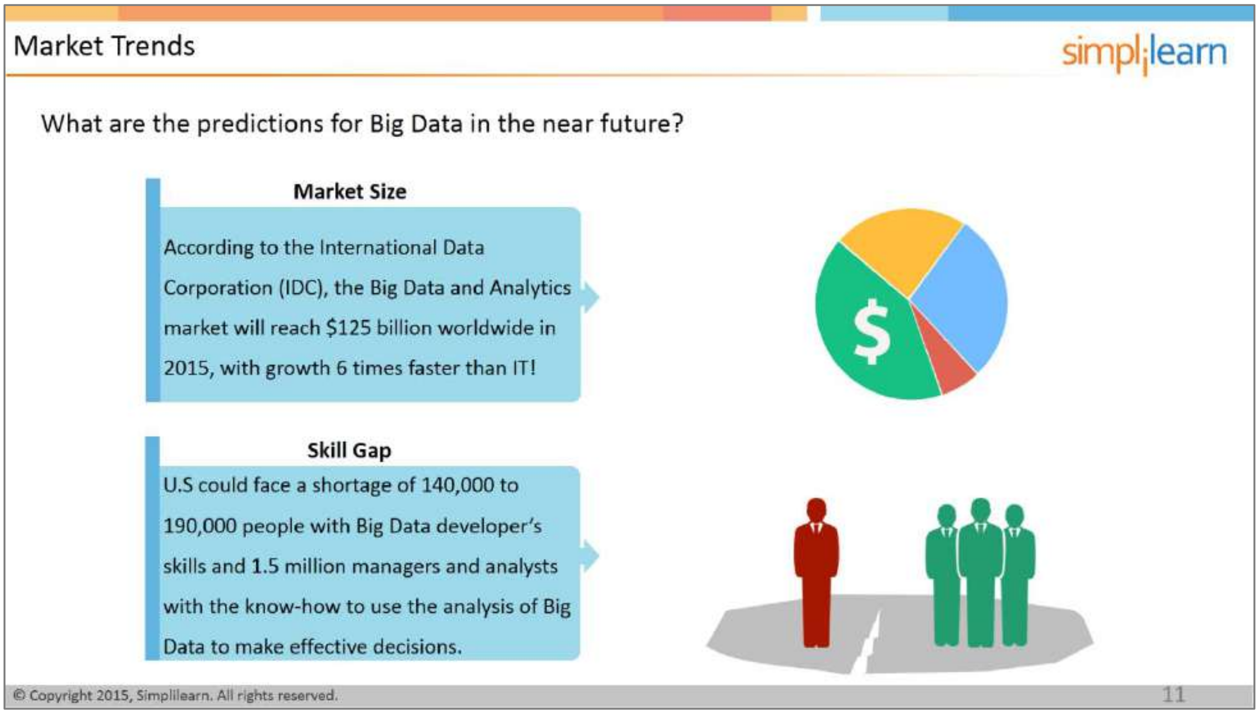


Case Study—Netflix and the House of Cards (contd.)

Combining these factors, and the popularity of political thrillers, makes it seem like an easy decision for Netflix to make.

Licensing movies from studios is expensive, so Netflix uses data, which is generated from its 69.17 million worldwide streaming customers, to help pick the movies from the limited ones to license that the viewers will most enjoy.

Do you have data and use it to help you make decisions? If not, Netflix provides a good case for why you should.



Market Trends

So what are the predictions for Big Data in the near future?

Market Size

According to the International Data Corporation (IDC), the Big Data and Analytics market will reach $125 billion worldwide in 2015, with growth 6 times faster than IT!

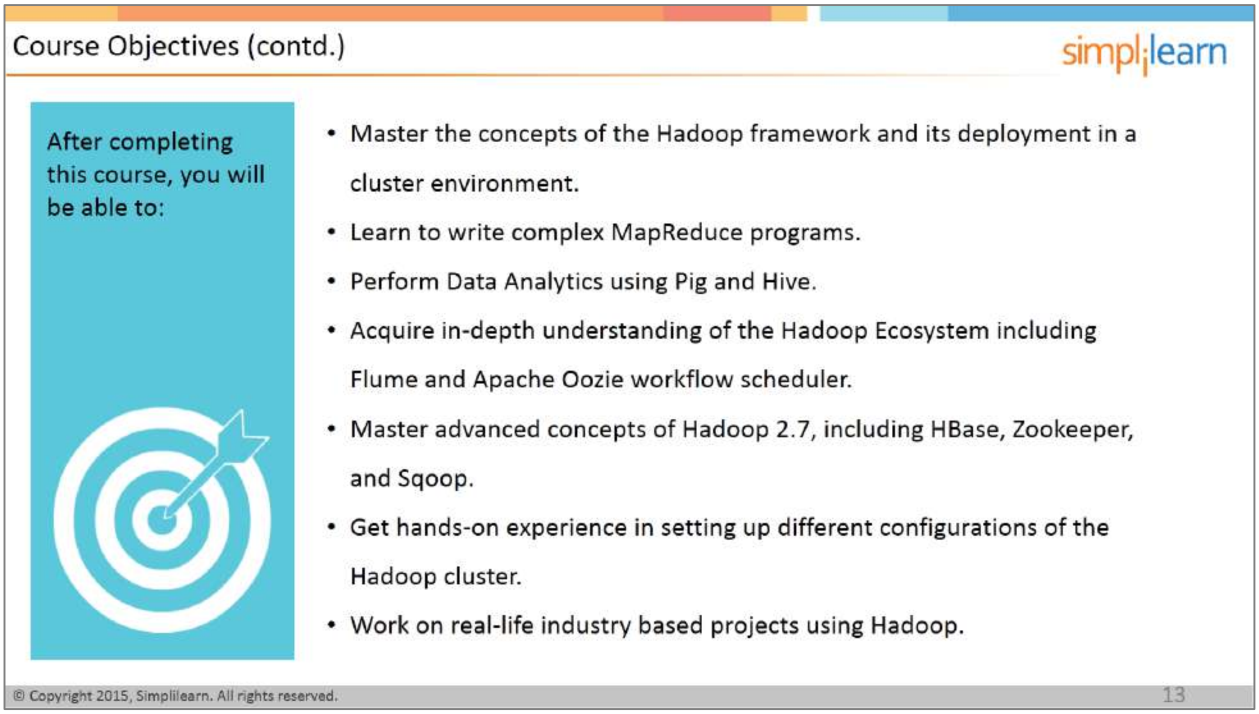
Skill Gap

According to the McKinsey Global Institute, by 2018, the United States alone could face a shortage of 140,000 to 190,000 people with Big Data developer's skills as well as 1.5 million managers and analysts with the know-how to use the analysis of Big Data to make effective decisions.

Buying and selling data will soon become the new bread and butter for business across the globe!

Course Objectives

Big Data and Hadoop Certification Training from Simplilearn is designed to ensure that you are job ready to take up an assignment in Big Data. This training not only equips you with essential Hadoop skills, but also gives you the required work experience in Big Data Analytics via implementation of real-life industry projects that span three months.



Course Objectives (contd.)

After completing this course, you will be able to:

- Master the concepts of the Hadoop framework and its deployment in a cluster environment.
- Learn to write complex MapReduce programs.
- Perform Data Analytics using Pig and Hive.
- Acquire in-depth understanding of the Hadoop Ecosystem including Flume and Apache Oozie workflow scheduler.
- Master advanced concepts of Hadoop 2.7, including HBase, Zookeeper, and Sqoop.
- Get hands-on experience in setting up different configurations of the Hadoop cluster.
- Work on real-life industry based projects using Hadoop.

Course Details

The course flow is as follows:

First, internalize the domains, which includes

- Introduction to Big Data and Hadoop,
- Concepts and architecture of Big Data tools,
- Installation and configuration of Hadoop components such as MapReduce, YARN, HDFS, Pig, Hive, HBase, Zookeeper, Sqoop, and Flume, and
- Hadoop Administration, Troubleshooting, and Security.

The next step is to master the components, which includes Hadoop 2.7, Yarn, MapReduce, HDFS, Pig, Hive, HBase, Zookeeper, Sqoop, and Flume.

Next, you will learn to manage the following: structured, unstructured, and semi structured data; Hadoop ecosystem and complex business problems; deployment of Hadoop cluster; and writing complex MapReduce programs.

Simplilearn provides you 12 case studies on different components of Hadoop. These case studies are specifically designed to ensure that you are able to grasp the important concepts of this course without any difficulty.

You will also have a chance to work on a full-fledged real-world project of your choice from various industries such as insurance, social media, and education of 60 hours with 3.5 Billion data points. The purpose of this project is to make you confidant and job ready once you successfully complete the course.

Once these steps are covered, you are ready to take the last step, which is to get certified. This course will facilitate you to achieve Big Data Hadoop Developer certification with Hadoop 2.7 after successfully completing a full scale industry project, and an objective test of 45 questions.

Project Submission and Certification

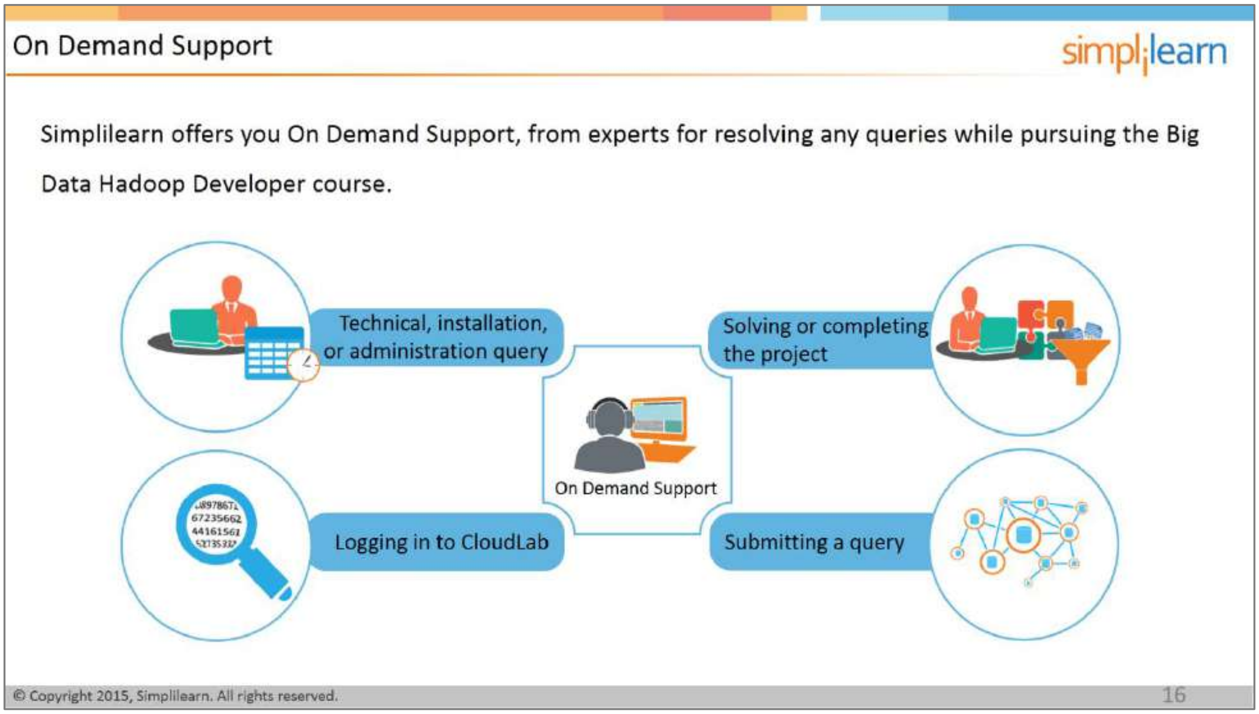
For Big Data and Hadoop Developer certification, it is mandatory to fulfill the following criteria:

1. Complete any one of the four projects provided by Simplilearn, within the Online Self Learning course access period. The completed project is then evaluated by the lead trainer. Screenshots of the final output and the source code used should be mailed to projectsubmission@simplilearn.com within the Online Self Learning course access period. For any queries or difficulties while solving projects, you can get assistance from our On Demand support. If you have subscribed for the Live Virtual Classroom training, you may attend any of ongoing batches of Big Data Hadoop Developer course for assistance if you face any issues while implementing the project.

Project Submission and Certification (contd.)

2. A minimum score of 80% is mandatory for clearing the online examination. If you are not able to clear the online exam at the first attempt, you can attempt the exam once more. At the end of the course, you will receive an experience certificate showcasing your experience of 3 months in implementing Big Data and Hadoop Projects.

Note: It is mandatory that you fulfill both the criteria, to become a Certified Big Data and Hadoop Developer.



On Demand Support

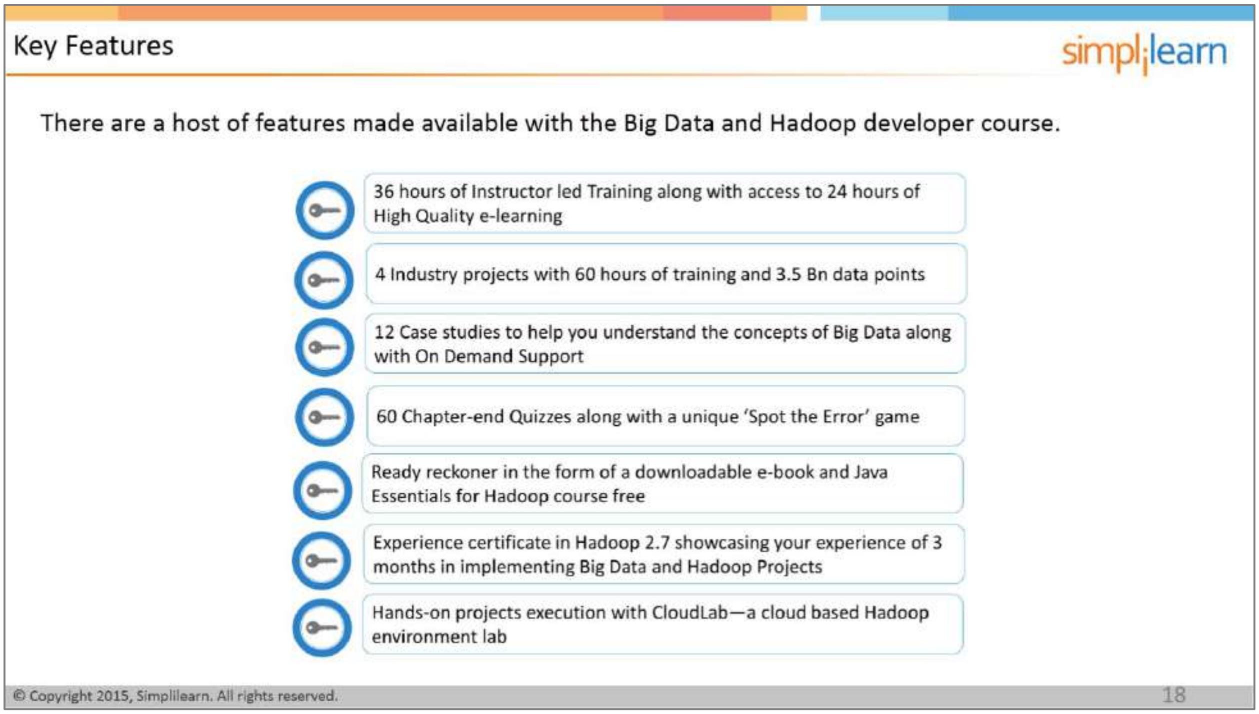
Simplilearn offers you On Demand Support, from experts for resolving any queries while pursuing the Big Data Hadoop Developer course.

Do you have a query on technical, installation, or administration? Our support team will work with you for a solution. What's more? In case of critical issues, support will be rendered through remote desktop.

Are you unsure about how to solve or complete the project? Or maybe you are stuck on a program and not clear on how to get the desired output. Our On Demand Support team will guide you through the programming in projects and case studies of the course.

Is this the first time you have logged in to CloudLab? Do you need a little help on how to work your way around a project? Our team will assist you in all your CloudLab queries on the execution of projects, case studies, and exercises.

Want to submit a query through On Demand Support? Simple, just visit www.simplilearn.com and click on Help and Support at the bottom of the page. You also have the option of Simplitalk and Live Chat and our On Demand Support will resolve the query within 48 hours with assistance.



Key Features

So, excited to know what all you get access to with the Big Data and Hadoop developer course?

There are a host of features made available to you!

You get 36 hours of Instructor led Training along with access to 24 hours of High Quality e-learning

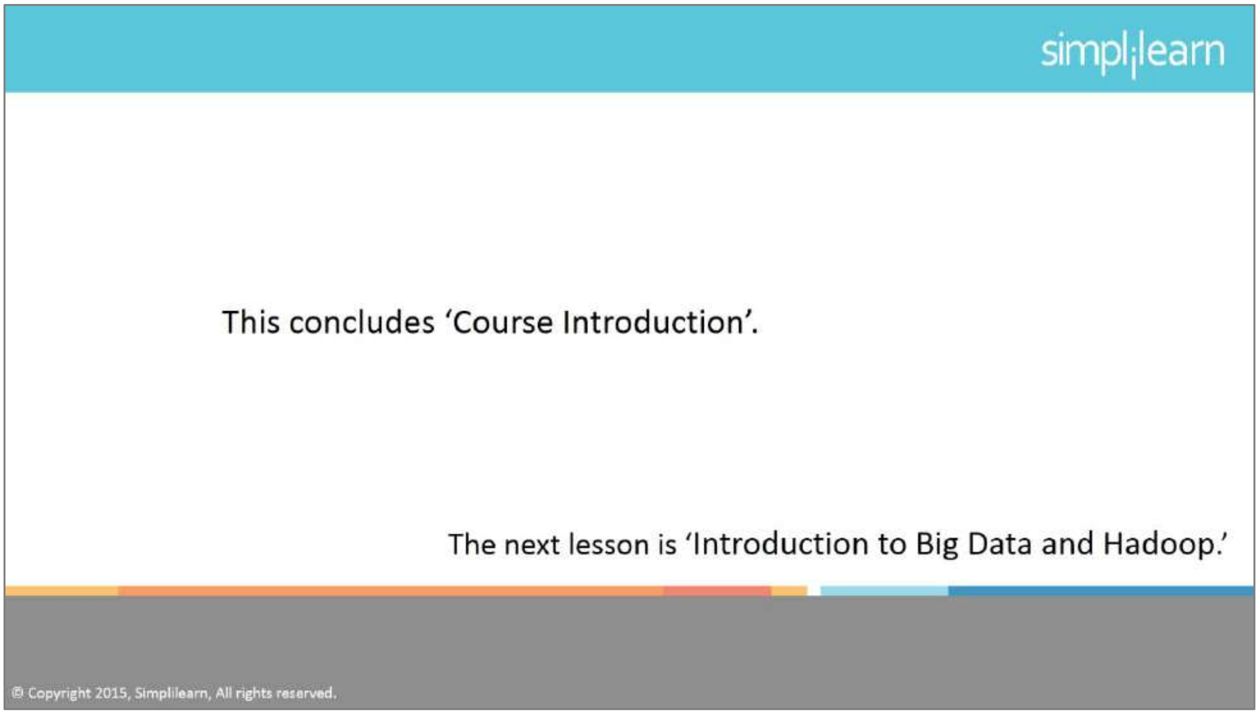
There are 4 Industry projects with 60 hours of training and 3.5 Bn data points that you will be working on in addition to 12 Case studies to help you understand the concepts of Big Data along with On demand support to take care of any queries that you have!

To test your understanding of the concepts covered, we give you 60 Chapter-end Quizzes along with a unique spot the error game that makes learning more fun!

We also provide you a ready reckoner in the form of a downloadable e-book.

What's more? You also get the Java Essentials for Hadoop course free!

Once you complete the course successfully, we award you with an experience certificate in Hadoop 2.7 showcasing your experience of 3 months in implementing Big Data and Hadoop Projects.



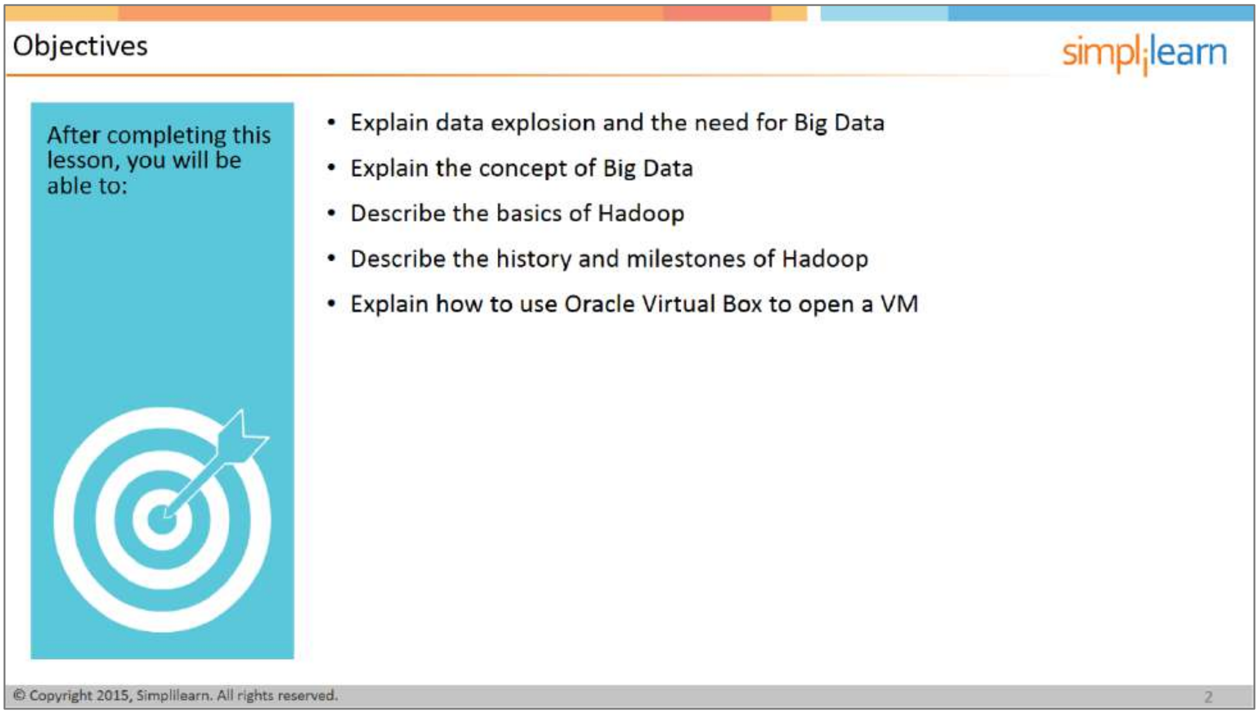
Thank You

This concludes the "Course Introduction."

The next lesson is "Introduction to Big Data and Hadoop."

Wishing you all the best for this course. Happy Learning!

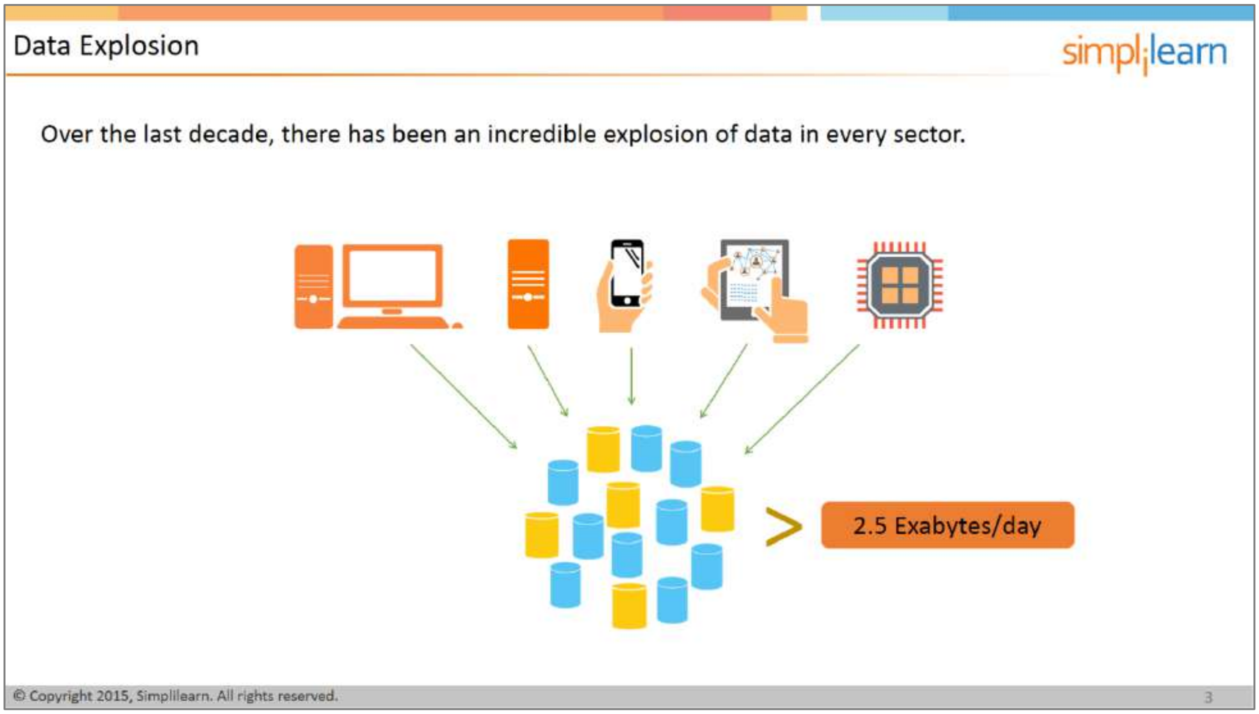
Lesson 1—Introduction to Big Data and Hadoop



Objectives

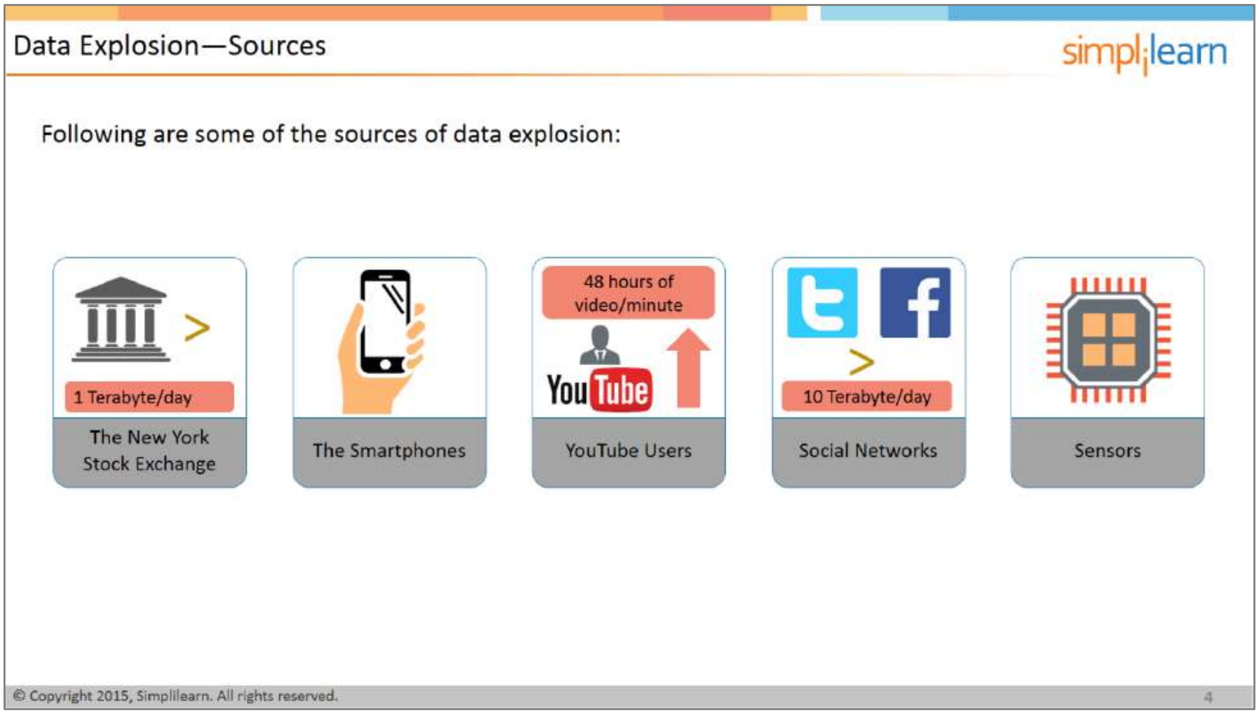
After completing this lesson, you will be able to:

- Explain data explosion and the need for Big Data
- Explain the concept of Big Data
- Describe the basics of Hadoop
- Describe the history and milestones of Hadoop
- Explain how to use Oracle Virtual Box to open a VM



Data Explosion

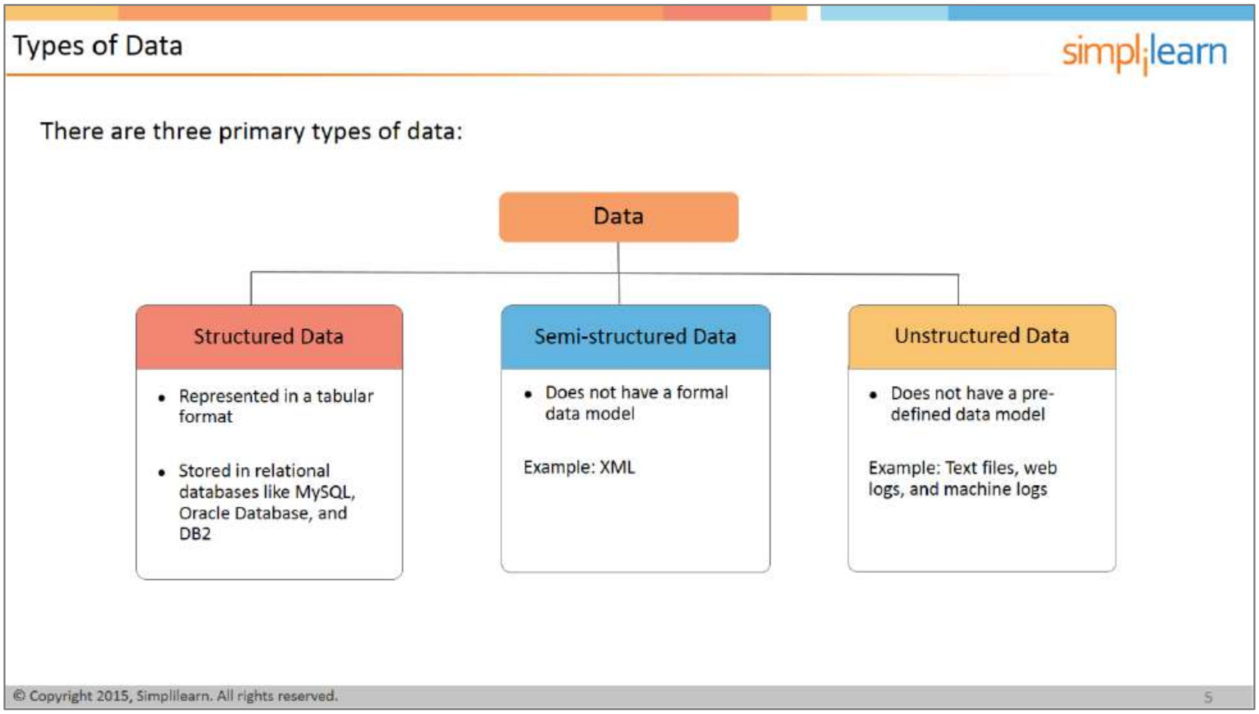
Over the last decade, there has been an incredible explosion of data in every sector. It is estimated that over 2.5 exabyte or 2.5 billion gigabytes of data is generated daily from various sources.



Data Explosion—Sources

Following are some of the sources of data explosion:

- A large stock exchange such as the New York Stock Exchange generates more than 1 terabyte of data daily.
- Worldwide, there are approximately 5 billion mobile phones in use; of these, nearly 1.75 billion are estimated to be smartphones.
- YouTube users upload more than 48 hours of video every minute. Every second of HD video generates bytes 2,000 times more than that required to store a single page of text.
- Large social networks like Twitter and Facebook generate more than 10 terabytes of data daily.
- There are more than 30 million networked sensors in the world, and each of them transmits data continuously.



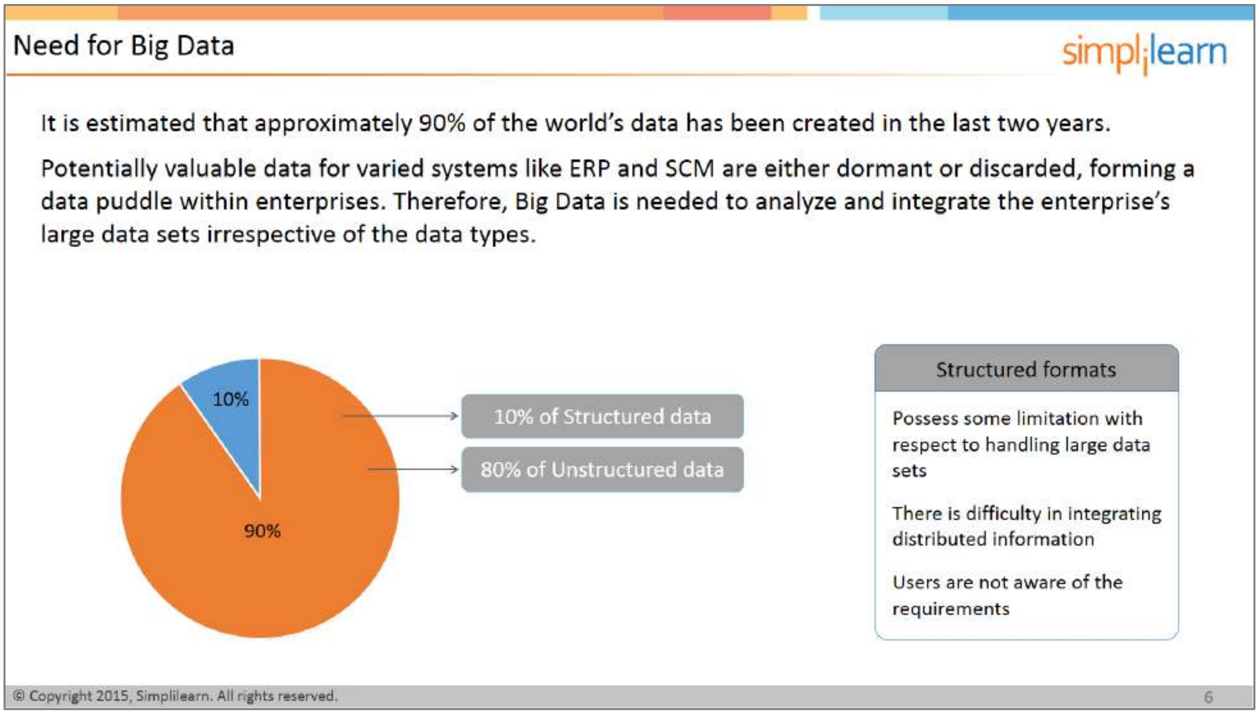
Types of Data

There are three primary types of data:

The first is structured data which can be represented in a tabular format and stored in relational databases such as MySQL, Oracle Database, and DB2.

The second is semi-structured data which does not have a formal data model, though it may have some semblance of a structure. For example, XML files represent semi-structured data.

The third is unstructured data, which does not have a pre-defined data model. Text files, web logs, and machine logs are examples of unstructured data.

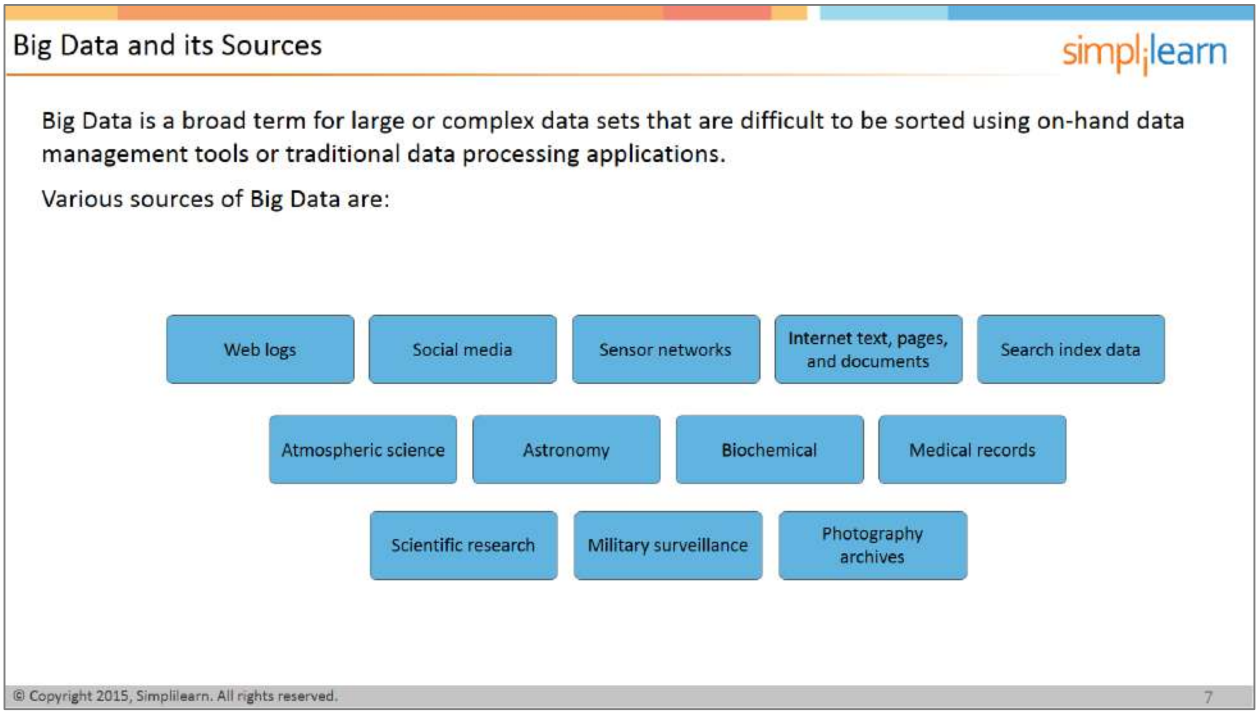


Need for Big Data

Did you know that according to an estimation approximately 90% of the world's data has been created in the last two years? It includes 10% of structured data and 80% of unstructured data that are difficult to analyze. The structured formats, such as databases, have limitations while handling large data sets and there is difficulty in integrating distributed information. Furthermore, most business users are not aware of the requirements during IT system development.

Potentially valuable data for varied systems like Enterprise Resource Planning or ERP and Supply Chain Management or SCM, are either dormant or discarded forming a data puddle within enterprises. It is often too expensive to justify the integration of large volumes of unstructured data.

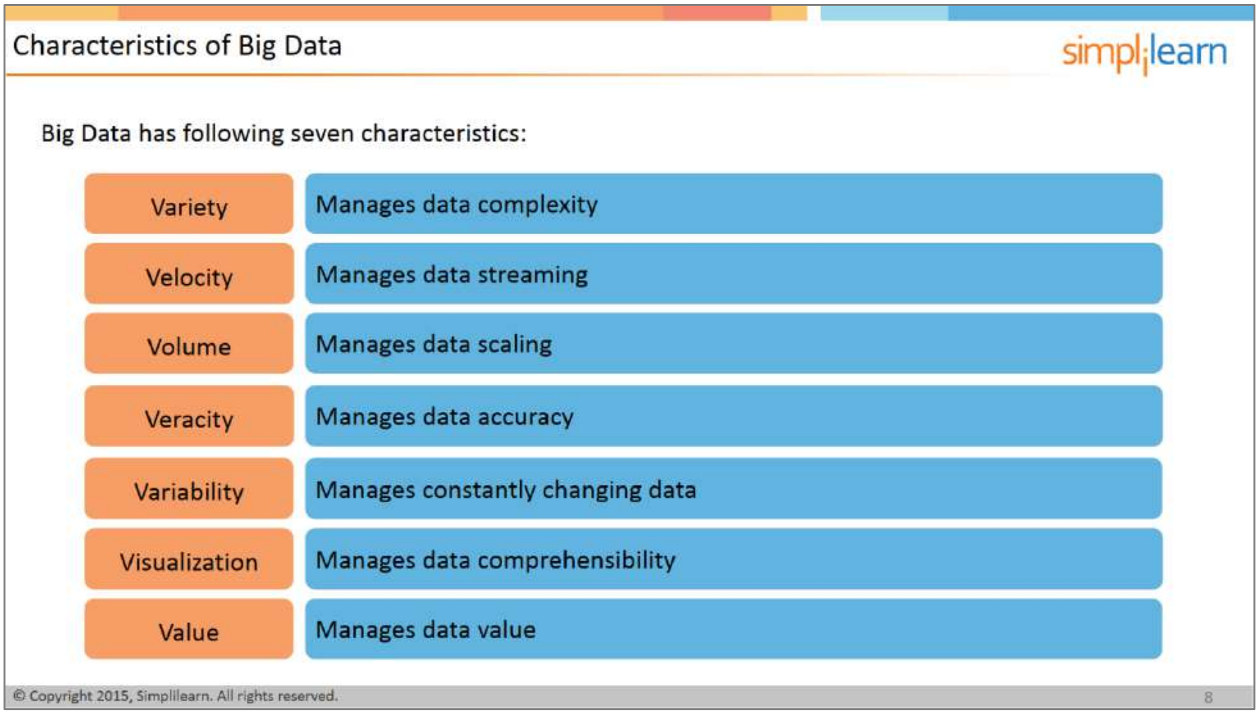
Therefore, Big Data is needed to analyze and integrate the enterprise's large data sets irrespective of the data types.



Big Data and its Sources

Big Data is a broad term for large or complex data sets that are difficult to be sorted using on-hand data management tools or traditional data processing applications...
Various sources of Big Data are:

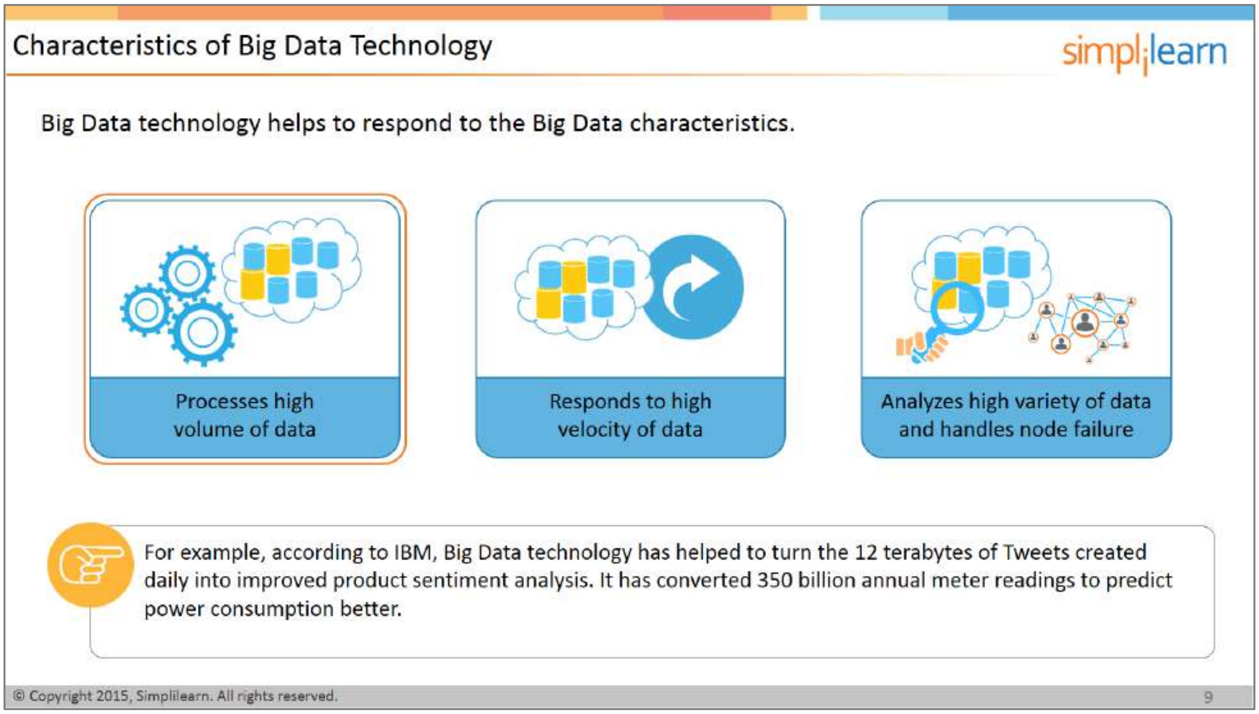
- web logs;
- social media;
- sensor networks;
- Internet text, pages, and documents;
- search index data;
- atmospheric science;
- astronomy;
- biochemical;
- medical records;
- scientific research;
- military surveillance; and
- photography archives.



Characteristics of Big Data

The characteristics of Big Data are as follows:

- Variety encompasses managing the complexity of data in different structures, ranging from relational data to logs and raw text.
- Velocity accounts for managing the streaming of data and the movement of voluminous data at a high speed.
- Volume denotes managing the huge scaling of data ranging from terabytes to zettabytes.
- Voluminous data is useless without accuracy, as incorrect data causes concern in organizations. Veracity manages accuracy of data and its analyses especially in automated decision-making process. Data veracity is essential if the organization aims to be information-centric.
- Big Data is variable, which means the meaning of the data is constantly changing. This makes it relevant in sentiment analyses. To perform a proper sentiment analyses, algorithms need to understand the context and decipher the exact meaning of a word in that context.
- Visualization makes voluminous data comprehensible. With correct analyses and visualizations, raw data can used. Visualizations mean complex graphs that can include variables of data while still remaining understandable and readable.
- The value of data is in analyses, which turns data into information, followed by knowledge. Based on data value, organizations can become information-centric.

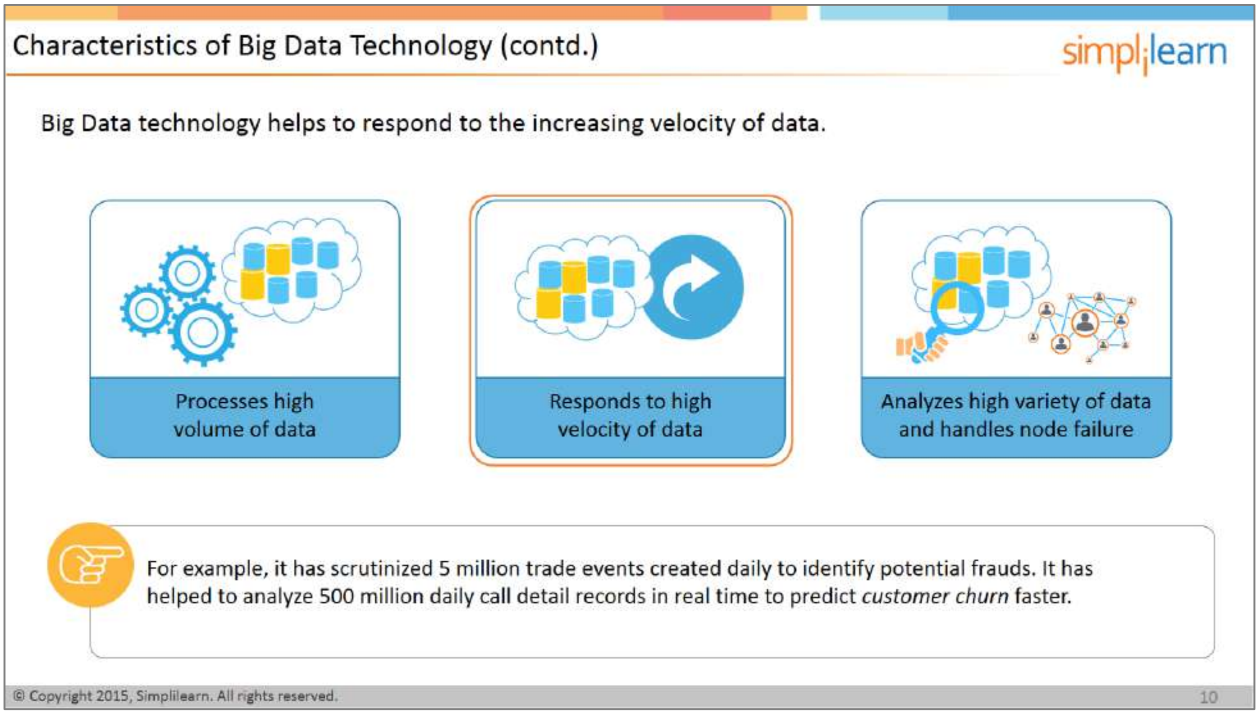


Characteristics of Big Data Technology

Big Data technology helps to respond to the Big Data characteristics. Following are the three characteristics of Big Data technology.

Firstly, it helps to cost-effectively process the growing volumes of data.

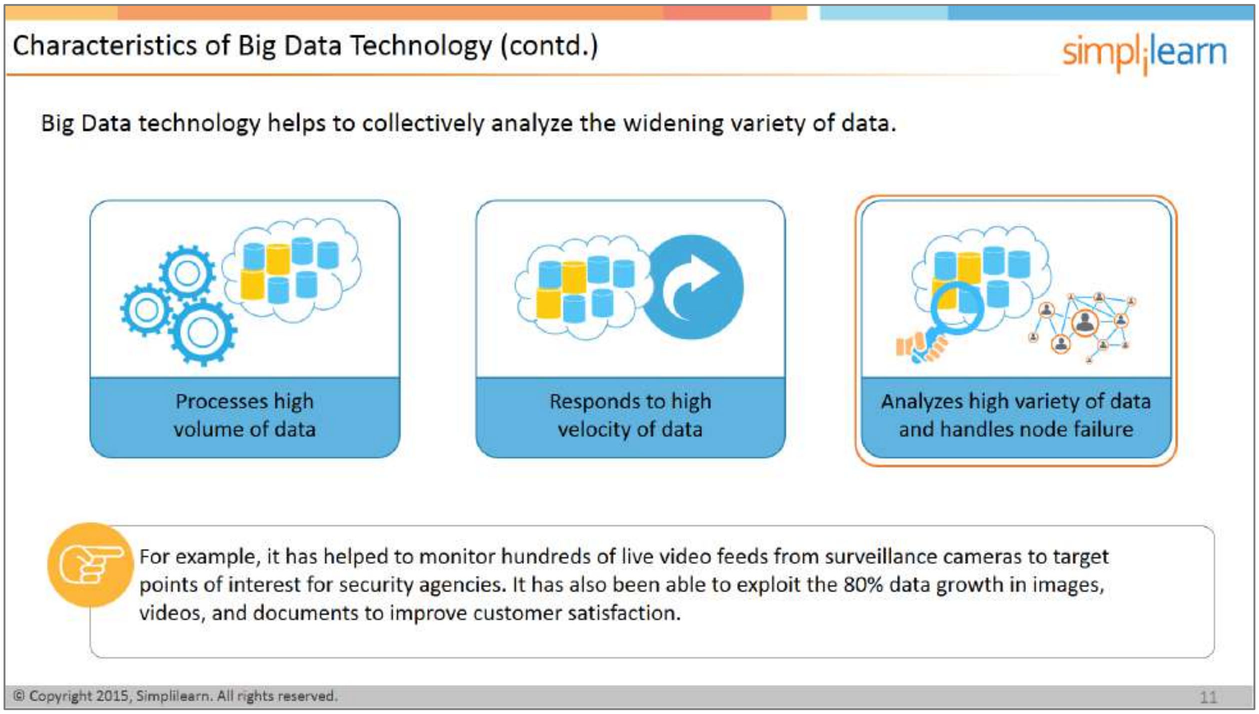
For example, per IBM, Big Data technology has helped to turn the 12 terabytes of tweets created daily into improved product sentiment analysis. It has converted 350 billion annual meter readings to predict power consumption better.



Characteristics of Big Data Technology (contd.)

The second characteristics of Big Data technology is that it helps to respond to the increasing velocity of data.

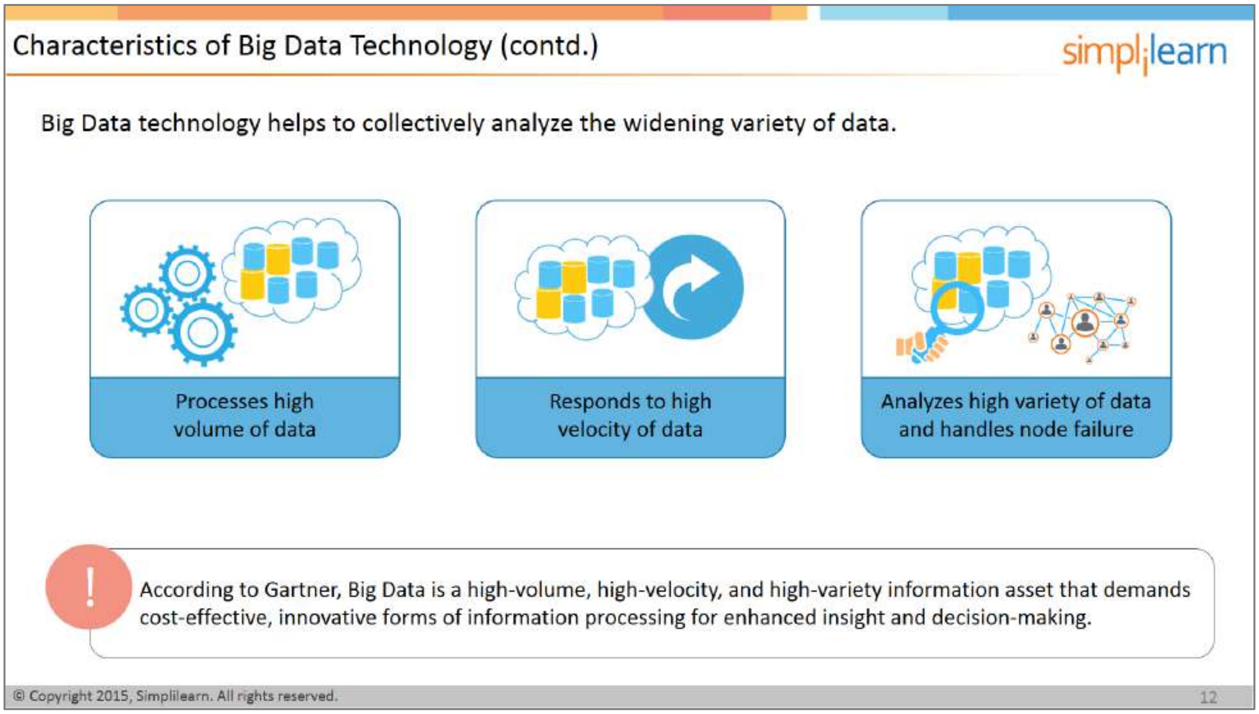
For example, it has scrutinized 5 million trade events created daily to identify potential frauds. It has helped to analyze 500 million daily call detail records in real time to predict *customer churn* faster.



Characteristics of Big Data Technology (contd.)

The third characteristics of Big Data technology is that it helps to collectively analyze the widening variety of data and handles failure of isolated nodes and tasks assigned to such nodes. It can also turn data into actionable insights.

For example, it has helped to monitor hundreds of live video feeds from surveillance cameras to target points of interest for security agencies. It has also been able to exploit the 80% data growth in images, videos, and documents to improve customer satisfaction.



Characteristics of Big Data Technology (contd.)

According to Gartner, Big Data is a high-volume, high-velocity, and high-variety information asset that demands cost-effective, innovative forms of information processing for enhanced insight and decision-making.

simplilearn

KNOWLEDGE CHECK 1

Why is Big Data technology popular and gaining attention among various enterprises?

a. Manages and processes high data in a cost-efficient manner

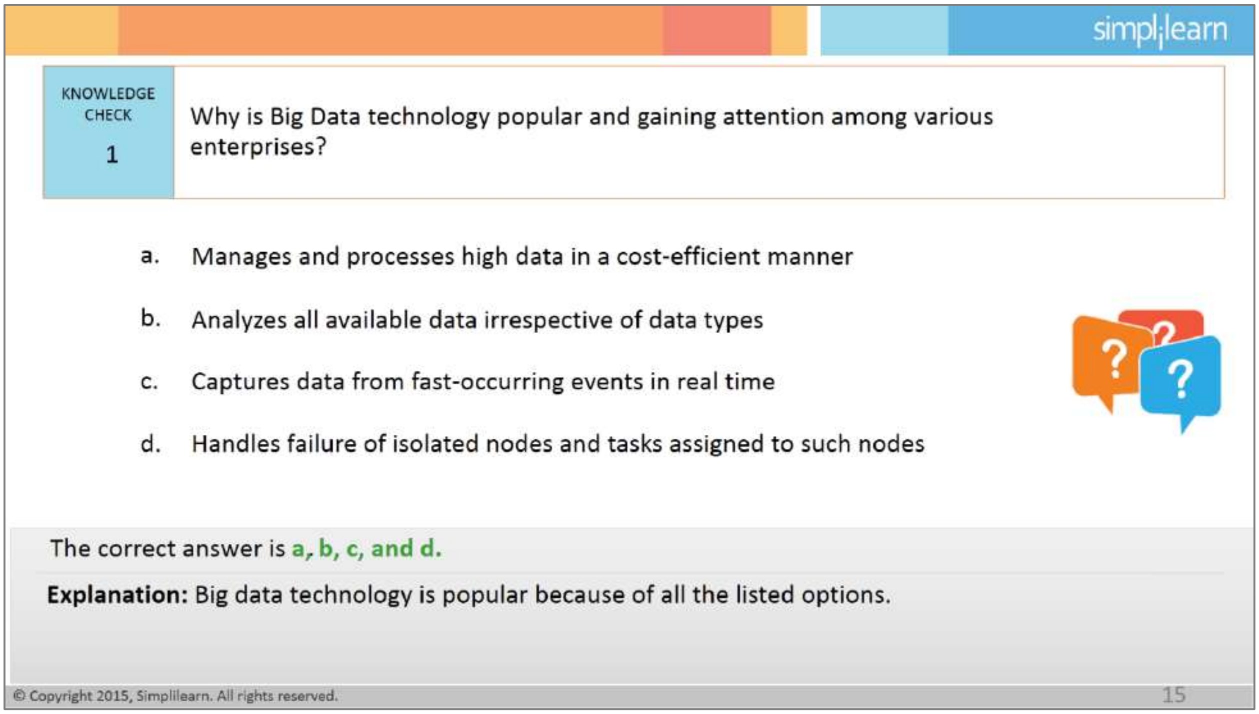
b. Analyzes all available data irrespective of data types

c. Captures data from fast-occurring events in real time

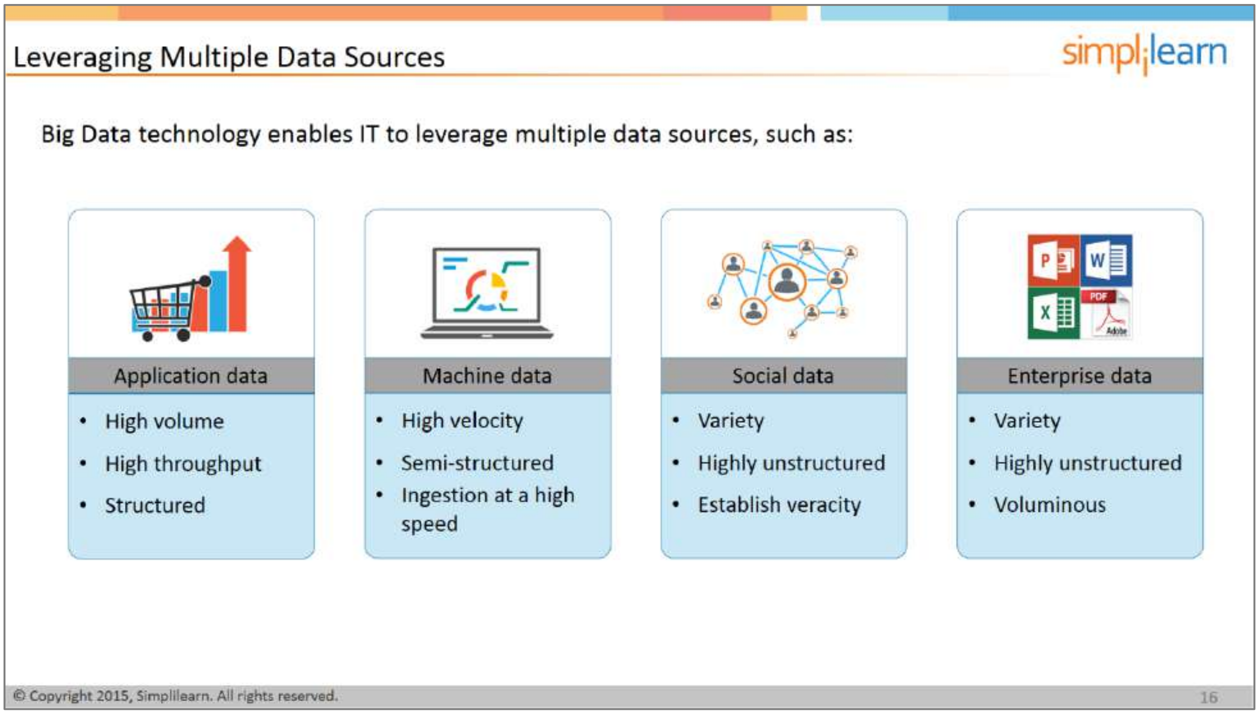
d. Handles failure of isolated nodes and tasks assigned to such nodes

 14

Knowledge Check



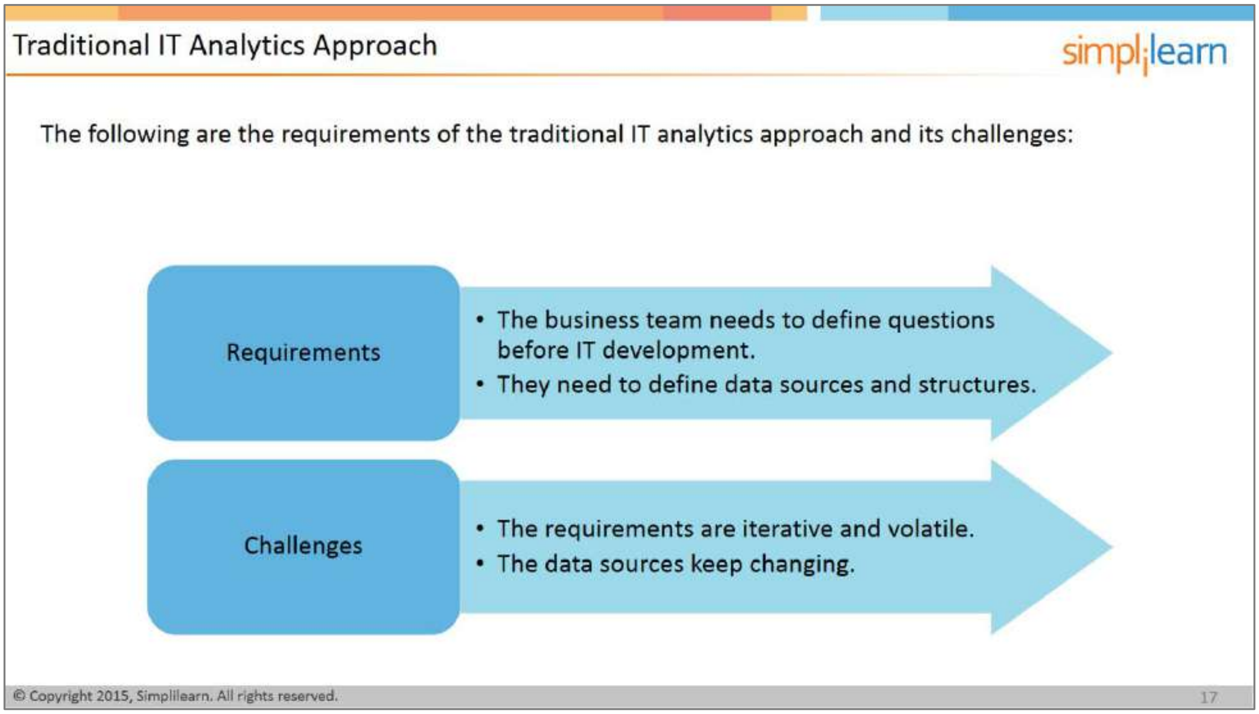
Knowledge Check—Answer



Leveraging Multiple Data Sources

Big Data technology enables IT to leverage multiple sources of data. The major sources of data include:

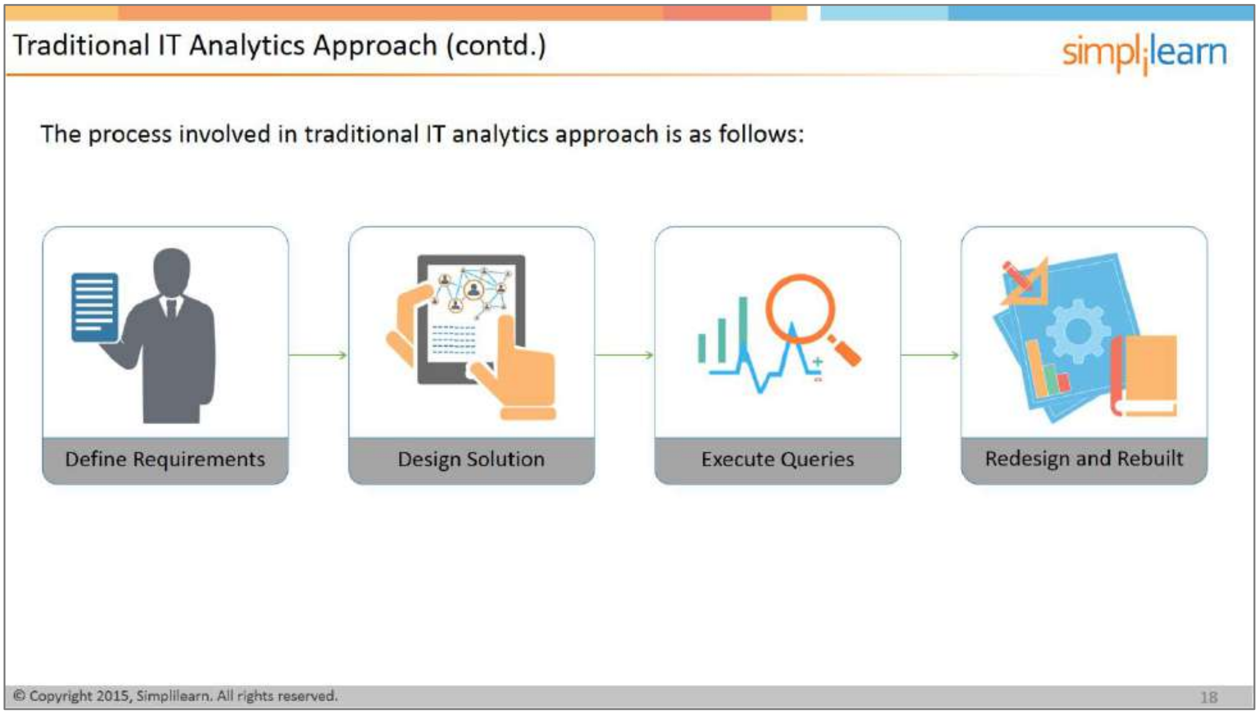
- Application data, which has high volume and throughput, and is structured.
- Machine data, which has high velocity, is semi-structured, and needs to be ingested at a high speed.
- Social data, which has high variety, is unstructured, and requires you to establish the veracity of the data.
- Enterprise data, which has variety, is available in different formats like pdf, spreadsheets, and documents. It is highly unstructured and can be voluminous.



Traditional IT Analytics Approach

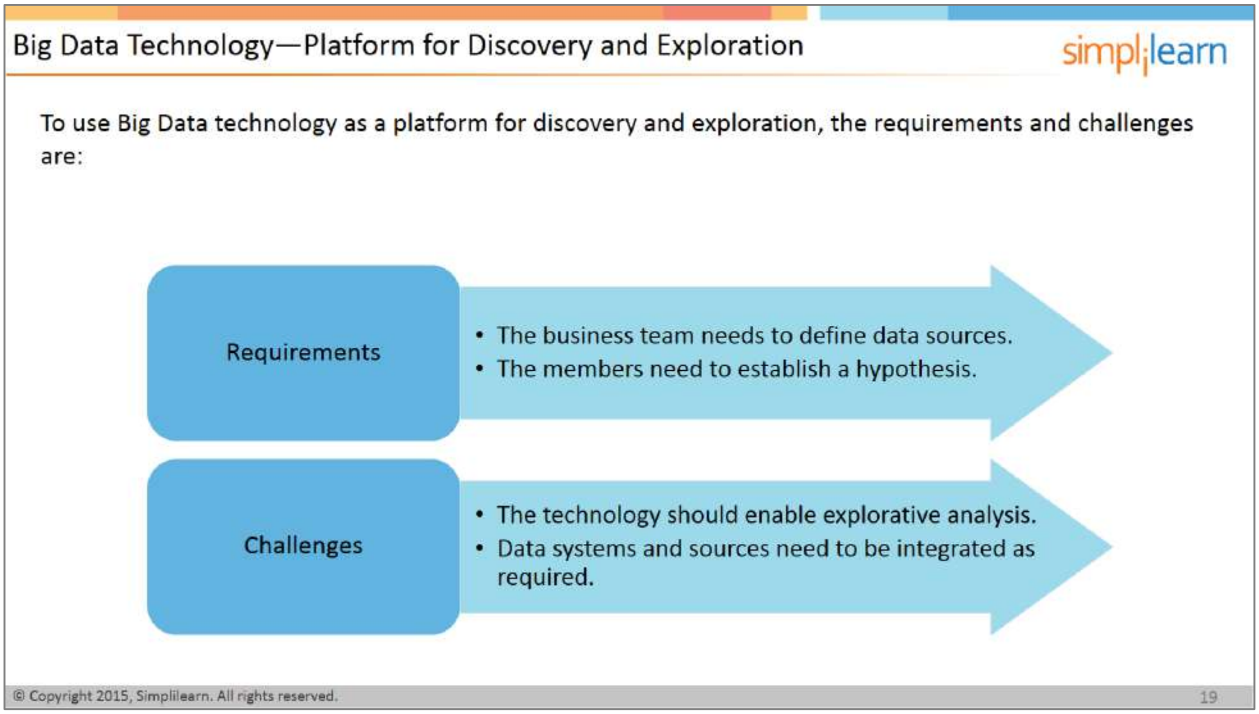
The following are the requirements of the traditional IT analytics approach and its challenging factors.

- The traditional IT analytics approach requires the business team to define questions before IT development. Furthermore, the team needs to pre-define the data sources and structures.
- The business team is usually challenged, if it has iterative and volatile requirements or if data sources keep changing



Traditional IT Analytics Approach (contd.)

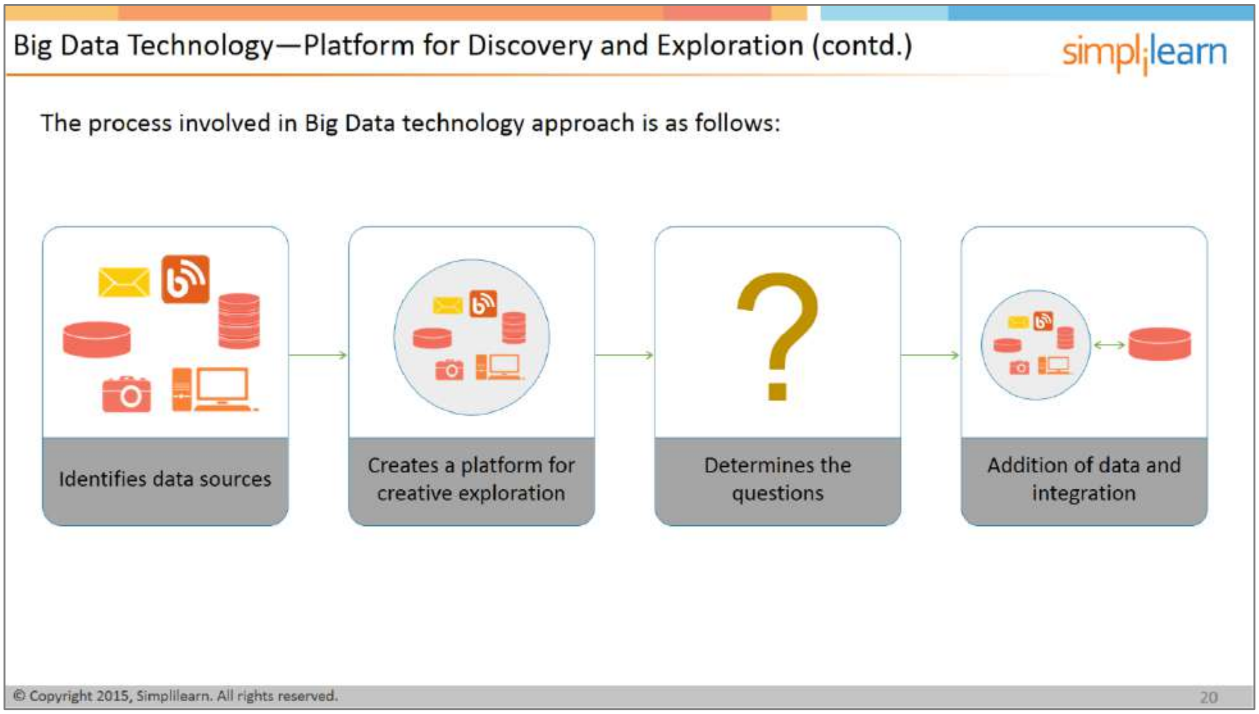
Let's now look at the process involved in traditional IT analytics approach. In a scenario of traditional IT systems development, the requirements are defined first. This is then followed by the solution design and build. Once the solution is implemented, queries are executed. If there are new requirements or queries, the system is redesigned and rebuilt.



Big Data Technology—Platform for Discovery and Exploration

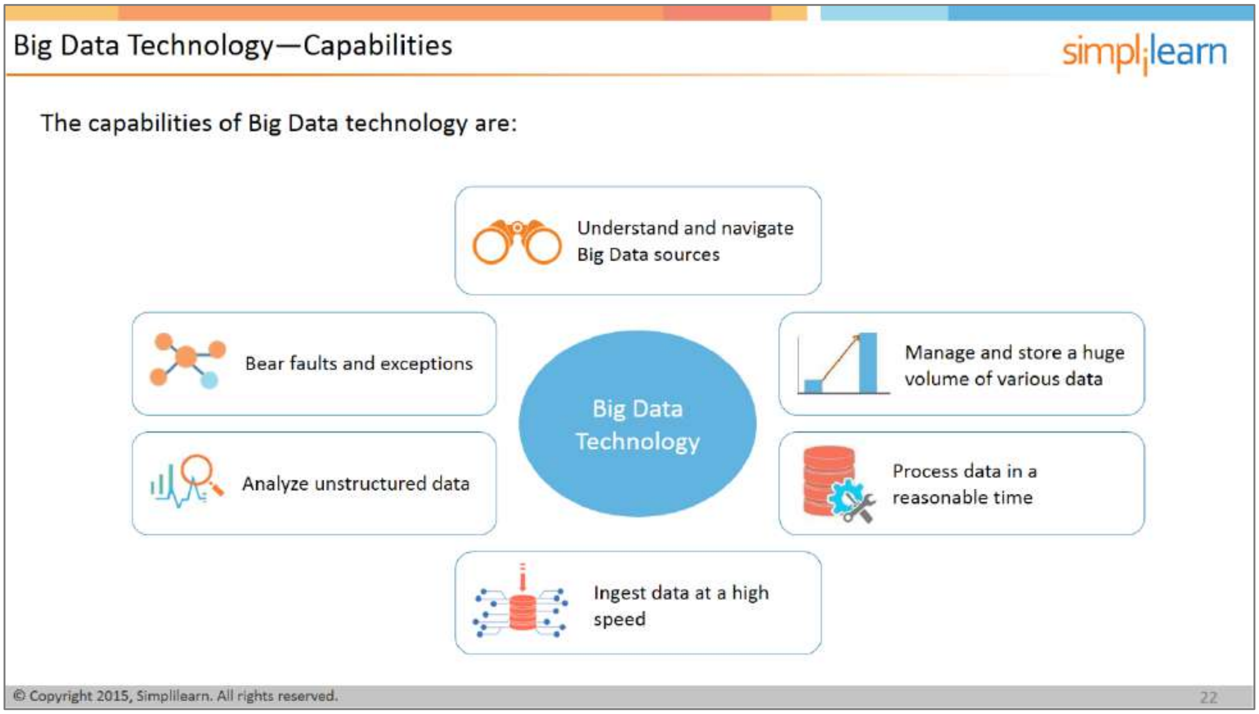
The following are the requirements and the challenging factors that must be overcome for using Big Data technology as a platform for discovery and exploration.

The Big Data technology approach requires the business team to define data sources and establish a hypothesis. Big Data technology should enable explorative analysis. The IT team should integrate data systems and sources, as required, based on new business questions and the hypotheses.



Big Data Technology—Platform for Discovery and Exploration (contd.)

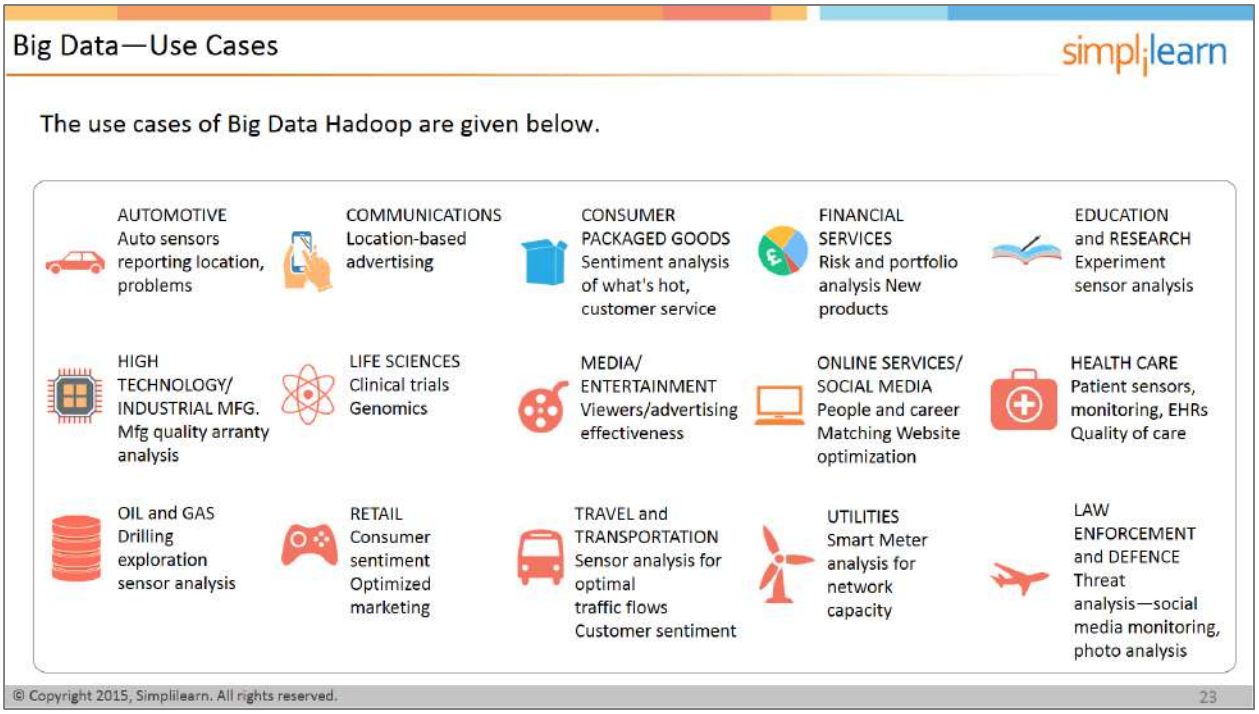
Let's now look at the process involved in Big Data technology approach. At the outset, the initial data sources are identified. The IT team creates a platform for creative exploration of available data and content. The business team then determines the questions to ask and tests their hypotheses. Any new question leads to the addition and integration of data sources without the need to redesign and rebuild the platform.



Big Data Technology—Capabilities

Big Data technology helps to:

- understand and navigate Big Data sources;
- manage and store a huge volume of various data;
- process data in a reasonable time;
- ingest data at a high speed;
- analyze unstructured data; and
- bear faults and exceptions.

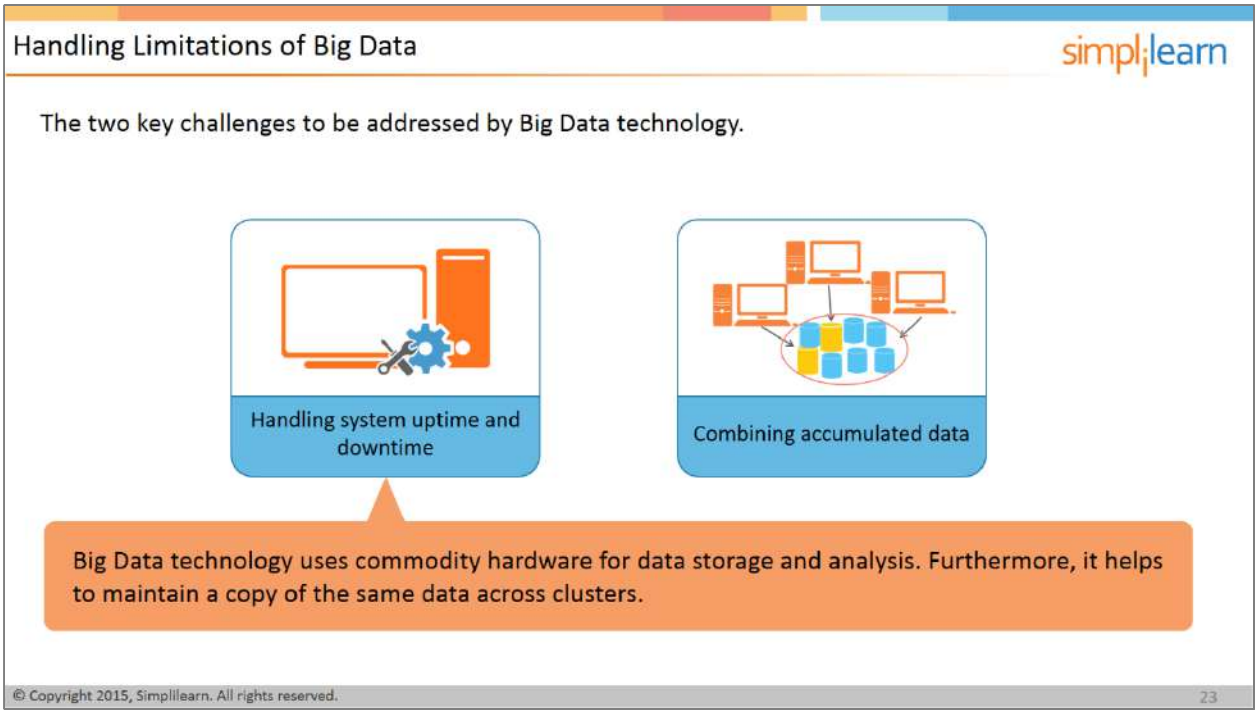


Big Data—Use Cases

Big Data technology appeals to various sectors and is used for a variety of use cases.

Per Cloudera, the technology has found uses in industries such as:

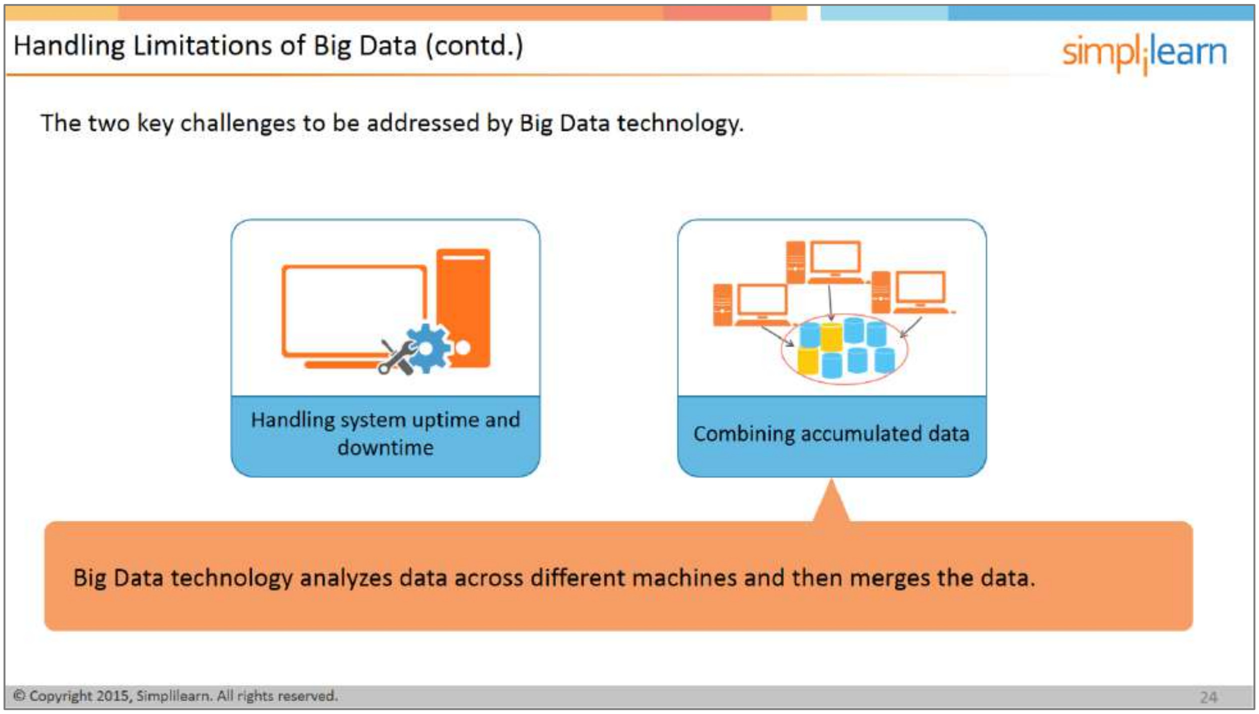
- Automotive;
- Communications;
- Consumer packaged goods;
- Financial services;
- Education and research;
- High technology and industrial manufacturing;
- Life sciences;
- Media and entertainment;
- Online services and social media;
- Health care;
- Oil and gas;
- Retail;
- Travel and transportation;
- Utilities; and
- Law enforcement and defense



Handling Limitations of Big Data

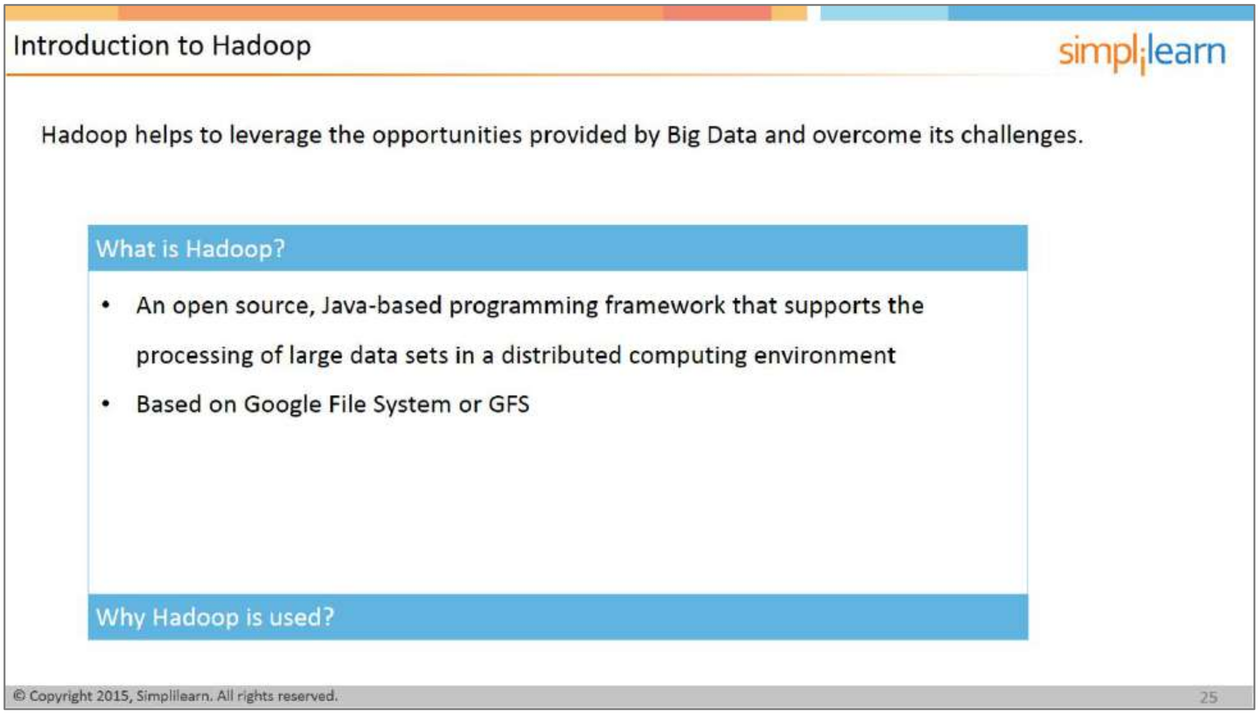
There are two key challenges to be addressed by Big Data technology. These are handling the system uptime and downtime, and combining data accumulated from all systems.

To overcome the first challenge, Big Data technology uses commodity hardware for data storage and analysis. Furthermore, it helps to maintain a copy of the same data across clusters.



Handling Limitations of Big Data (contd.)

To overcome the second challenge, Big Data technology analyzes data across different machines and then merges the data.

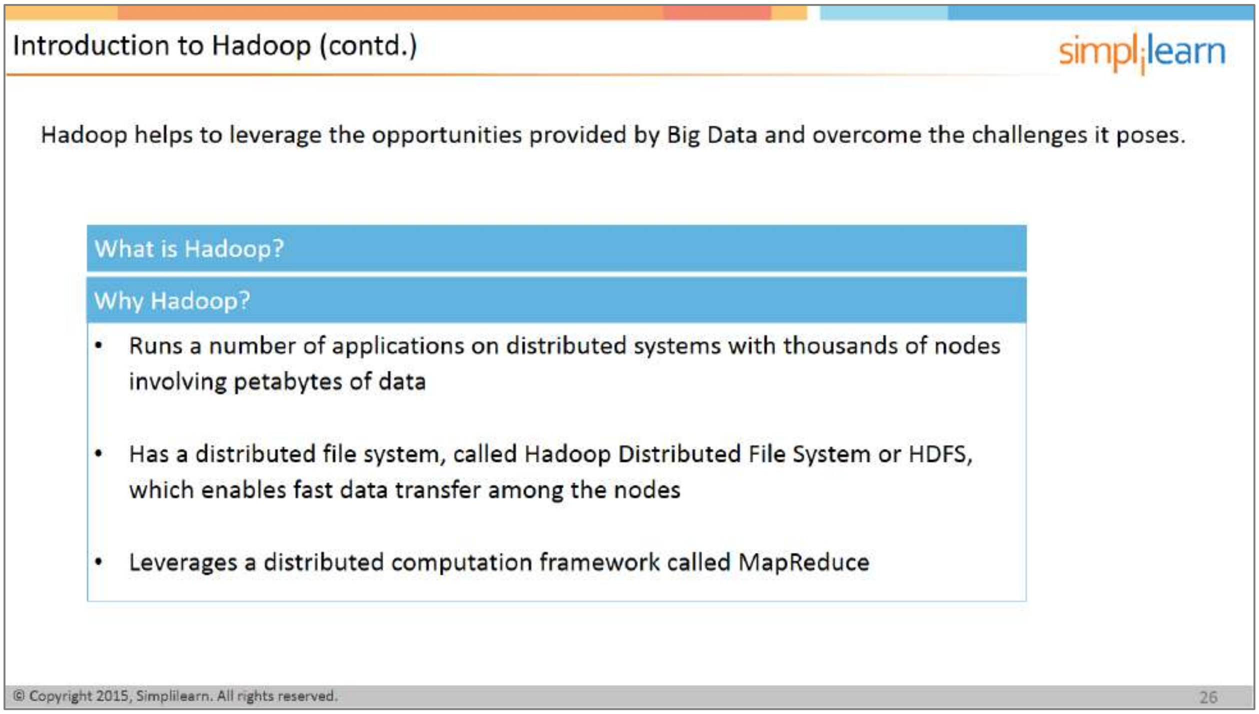


Introduction to Hadoop

Hadoop helps to leverage the opportunities provided by Big Data and overcome the challenges it poses.

Let's now look at the definition of Hadoop and its requirement

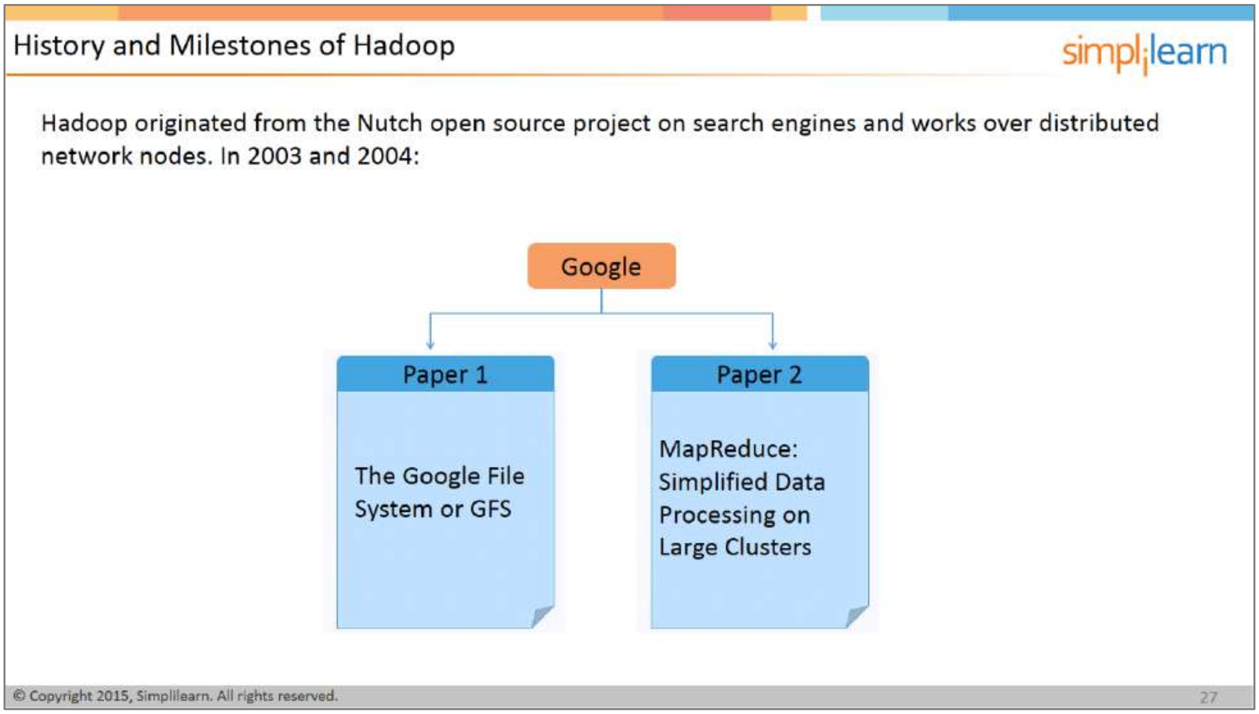
Hadoop is an open source, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is based on Google File System or GFS.



Introduction to Hadoop (contd.)

The next question to be addressed is why is Hadoop used?

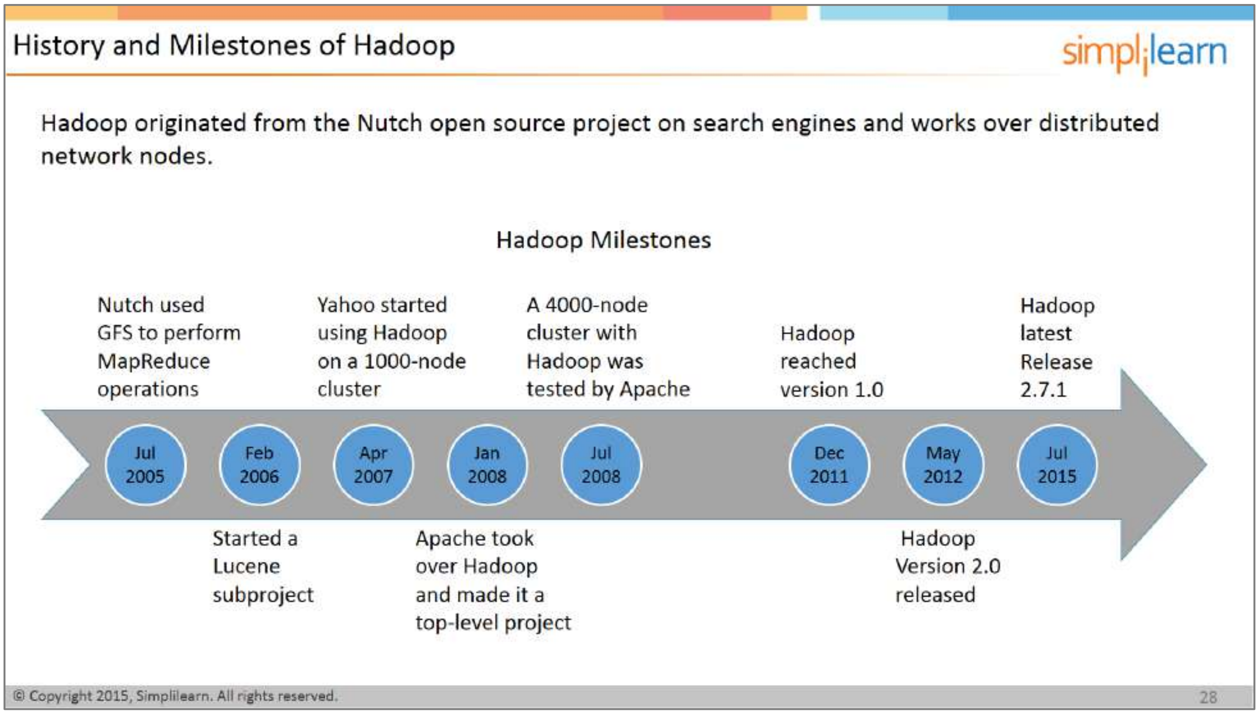
Hadoop runs a number of applications on distributed systems with thousands of nodes involving petabytes of data. It has a distributed file system, called the Hadoop Distributed File System or HDFS, which enables fast data transfer among the nodes. Furthermore, it leverages a distributed computation framework called MapReduce.



History and Milestones of Hadoop

Hadoop originated from the Nutch open source project on search engines and works over distributed network nodes.

In 2003 and 2004, Google released two papers that provided insight into their success—"The Google File System or GFS" and "MapReduce: Simplified Data Processing on Large Clusters." The papers told the world how Google performed large-scale data processing.



History and Milestones of Hadoop (contd.)

In July 2005, Nutch used GFS to perform MapReduce operations. In February 2006, Nutch started a Lucene subproject which led to the era of Hadoop. In April 2007, Yahoo started using Hadoop on a 1000-node cluster. In January 2008, Apache took over Hadoop and made it a top-level project. In July 2008, a 4000-node cluster with Hadoop was tested by Apache. The performance of that cluster was surprisingly the fastest when compared to the other technologies implemented that year. In May 2009, a test revealed that Hadoop successfully sorted a petabyte of data in 17 hours. Hadoop reached version 1.0 in December 2011. It is open source and written in Java.

On 23 May, 2012 Hadoop 2.0.0 was released which delivers significant features including: YARN, high availability for HDFS, HDFS Federation, HDFS Snapshots, NFSv3 access to data in HDFS, support for running Hadoop on Microsoft Windows, binary compatibility for MapReduce applications built on hadoop-1.x, and substantial integration testing with rest of the projects in the ecosystem.

The current version of Hadoop is 2.7.1 was released in 06 July, 2015. It is completely open source and written in Java.

Organizations Using Hadoop

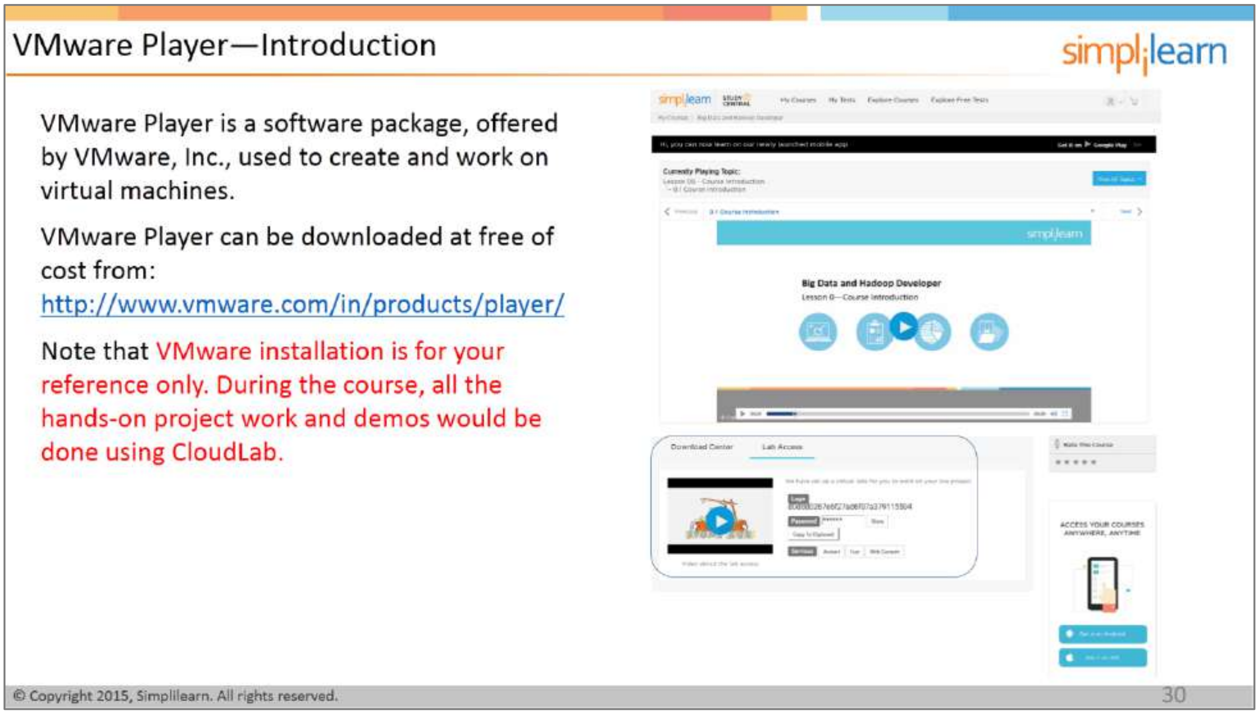
The following table shows how various organizations use Hadoop:

Name of the organization	Cluster specifications	Uses
A9.com: Amazon	Clusters vary from 1 to 100 nodes	• Amazon's product search indices are built using this program • Processes millions of sessions daily for analytics
Yahoo	More than 100,000 CPUs in approximately 20,000 computers running Hadoop; biggest cluster has 2000 nodes (2*4 cpu boxes with 4 TB disk each)	• To support research for ad systems and web search
AOL	Cluster size is 50 machines, Intel Xeon, dual processors, and dual core, each with 16 GB RAM and 800 GB hard disk resulting in a total of 37 TB HDFS capacity	• For a variety of functions ranging from generating data to running advanced algorithms for performing behavioral analysis and targeting
Facebook	320-machine cluster with 2,560 cores and about 1.3 PB raw storage	• Storing copies of internal log and dimension data sources • Used as a source for reporting analytics and machine learning

 29

Organizations using Hadoop

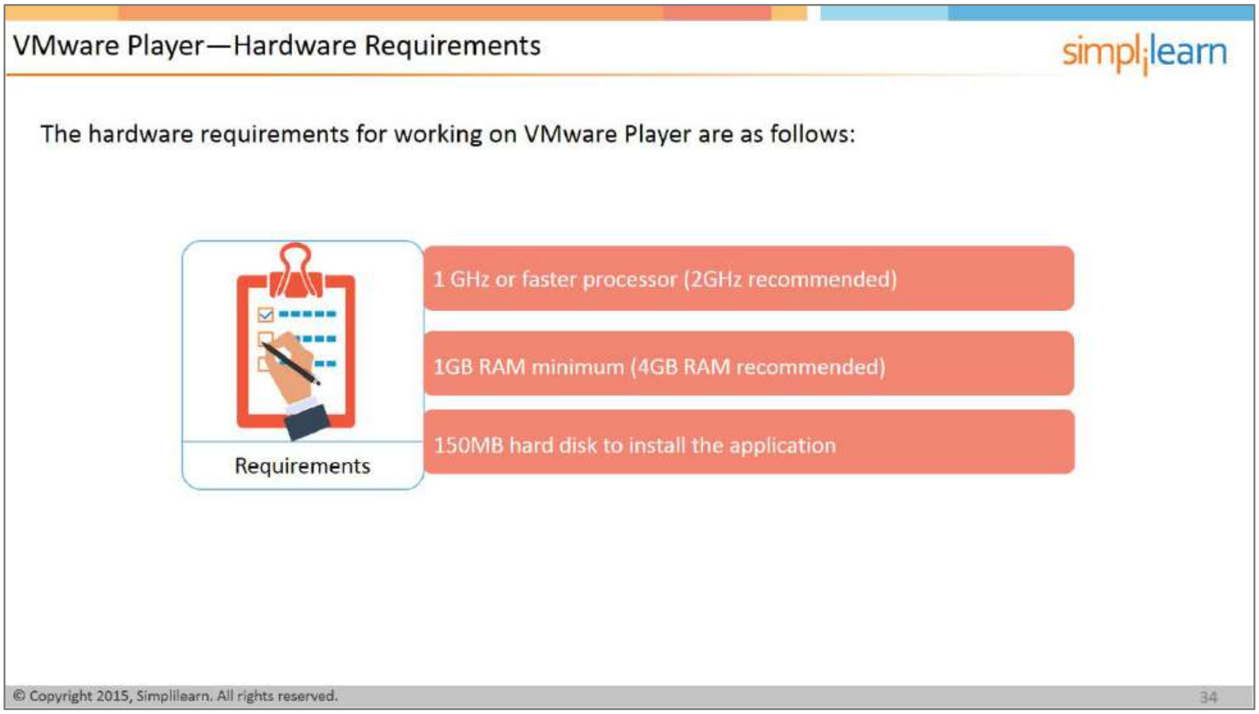
Yahoo was the first company to design and use Hadoop as a core part of their system operations. Now, Hadoop is a core part of systems at Internet companies such as Facebook, LinkedIn, Twitter, and many enterprise organizations such as JP Morgan and Chevron.



VMWare Player—Introduction

VMware Player is a free software package offered by VMware, Inc. used to create and manage virtual machines.

The VMware Player can be downloaded for personal use from the URL mentioned here.

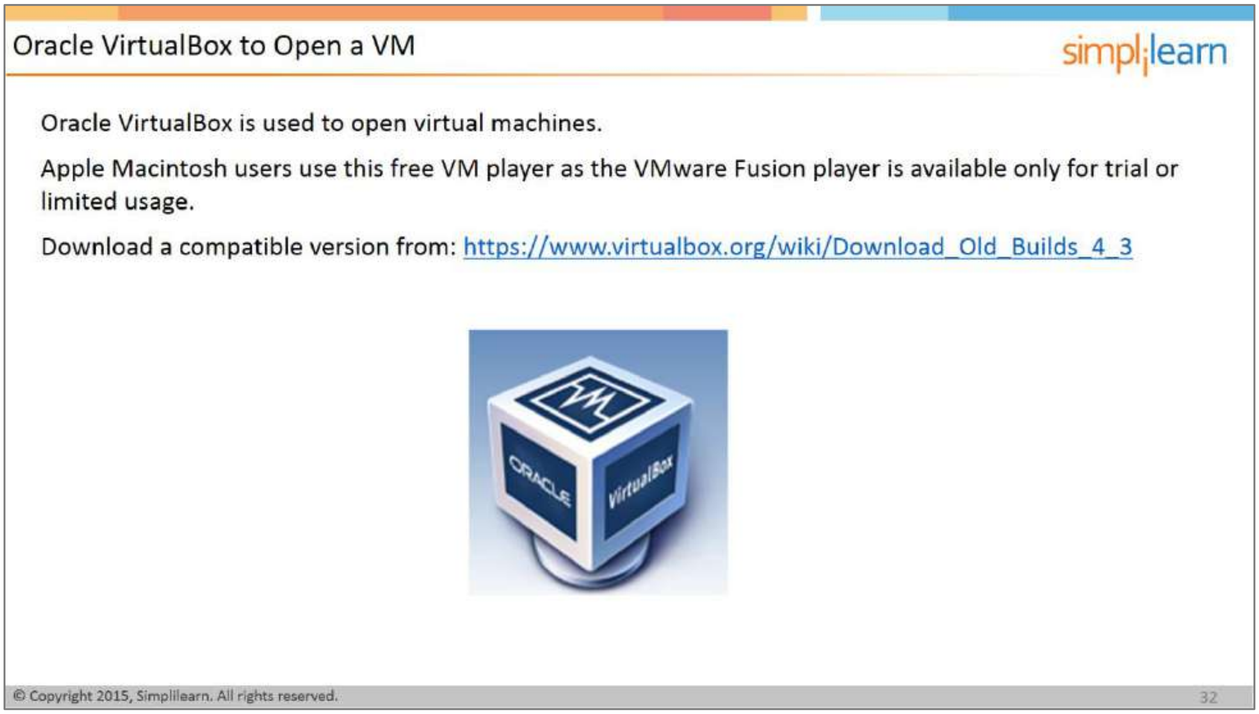


VMWare Player—Hardware Requirements

The basic hardware requirements for working on VMware Player are as follows:

A 1GHz processor to support Intel VT or Virtualization Technology, 1GB RAM, and 150MB hard disk to install the application.

However, it is recommended that you have a 2GHz processor and 4GB RAM for optimum performance.

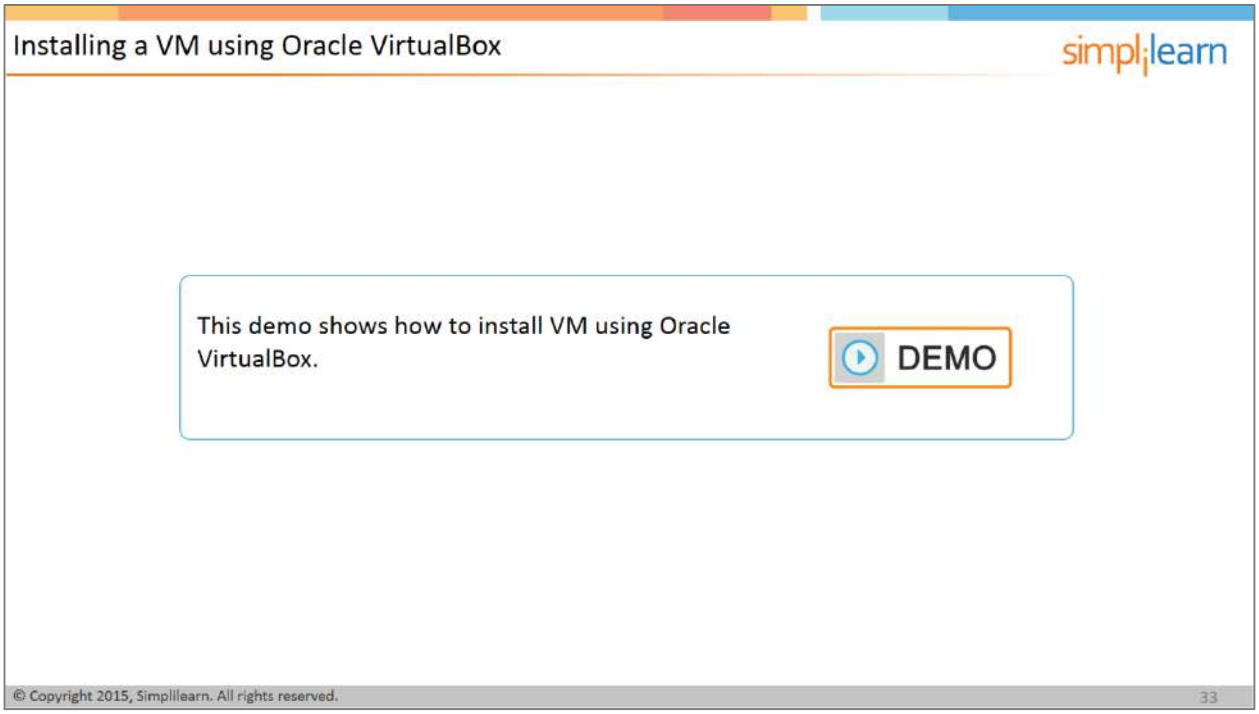


Oracle VirtualBox to Open a VM

The Oracle VirtualBox is used to open virtual machines.

Apple Macintosh users use this free VM player as the VMware Fusion player is available only for trial or limited usage.

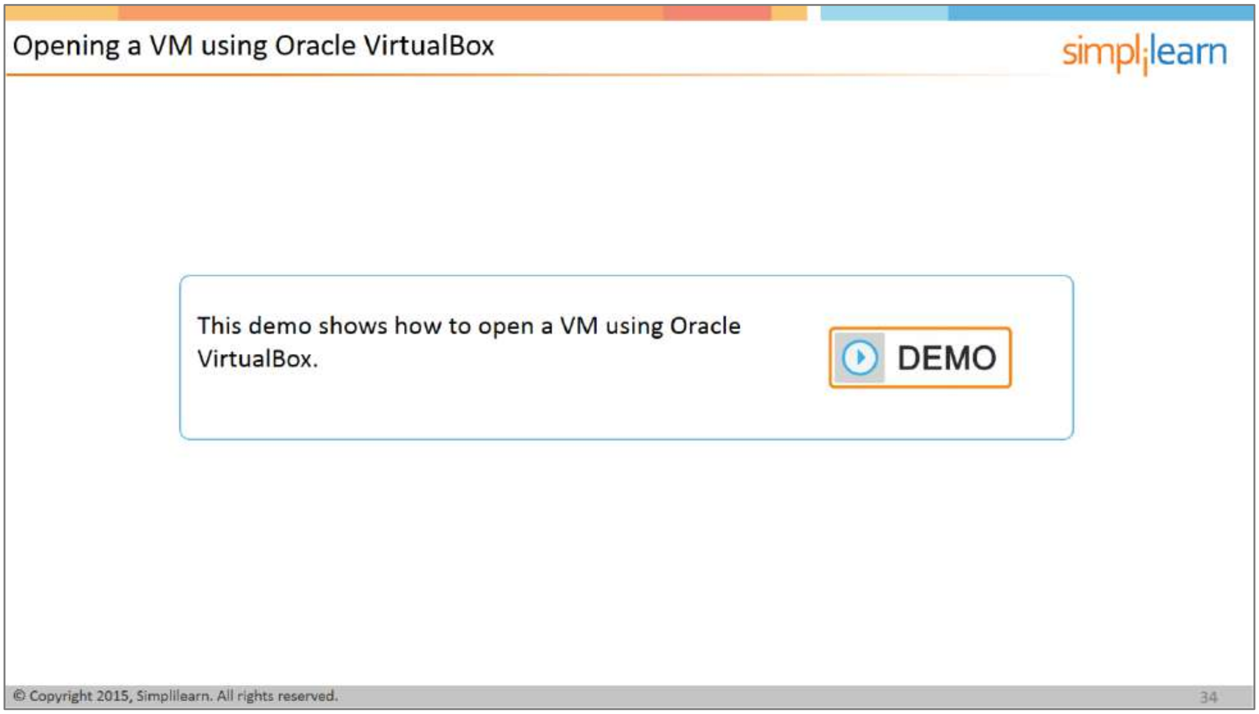
Download the latest version of VirtualBox compatible with your machine and firewall settings from the URL mentioned here.



Installing a VM using Oracle VirtualBox

Let us summarize the steps in this demo.

1. Download Oracle VirtualBox from the link https://www.virtualbox.org
2. Click **Downloads**.
3. Select your OS and download the associated file.
4. Launch the downloaded Oracle VirtualBox installation file.
5. In the Oracle VM VirtualBoxsetup window that opens, click **Next**.
6. In the custom setup section that appears, click **Next**.
7. A warning message about network connection opens. Click **Yes** to proceed.
8. Click **Install**. Click **Finish** to complete the installation process.



Opening a VM using Oracle VirtualBox

Let us summarize the steps in this demo.

1. Open Oracle VM VirtualBox Manager. Click **New**.
2. The ‘Create Virtual Machine’ dialog box appears. Enter the name and target OS, and click **Next**.
3. Set memory as 2048MB or more.
4. Select **Use an existing virtual hard drive file**.
5. Click **Create**.
6. Select the new VM from the left pane and click **Start** to open it.
7. In the Hadoop Simplilearn-Oracle VM Virtual Box windows that opens, select **Switch to Scaled mode** from the **View** menu to resize the VM. Click **Switch** to continue.

QUIZ 1

Which types of data are handled by Hadoop?

a. Structured data

b. Semi-structured data

c. Unstructured data

d. Flexible-structure data

QUIZ 2

Which of the following is an unstructured data?

a. Collection of text files

b. Collection of XML files

c. Collection of tables in databases

d. Collection of tickets

QUIZ 3

Which of the following is structured data?

a. Collection of text files having unformatted log messages

b. Collection of JSON data

c. Collection of tables in databases

d. Collection of XML files

QUIZ 4

Which of the following is semi-structured data?

a. Collection of tables in databases

b. Collection of text files

c. Collection of sensor data

d. Collection of XML files

QUIZ 5

Which of the following aspects of Big Data refers to data size?

a. Volume

b. Velocity

c. Variety

d. Value

 44

QUIZ 6

Which of the following aspects of Big Data refers to the response speed of appropriate data request generated by the user?

a. Variety

b. Value

c. Velocity

d. Volume

 46

QUIZ 7

Which of the following aspects of Big Data refers to multiple data sources?

a. Variety

b. Value

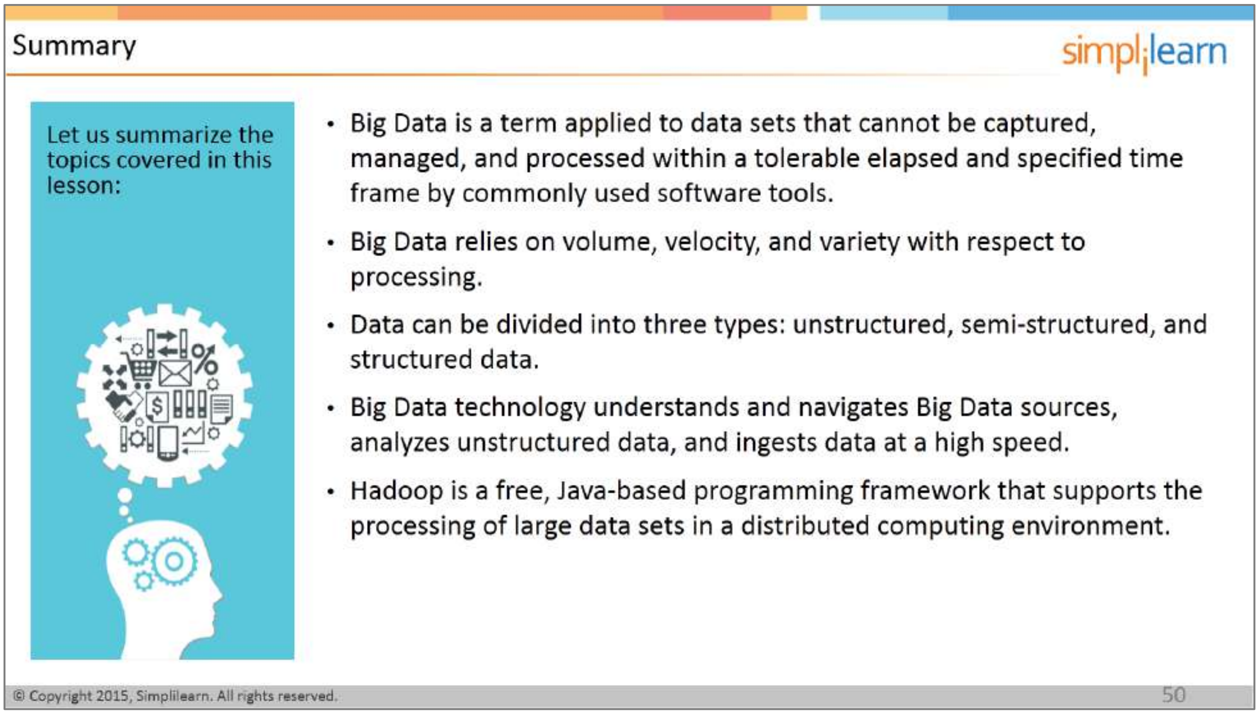
c. Volume

d. Velocity

 48

ANSWERS:

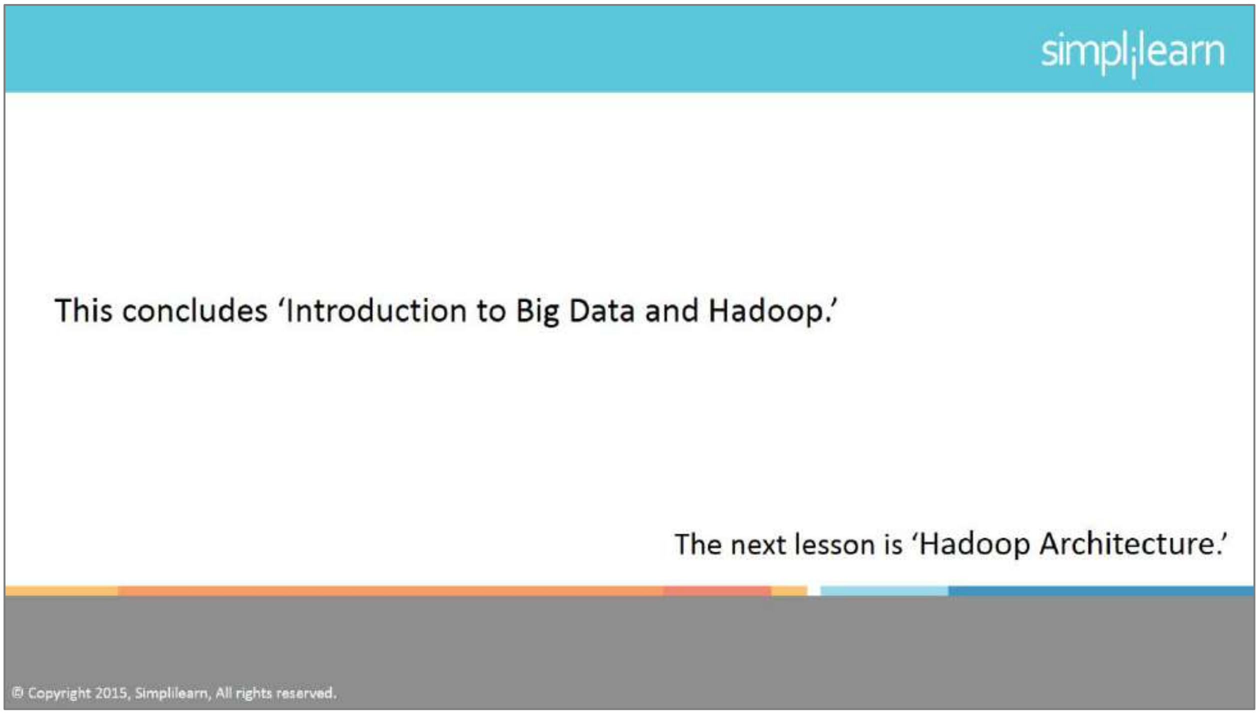
S.No.	Question	Answer & Explanation
1	Which types of data are handled by Hadoop?	a, b, and c. Hadoop handles structured, unstructured and semi-structured data for processing.
2	Which of the following is an unstructured data?	a. Review command inside Text will be considered as unstructured data.
3	Which of the following is structured data?	c. Databases are sources of highly structured data.
4	Which of the following is semi-structured data?	d. XML files are included in the category of semi-structured data.
5	Which of the following aspects of Big Data refers to data size?	a. Volume in Big Data refers to the size of the data set to be processed.
6	Which of the following aspects of Big Data refers to the response speed of appropriate data request generated by the user?	c. Velocity in Big Data refers to the response speed of appropriate data request generated by the user.
7	Which of the following aspects of Big Data refers to multiple data sources?	a. Variety in Big Data refers to multiple data sources.



Summary

Let us summarize the topics covered in this lesson:

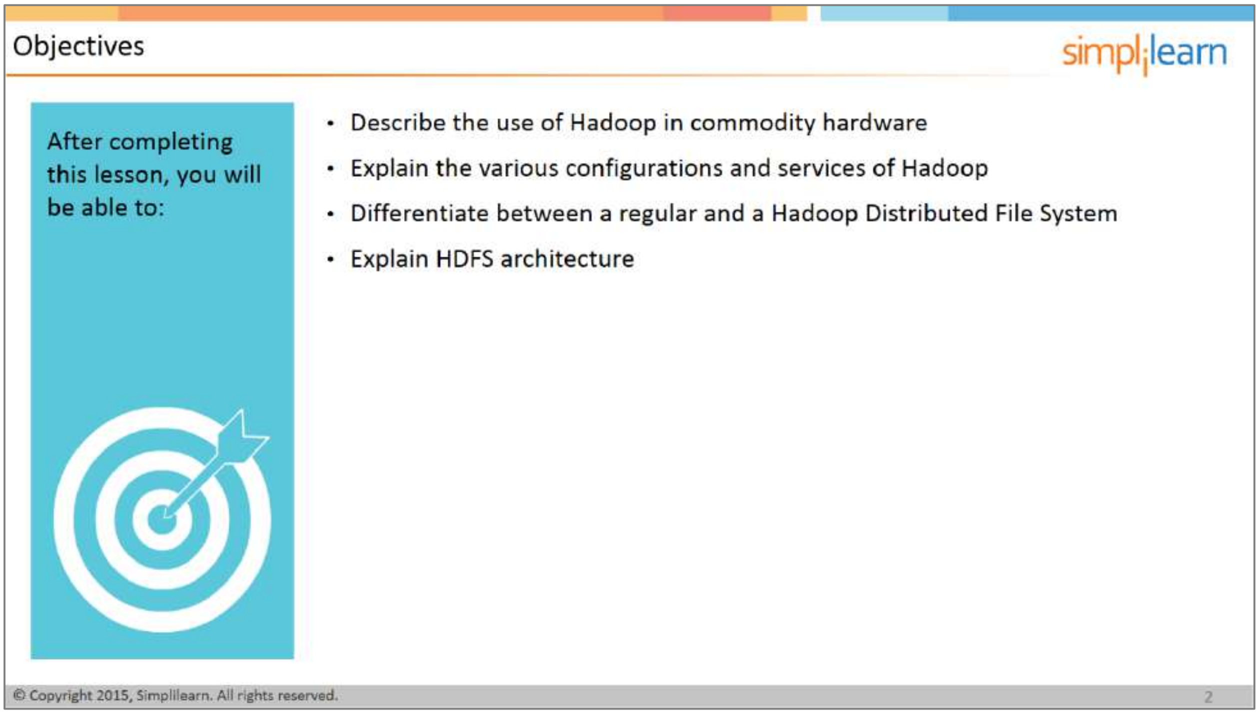
- Big Data is a term applied to data sets that cannot be captured, managed, and processed within a tolerable elapsed and specified time frame by commonly used software tools.
- Big Data relies on volume, velocity, and variety with respect to processing.
- Data can be divided into three types—unstructured, semi-structured, and structured data.
- Big Data technology understands and navigates Big Data sources, analyzes unstructured data, and ingests data at a high speed.
- Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment.



Conclusion

This concludes 'Introduction to Big Data and Hadoop.' In the next lesson we will focus on 'Hadoop Architecture.'

Lesson 2—Hadoop Architecture



Hadoop Architecture—Objectives

After completing this lesson, you will be able to:

- Describe the use of Hadoop in commodity hardware
- Explain the various configurations and services of Hadoop
- Differentiate between a regular and a Hadoop Distributed File System
- Explain HDFS architecture

Key Terms

Key terms in Hadoop Architecture:

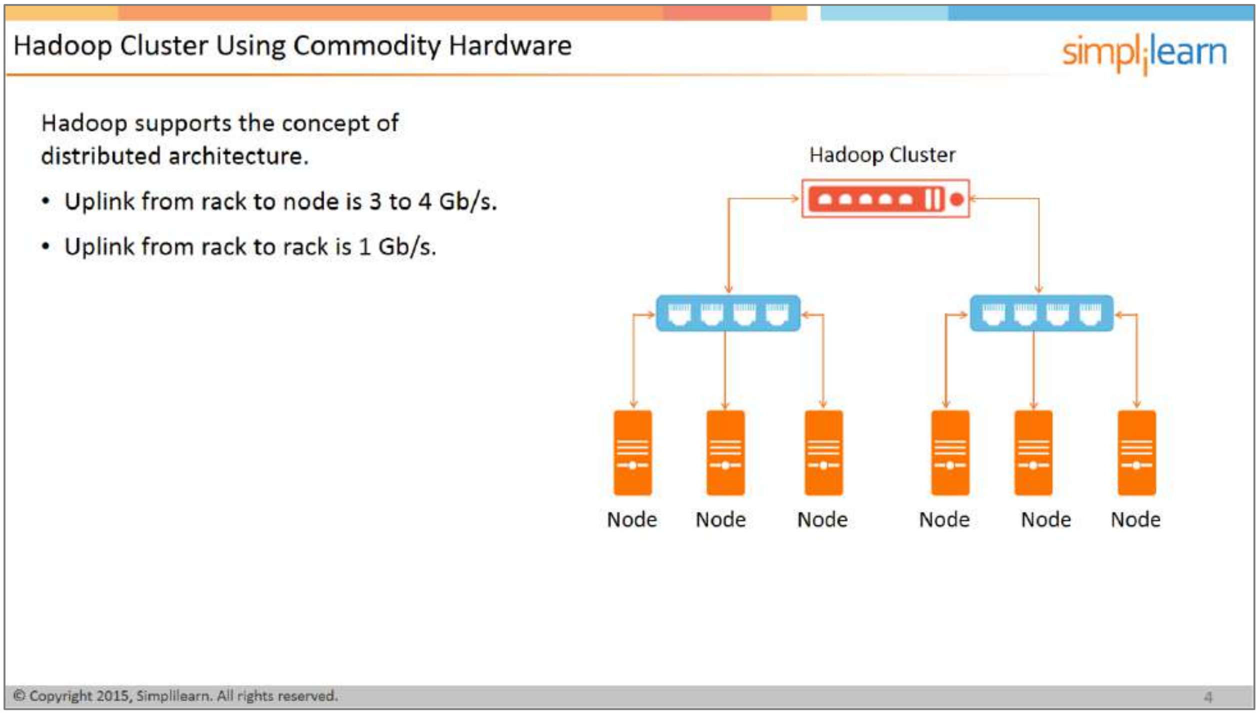
- Commodity hardware: PCs which can be used to make a cluster
- Cluster: Interconnection of systems in a network
- Node: Commodity servers interconnected through a network device

 3

Key Terms

An understanding of the key terms is essential when discussing Hadoop Architecture.

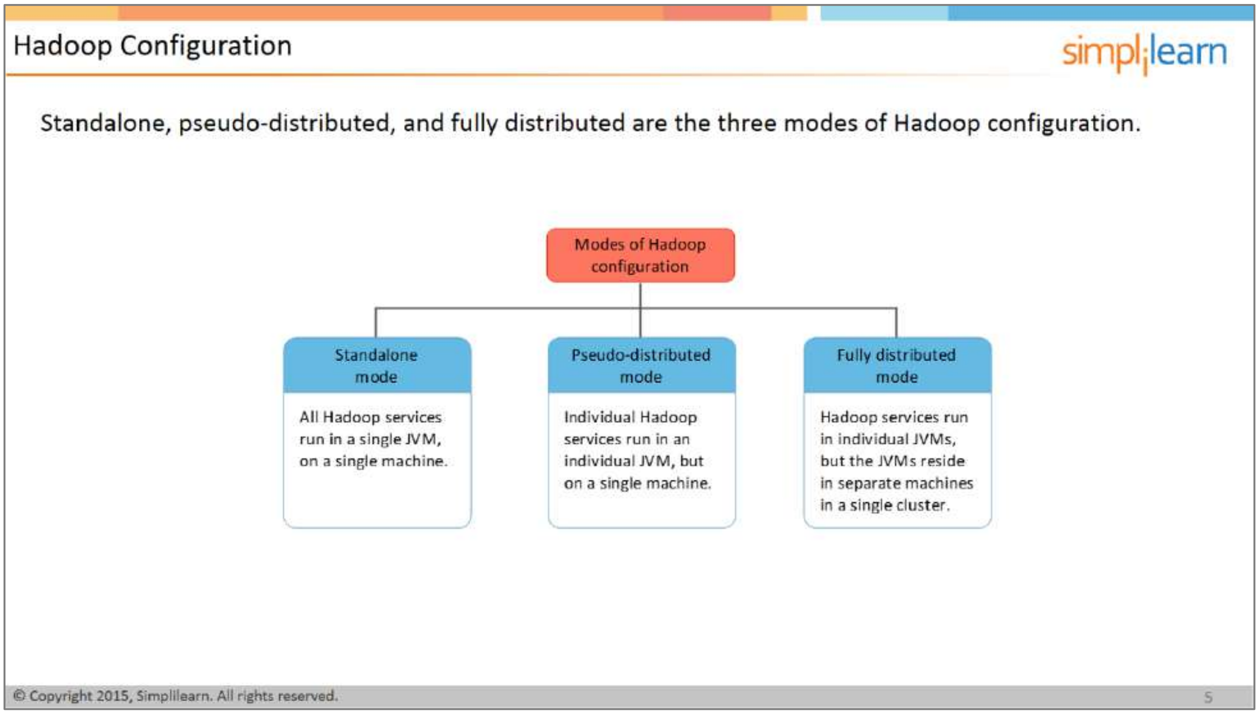
The term ‘commodity hardware’ refers to regular PCs which can successfully be used to make a cluster. The word ‘cluster’ refers to multiple systems logically interconnected in the same network for a distributed mode process. ‘Nodes’ are the commodity servers which are interconnected through a network device.



Hadoop Cluster using Commodity Hardware

Hadoop is a framework which is often used in commodity hardware. It supports the concept of distributed architecture.

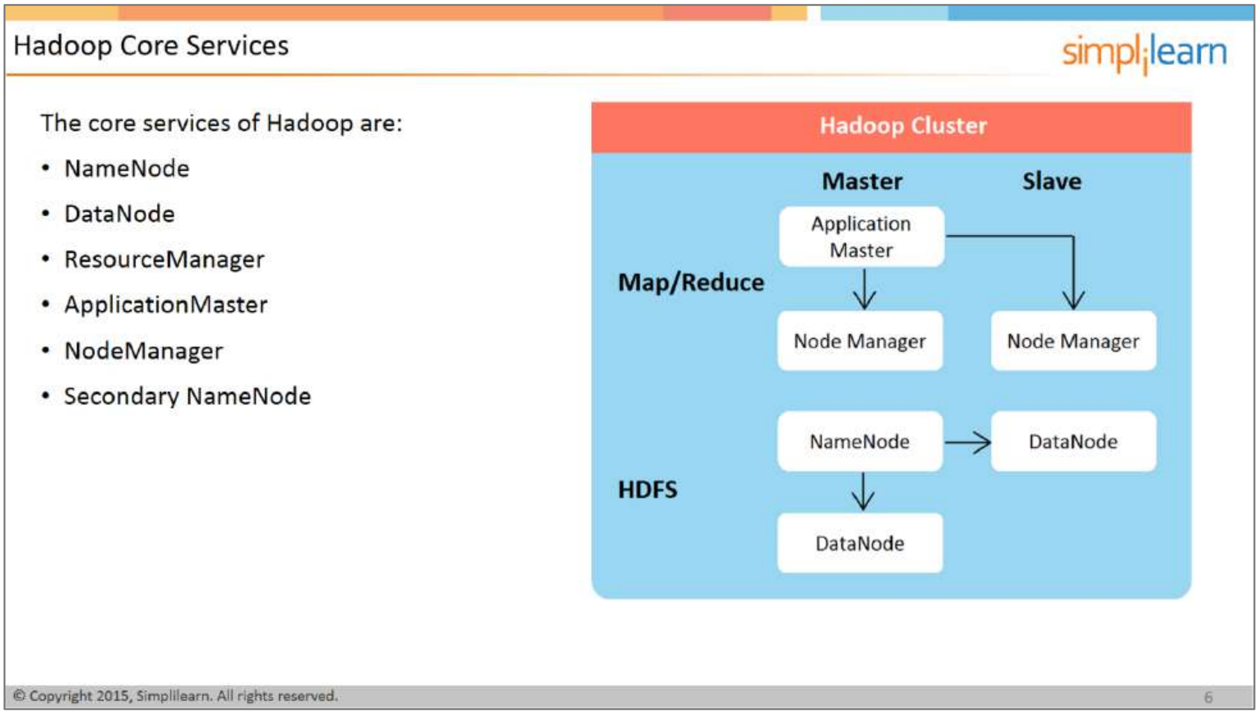
The diagram represents the cluster of nodes connected and installed with Hadoop. The maximum number of nodes that can reside in a single rack is approximately 30 to 40. However, it is not an exact count. You can have more nodes if you have adequate network speed. Normally, the uplink from rack to node is three to four gigabits per second. The uplink from rack to rack internally is one gigabit per second



Hadoop Configuration

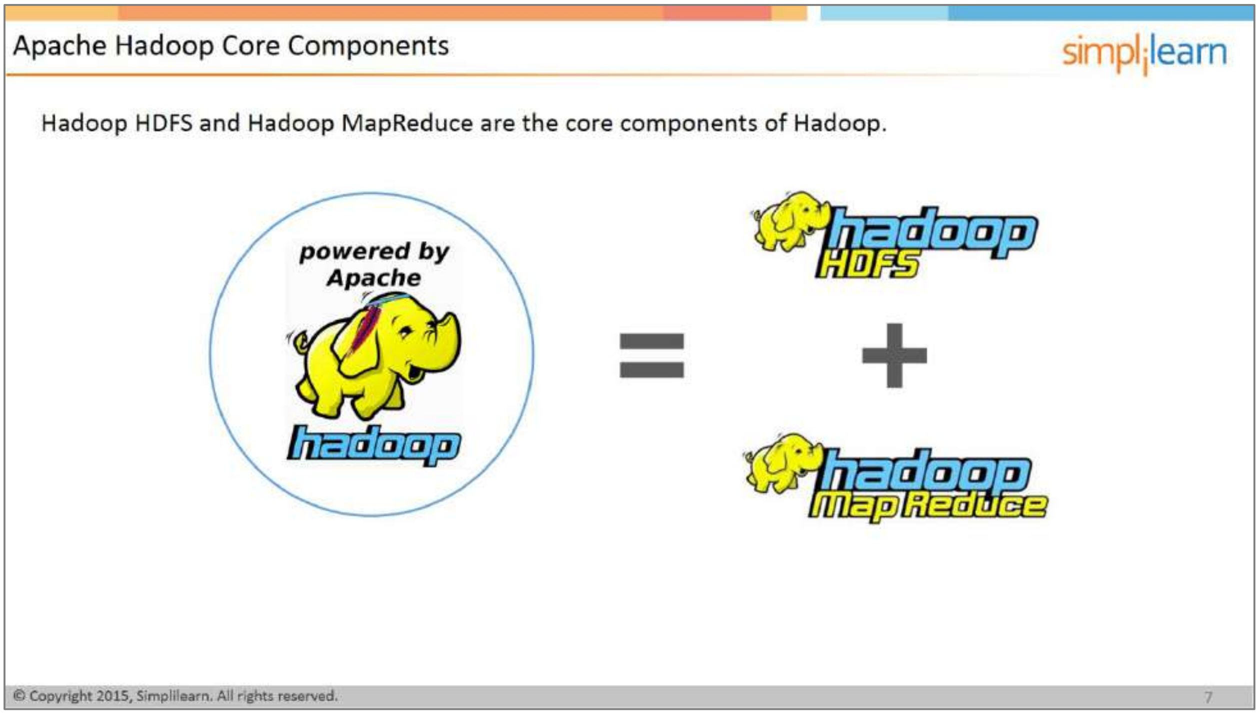
Hadoop supports three configuration modes when implemented on commodity hardware: standalone, pseudo-distributed, and fully-distributed mode.

- In a standalone mode, all Hadoop services run in a single Java Virtual Machine or JVM on a single machine.
- In a pseudo-distributed mode, each Hadoop service runs in its own JVM but on a single machine.
- In a fully distributed mode, the Hadoop services run in individual JVMs, but these JVMs reside on different commodity hardware in a single cluster.



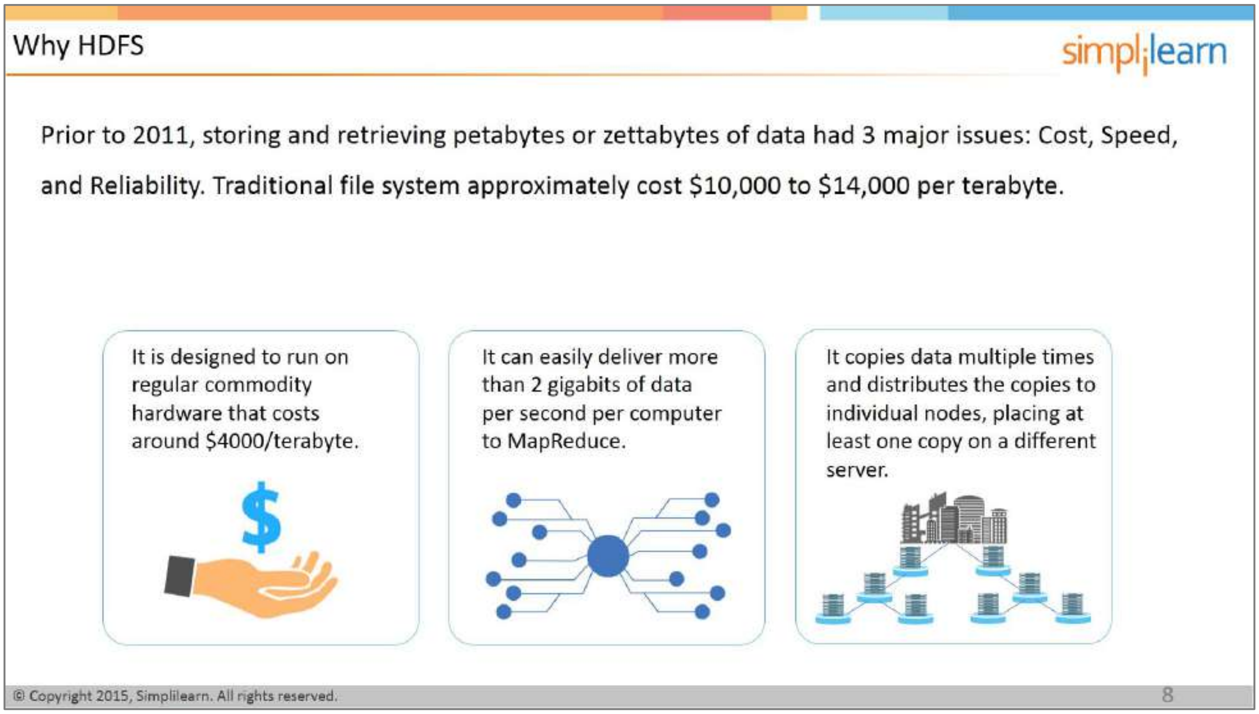
Hadoop Core Services

The core Hadoop services are: NameNode, DataNode, ResourceManager, ApplicationMaster, NodeManager, and Secondary NameNode. Generally, in the fully distributed mode, the NameNode, Secondary NameNode, and ApplicationMaster are identified as master services while the DataNode and NodeManager are classified as slave services.



Apache Hadoop Core Components

There are two major components of Apache Hadoop, Hadoop Distributed File System or HDFS and Hadoop MapReduce. HDFS is used to manage the storage aspects of Big Data, and MapReduce is responsible for processing jobs in a distributed environment.



Why HDFS

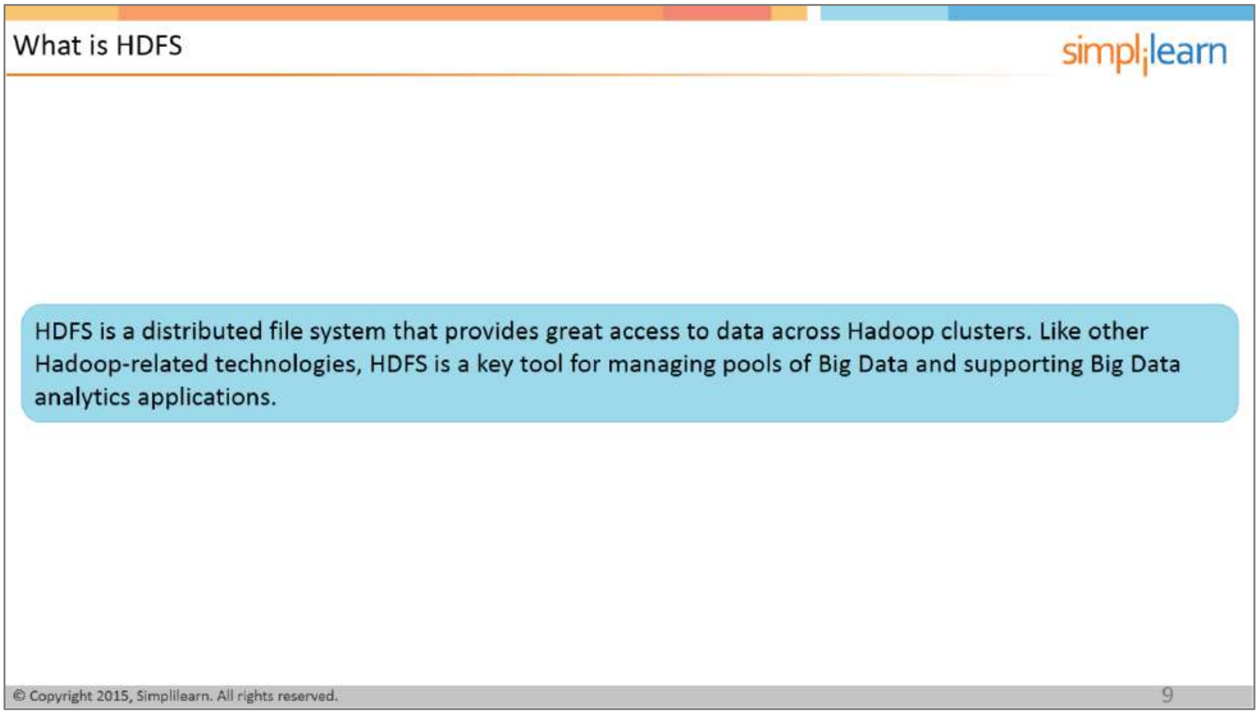
Prior to 2011, storing and retrieving petabytes or zeta bytes of data had 3 major issues: Cost, Speed, and Reliability.

Traditional file system approximately cost $10,000 to $14,000 per terabyte. Searching and analyzing data was both time consuming and expensive. And, if search components were saved in different servers, fetching data was a probability. Things would break every day in different ways.

Here, HDFS or Hadoop Distributed File System comes into picture. Its major USPs are:

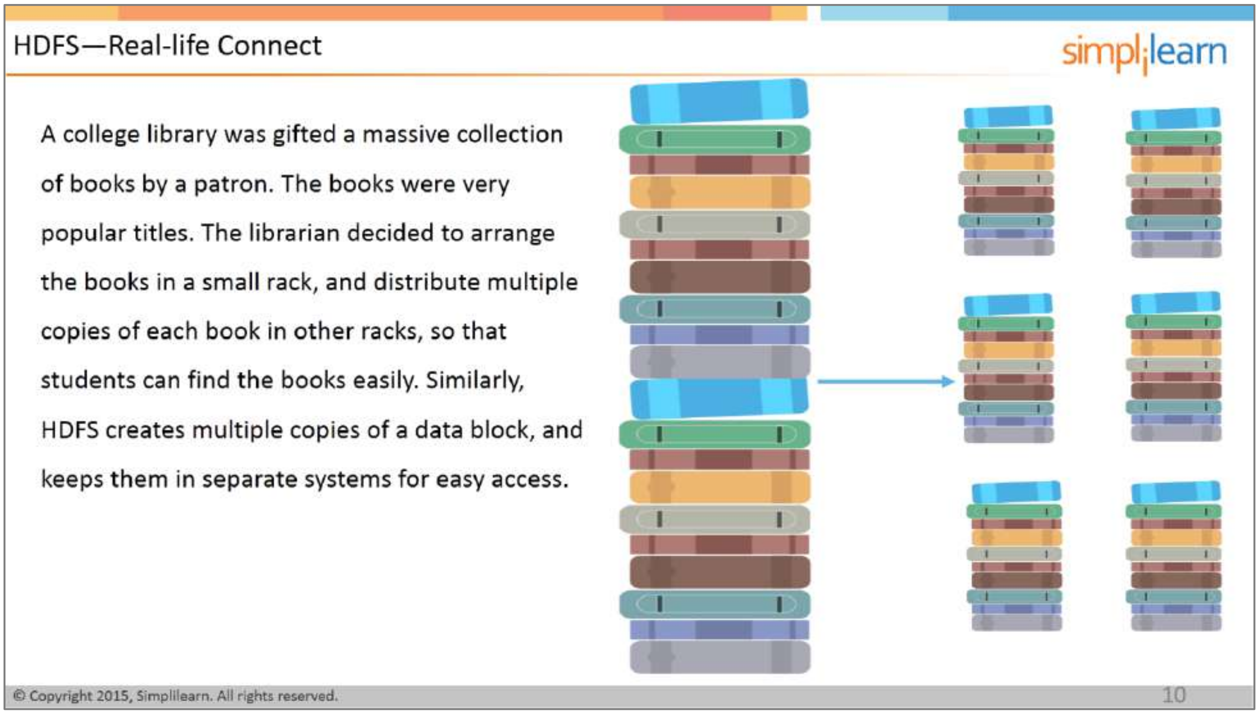
- It is designed to run on regular commodity hardware that costs around $4000/terabyte. Since it is an open source software, it can be used with zero licensing and support costs. This cost advantage lets organizations store and process more data per dollar than traditional systems. In Big Data deployments, the storage cost often determines viability of the system.
- It can easily deliver more than 2 gigabits of data per second per computer to MapReduce. This means, large Hadoop clusters can read/write more than a terabyte of data per second.

- It copies data multiple times and distributes the copies to individual nodes, placing at least one copy on a different server. As a result, the data on nodes that crash can be found within the cluster, which allows processing to continue while the failure is resolved.



What is HDFS

HDFS is a distributed file system that provides great access to data across Hadoop clusters. Like other Hadoop-related technologies, HDFS is a key tool for managing pools of Big Data and supporting Big Data analytics applications.



HDFS—Real-life Connect

A college library was gifted a massive collection of books by a patron. The books were very popular titles. The librarian decided to arrange the books in a small rack, and distribute multiple copies of each book in other racks, so that students can find the books easily. Similarly, HDFS creates multiple copies of a data block, and keeps them in separate systems for easy access.

Regular File System vs. HDFS

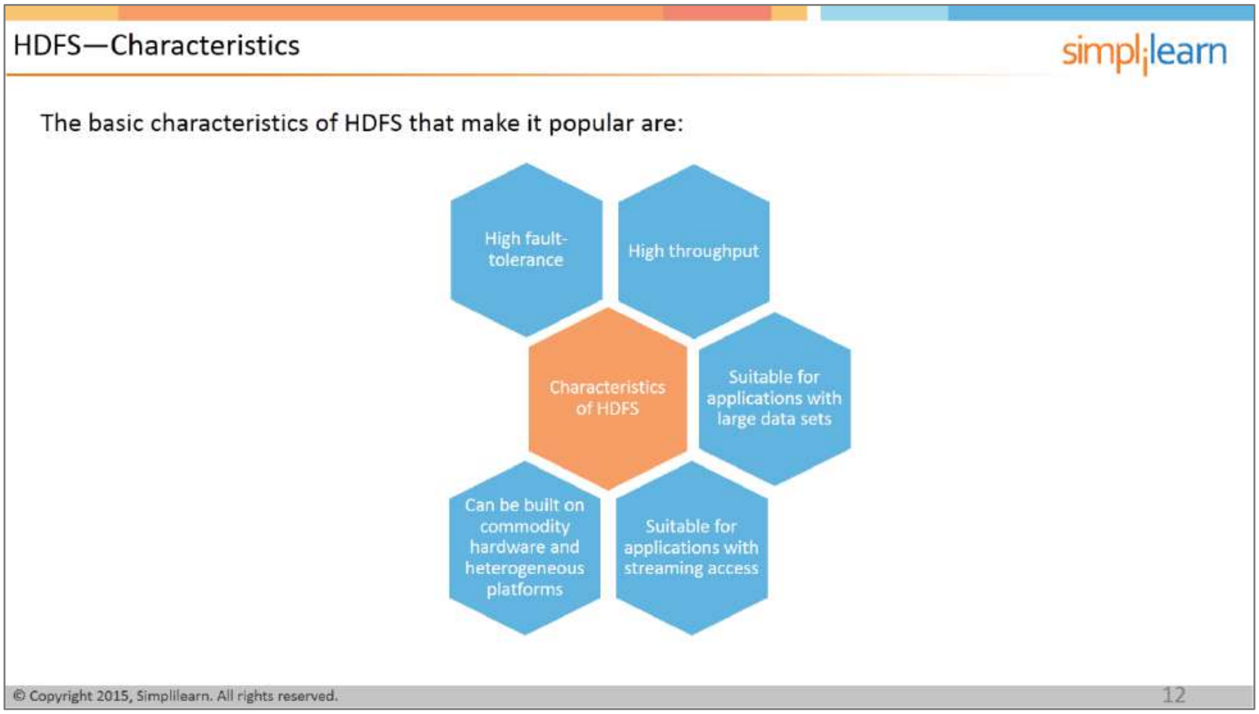
The table compares a regular file system with HDFS.

Regular File System	HDFS
Each block of data is small, approximately 51 bytes	Each block of data is large, 64MB by default
Large data access suffers from disk I/O problems, primarily because of multiple seek operations	Reads huge data sequentially after a single seek

 11

Regular File System vs. HDFS

In a regular file system, each block of data is small, usually about 51 bytes. However, in HDFS, each block is of 64MB by default. A regular file system provides access to large data, but this feature of the file system may suffer from disk I/O problems, mainly due to multiple seek operations. HDFS can read large quantities of data sequentially after a single seek. This makes it unique since all of these operations are performed in distributed mode.



HDFS Characteristics

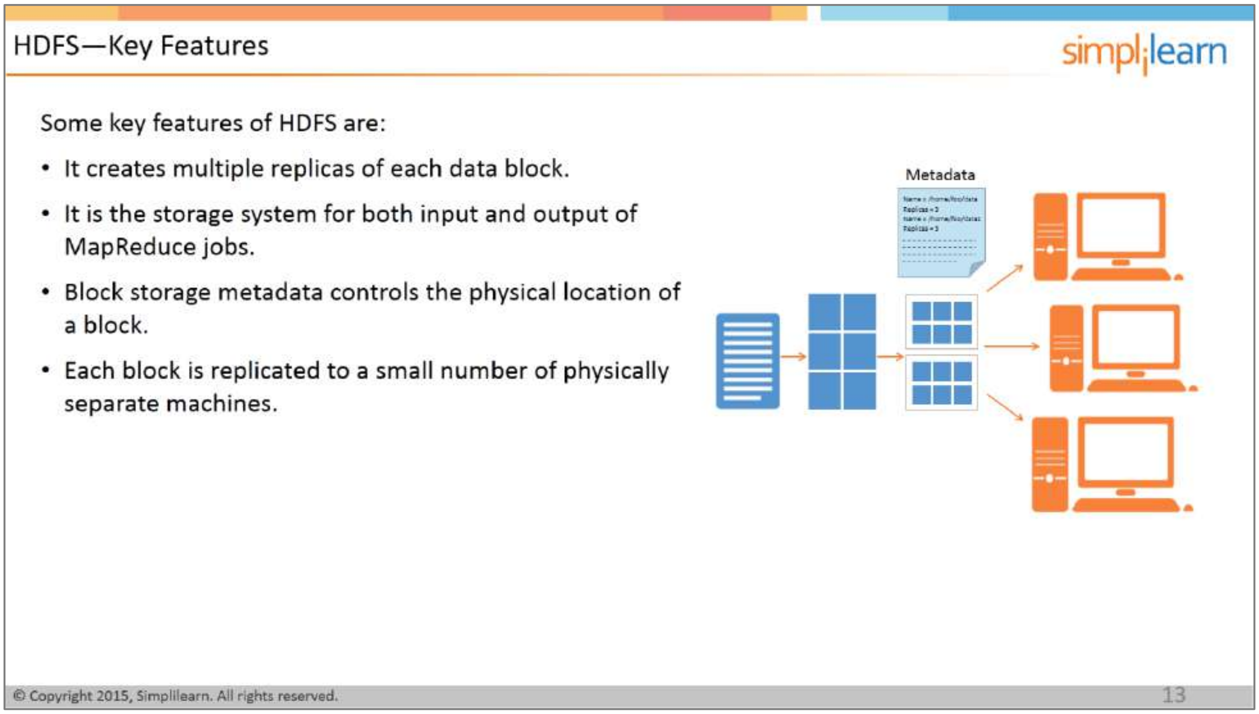
The basic characteristics of HDFS that make it popular are:

First, HDFS has a high fault-tolerance. An HDFS instance may consist of thousands of server machines, each storing a part of the file system's data. There exists numerous components and each has a probability of failure and non-functionality. The core architectural goals of HDFS are detection of faults and quick, automatic recovery. Second, HDFS provides high throughput. HDFS is designed to store and scan millions of rows of data and to count or add some subsets of the data. The time required in this process is dependent on the involved complexities.

Third, HDFS is suitable for applications with large data sets. Traditionally, HDFS has been designed to support high throughput of data sets in batch style jobs.

Fourth, HDFS is suitable for applications with streaming access to file system data. They are not targeted for general purpose applications. HDFS is designed more for batch processing rather than for interactive use. The emphasis is on high throughput of data access rather than low latency of data access.

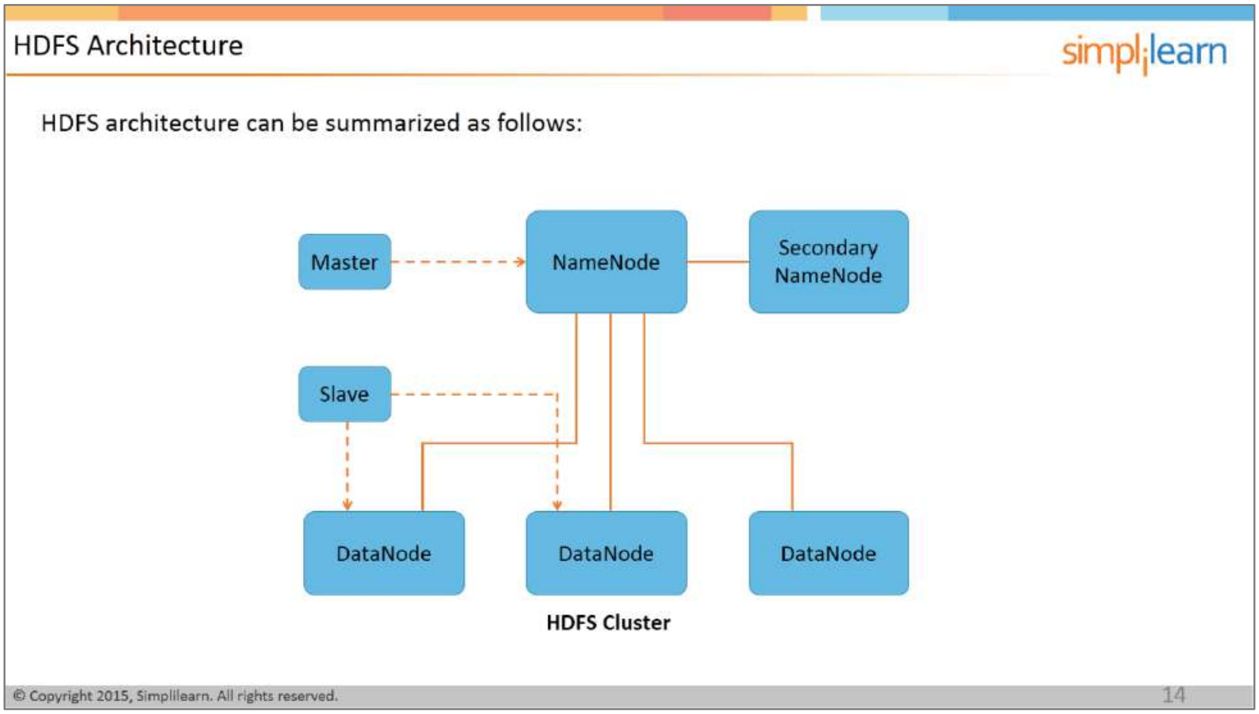
Finally, HDFS is designed in such a way that it can be built on commodity hardware and heterogeneous platforms.



HDFS—Key Features

The key features of HDFS are as follows:

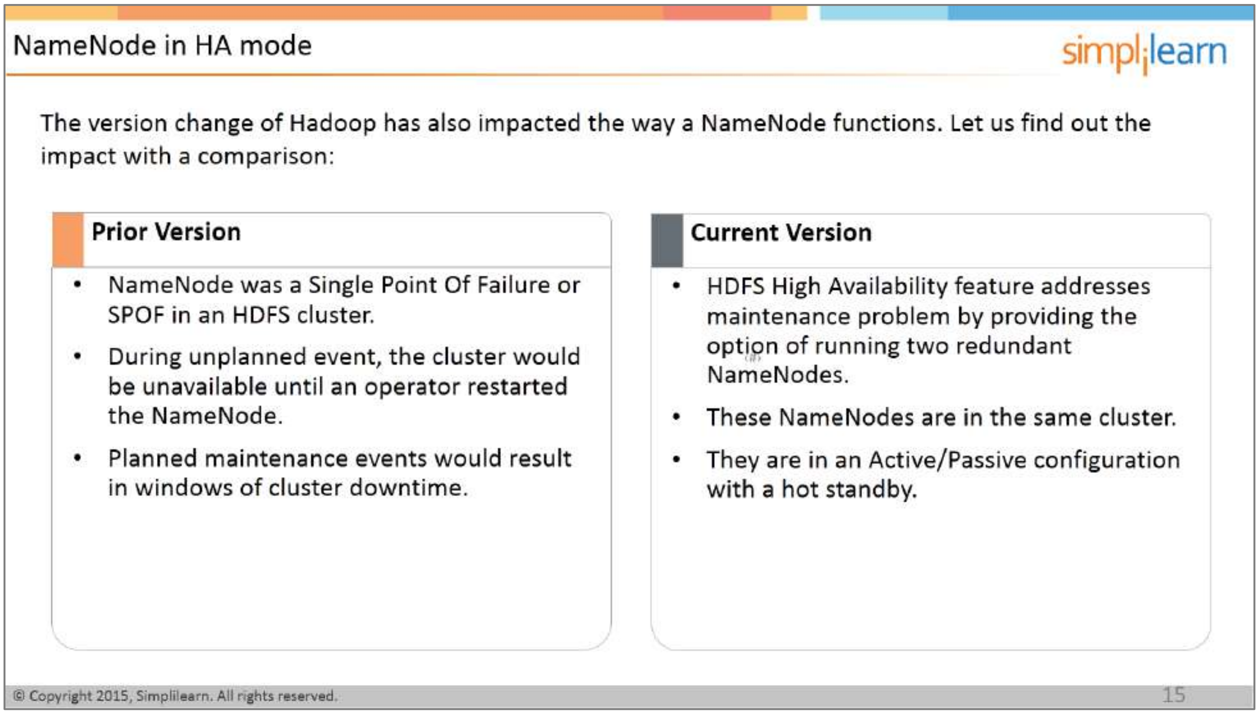
- It creates multiple replicas of each data block and distributes them on computers throughout a cluster to enable reliable and rapid data access.
- It is the storage system for both input and output of MapReduce jobs. This file system can be accessed by using the HDFS protocol to perform I/O operation on the file which is stored in HDFS.
- Block storage metadata controls the physical location of a block and its replication within the cluster.
- Each block is replicated to a few physically separate machines. However, the count can be modified by the administrator.



HDFS Architecture

As discussed earlier, HDFS architecture consists of: NameNode and Secondary NameNode services, which constitute the master services. The DataNode service falls under the slave service.

The master service is responsible for accepting a job from clients. It has the task of ensuring that the data required for the operation is loaded and segregated into chunks of data blocks. HDFS exposes a file system namespace and allows user data to be stored in files. A file is split into one or more blocks, stored, and replicated in DataNodes. The data blocks are then distributed to the DataNode systems within the cluster. This ensures that replicas of the data are maintained.



NameNode in HA mode

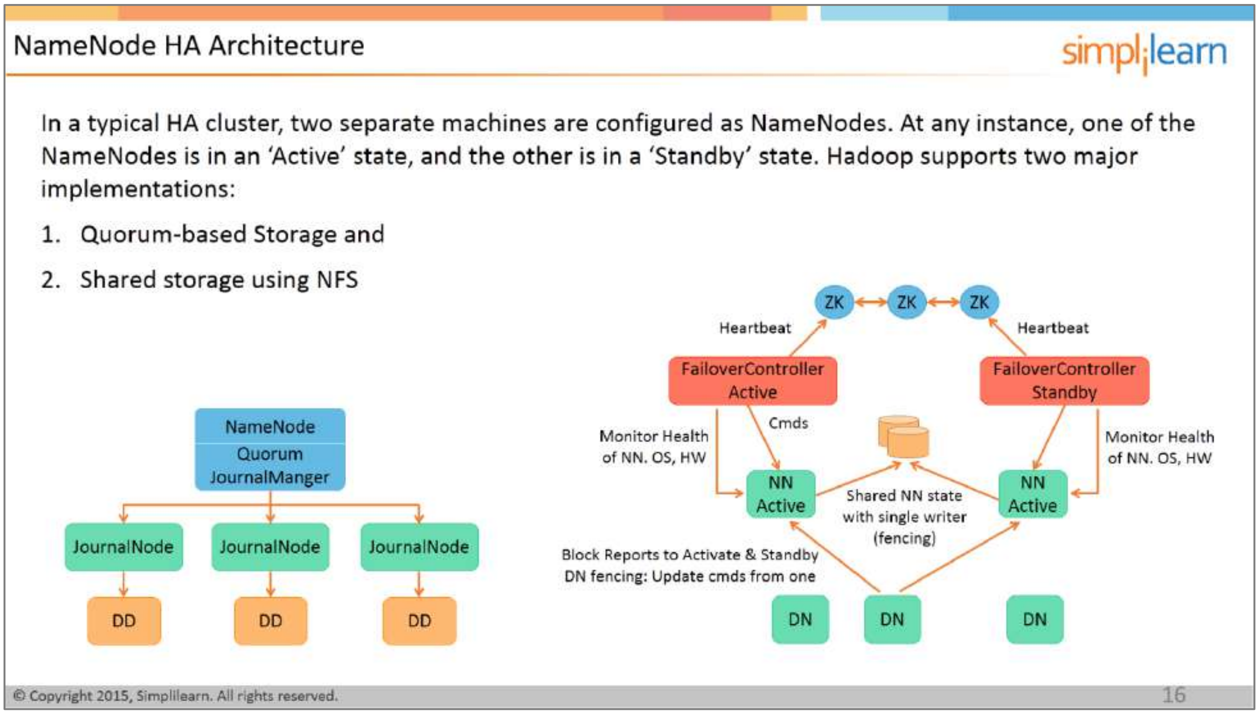
Prior to Hadoop 2.0.0, the NameNode was a Single Point Of Failure or SPOF in an HDFS cluster. Each cluster had a single NameNode, and if that machine or process became unavailable, the cluster as a whole would be unavailable until the NameNode was either restarted or brought up on a separate machine.

This impacted the total availability of the HDFS cluster in two major ways:

In the case of an unplanned event such as a machine crash, the cluster would be unavailable until an operator restarted the NameNode.

Planned maintenance events such as software or hardware upgrades on the NameNode machine would result in windows of cluster downtime.

The HDFS High Availability or HA feature addresses these problems by providing the option of running two redundant NameNodes in the same cluster in an Active/Passive configuration with a *hot standby*. This allows a fast failover to a new NameNode in the case that a machine crashes, or a *graceful* administrator-initiated *failover* for the purpose of planned maintenance.



NameNode HA Architecture

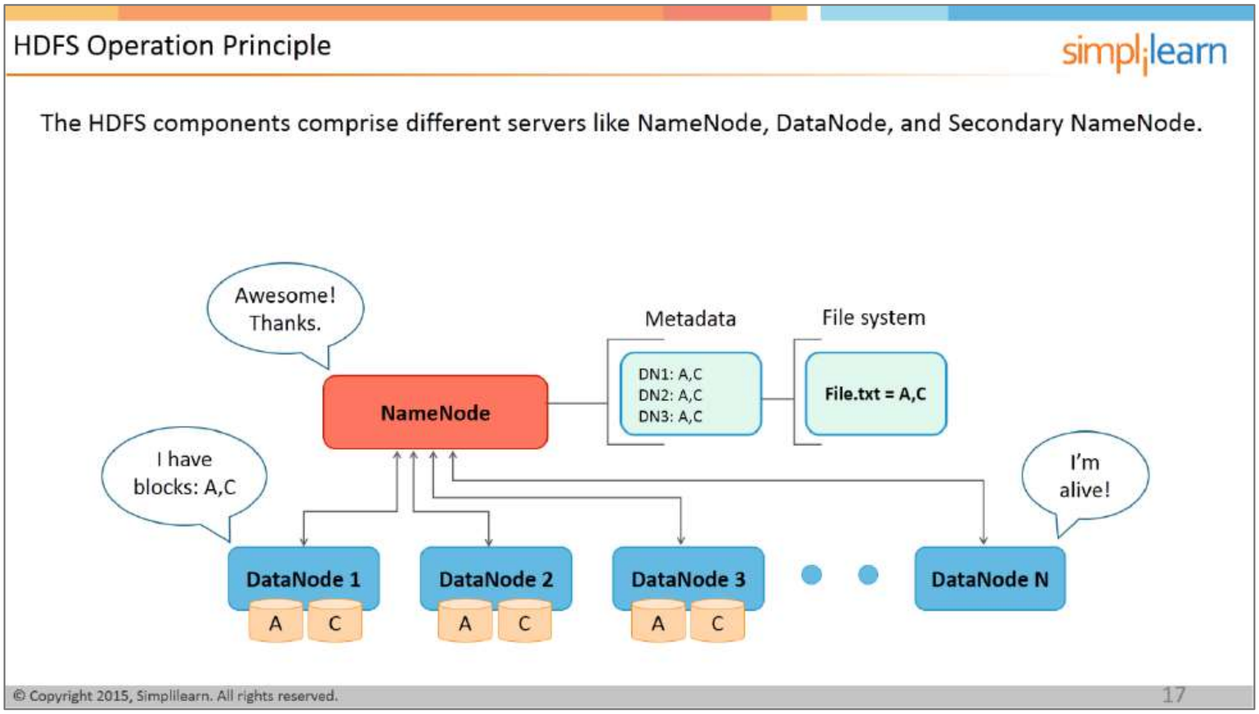
In a typical HA cluster, two separate machines are configured as NameNodes. At any instance, one of the NameNodes is in an 'Active' state, and the other is in a 'Standby' state. The Active NameNode is responsible for all client operations in the cluster, while the Standby is simply acting as a slave, maintaining enough state to provide a fast failover if necessary. Hadoop supports two major implementations:

1. Quorum-based Storage and

2. Shared storage using NFS

Quorum-based Storage refers to the HA implementation that uses Quorum Journal Manager or QJM. For the Standby node to keep its state synchronized with the Active node in this implementation, both nodes communicate with a group of separate daemons called JournalNodes. When any namespace modification is performed by the Active node, it durably logs a record of the modification to a majority of these JournalNodes. The Standby node is capable of reading the edits from the JournalNodes, and is constantly watching them for changes to the edit log. As the Standby Node sees the edits, it applies them to its own namespace. In the event of a failover, the Standby will ensure that it has read all the edits from the JournalNodes before promoting itself to the Active state. This ensures that the namespace state is fully synchronized before a failover occur

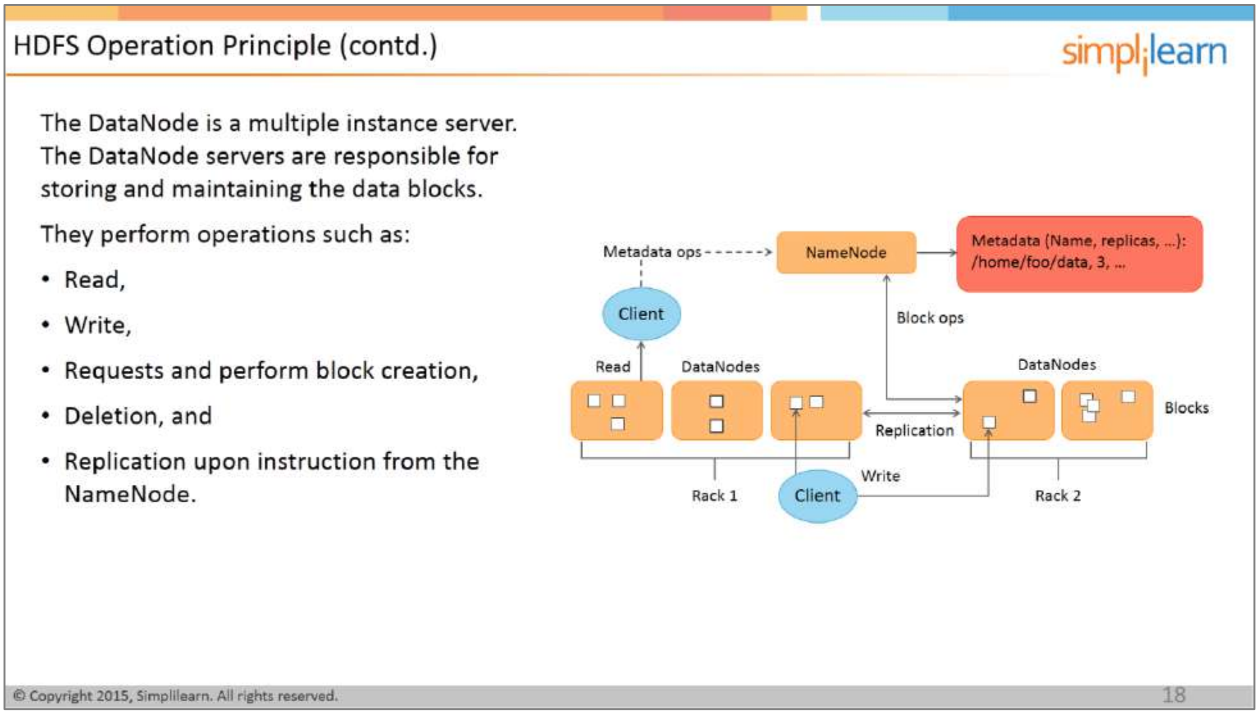
In Shared Storage Using NFS for the Standby node to keep its state synchronized with the Active node, this implementation requires both the nodes to have access to a directory on a shared storage device for example, an NFS mount from a NAS.



HDFS Operation Principle

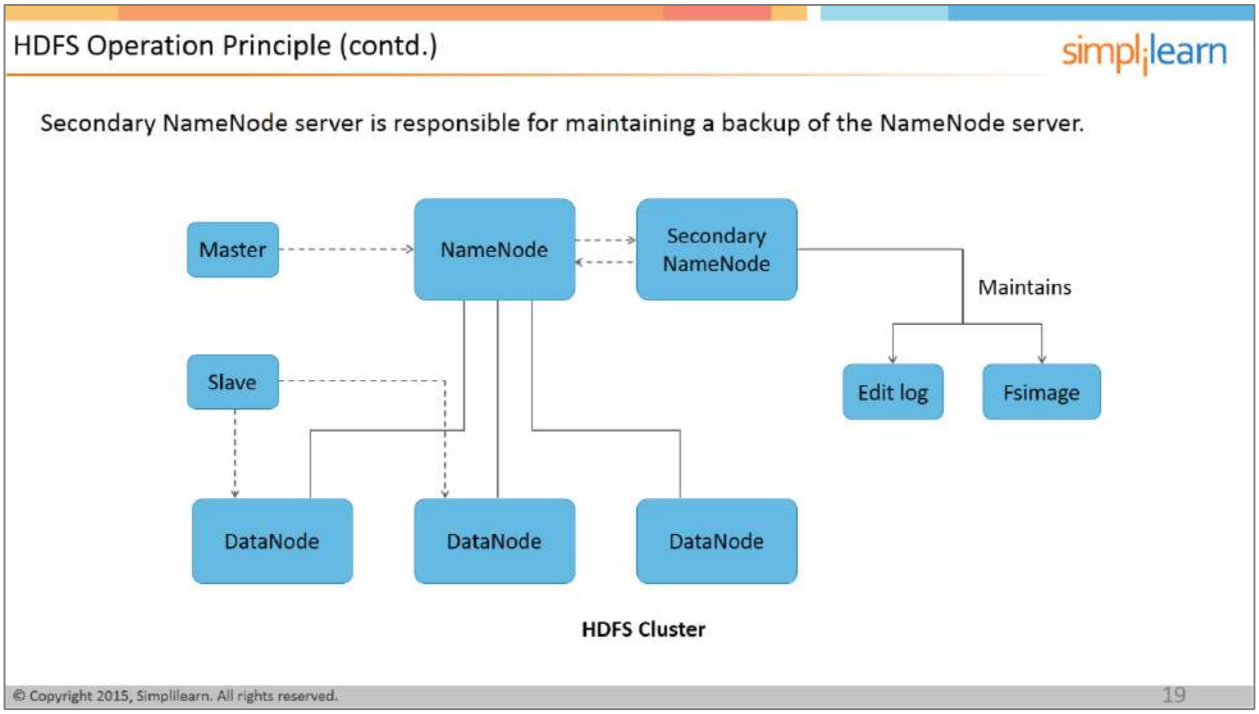
HDFS components include different servers such as NameNode, DataNode, and Secondary NameNode server.

The NameNode server is the core component of an HDFS cluster. There can be just one NameNode server in an entire cluster. It is responsible for maintaining the file system namespace. The NameNode helps to execute file system namespace operations like opening, closing, and renaming of files and directories which are present in HDFS. Information of the data and the metadata is stored in the namespace image and the edit log. It also determines the linking of blocks to DataNodes. Furthermore, the NameNode is aware of the DataNode status; that is, it knows the DataNode in which certain data is stored. It is also a critical, single point of failure. If it fails, the entire cluster goes down. However, the NameNode server can be partially restored by using a Secondary NameNode server.



HDFS Operation Principle (contd.)

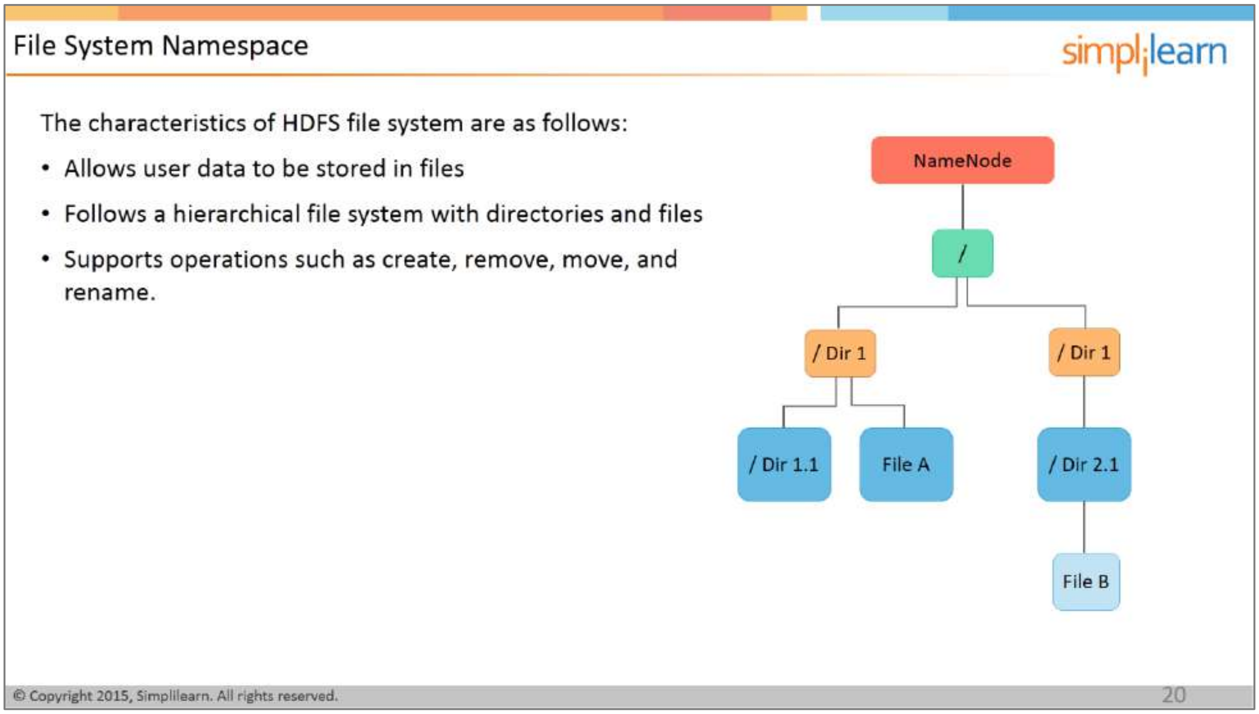
The DataNode is a multiple instance server. There can be 'n' number of DataNode servers depending on the type of networking and storage system. The DataNode servers are responsible for storing and maintaining the data blocks which are provisioned by the NameNode Server on the basis of the type of job submitted by the client. It is also responsible for storing and retrieving the blocks when referred by clients or the NameNode. Furthermore, the DataNode servers read, write requests, and perform block creation, deletion, and replication upon instruction from the NameNode.



HDFS Operation Principle (contd.)

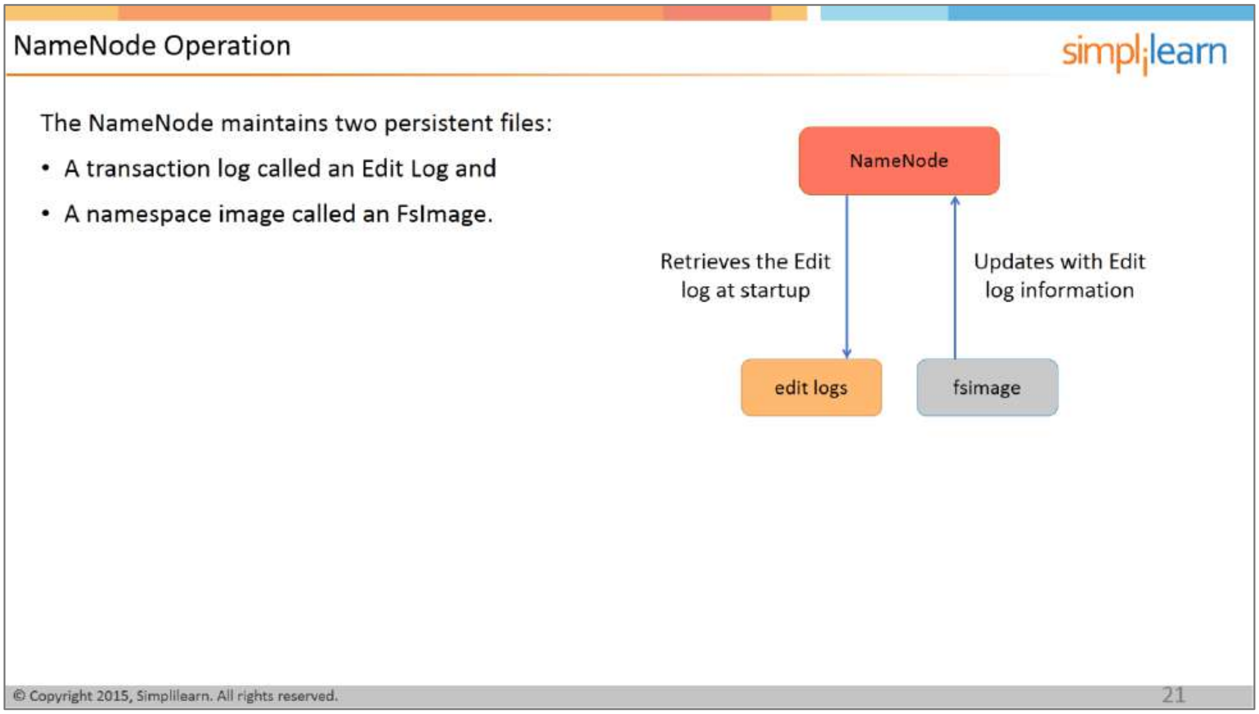
There can be only one Secondary NameNode server in a cluster. It is responsible for maintaining a backup of the NameNode server.

However, you cannot treat the Secondary NameNode server as a disaster recovery server. It will partially restore the NameNode server in case of failure. The Secondary NameNode server maintains the edit log and namespace image information which is synced with the NameNode server. At times, the namespace images from the NameNode server are not updated; therefore, you cannot totally rely on the Secondary NameNode server for the recovery process.



File System Namespace

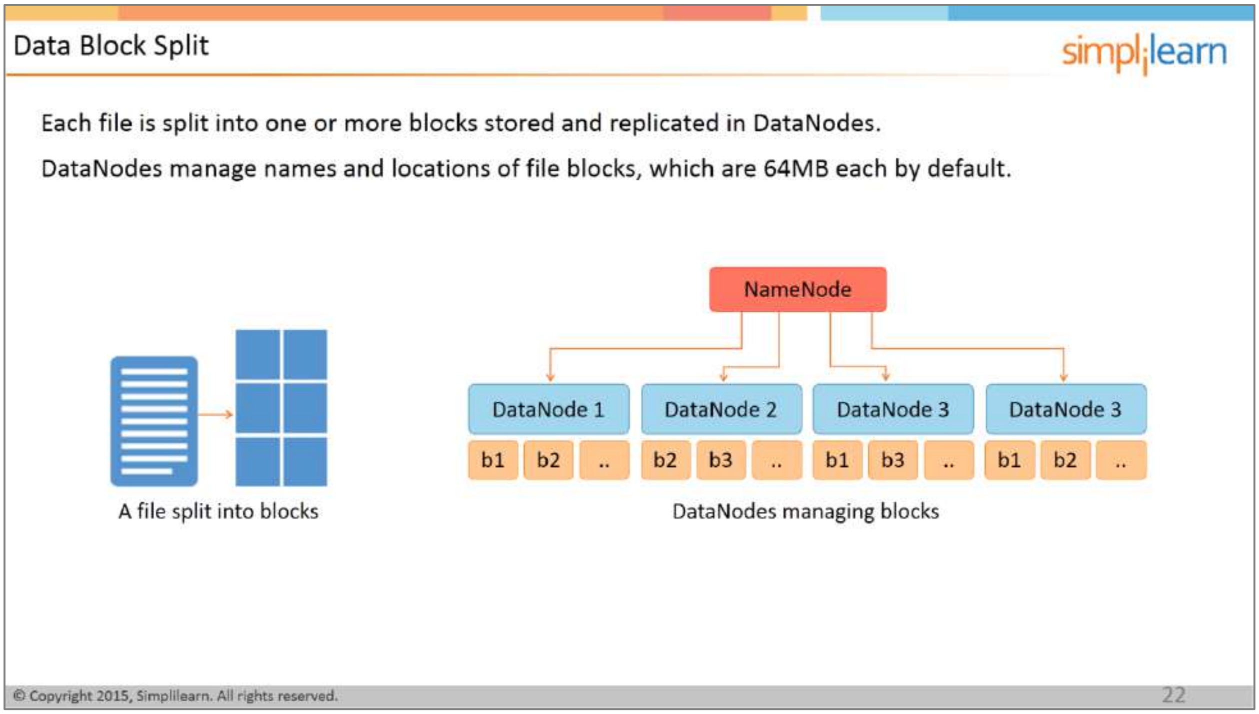
HDFS exposes a file system namespace and allows user data to be stored in files. HDFS has a hierarchical file system with directories and files. The NameNode manages the file system namespace, allowing clients to work with files and directory subtrees. A file system supports operations like create, remove, move, and rename. The NameNode, apart from maintaining the file system namespace, records any change to metadata information.



NameNode Operation

The NameNode maintains two persistent files – a transaction log called an Edit Log and a namespace image called an FsImage. The Edit Log records every change that occurs in the file system metadata such as creating a new file. The Edit Log is stored in the NameNode's local file system. The entire file system namespace including mapping of blocks, files, and file system properties is stored in FsImage. This is also stored in the NameNode's local file system.

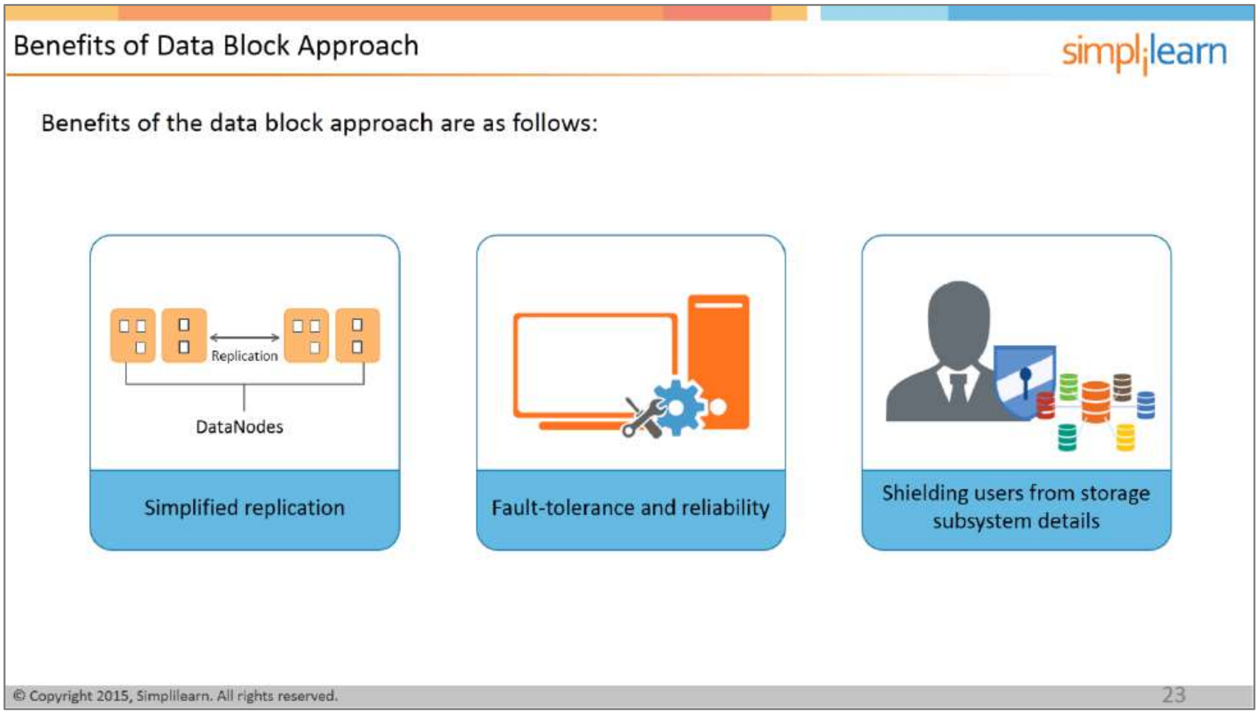
However, metadata loads into its memory the blocks of data that reside on a specific DataNodes at start up, when new DataNodes join a cluster. Metadata then periodically loads at user defined, or default intervals. When the NameNode starts up, it retrieves the Edit Log and FsImage from its local file system. It then updates the FsImage with Edit Log information, and stores a copy of the FsImage on the file system as a checkpoint. The metadata size is limited to the RAM available on the NameNode. A large number of small files would require more metadata than a small number of large files. Hence, the in-memory metadata management issue explains why HDFS favors a small number of large files. If a NameNode runs out of RAM, it will crash, and applications won't be able to use HDFS until the NameNode is online again.



Data Block Split

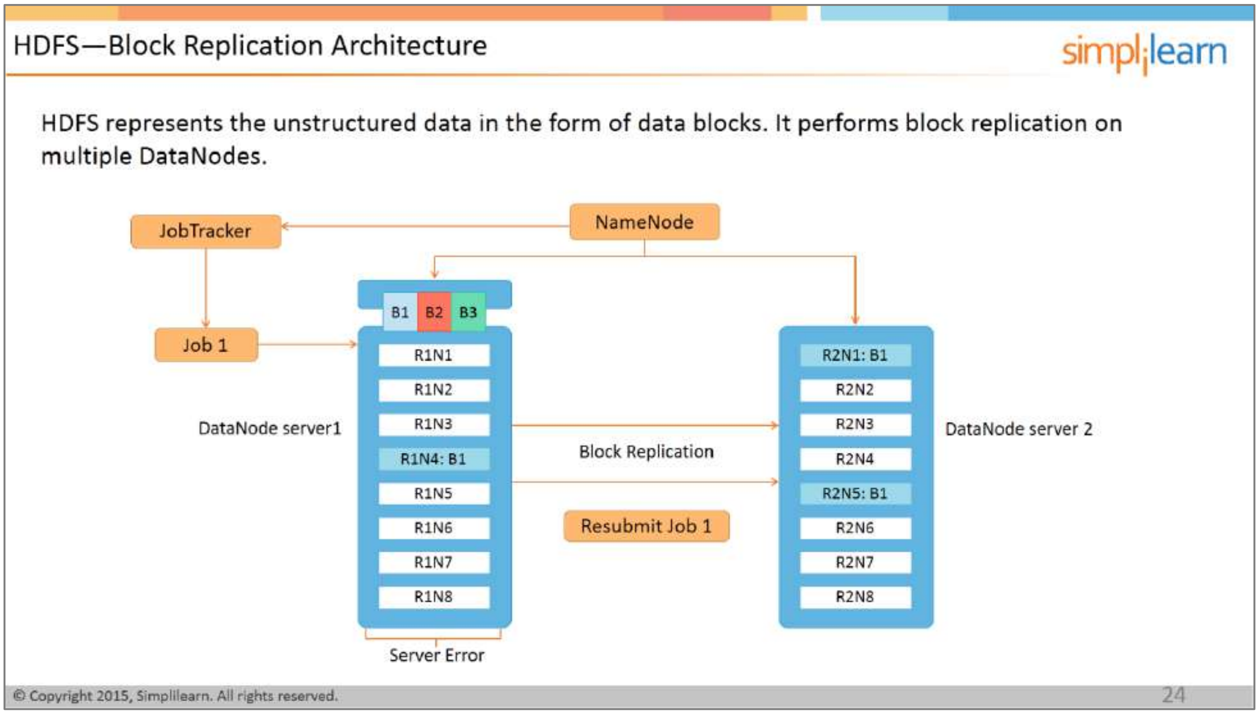
Each file is split into one or more blocks stored and replicated in DataNodes.

DataNodes manage names and locations of file blocks. Each file block is 64MB by default. For large files, it is better to increase the block size to 128MB. Doing so decreases pressure on the NameNode's memory. However, this potentially reduces the amount of parallelism that can be achieved, as the number of blocks per file decreases. Each map task operates on one block, so if tasks are lesser than nodes in the cluster, the jobs will run slowly. However, this becomes a lesser issue when the average MapReduce job involves more files or larger individual files.



Benefits of Data Block Approach

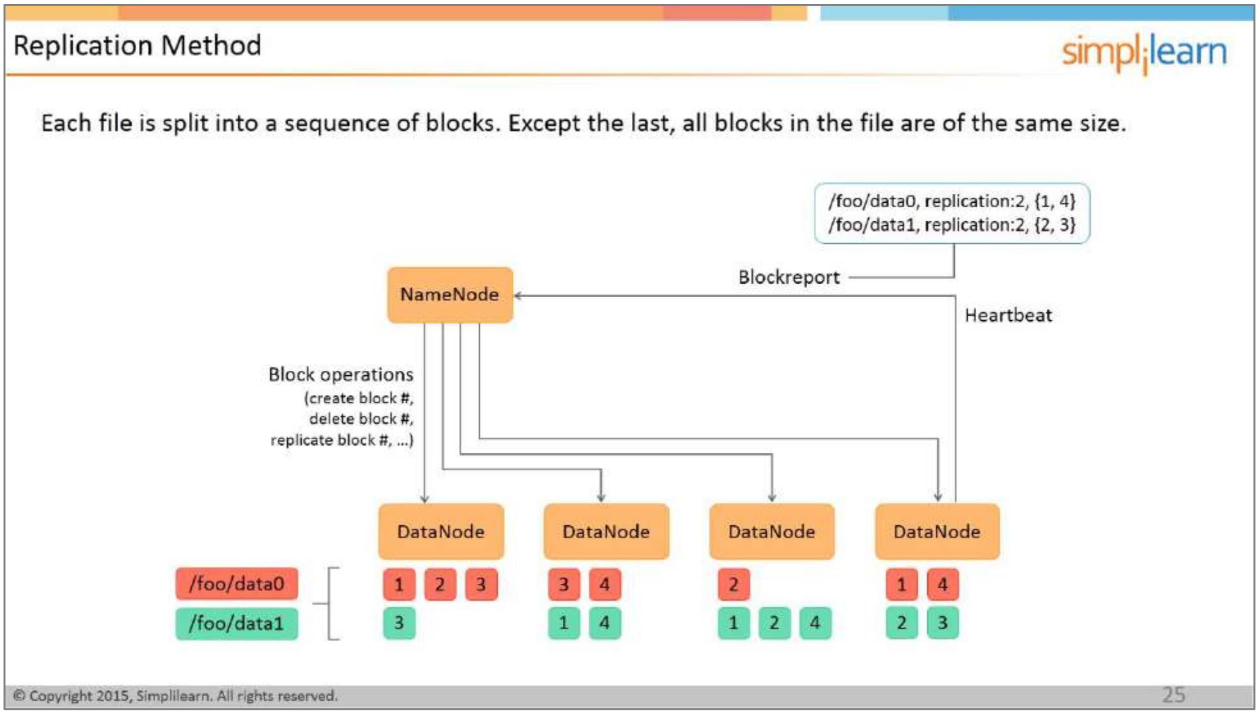
Some benefits of the data block approach include simplified replication, fault-tolerance and reliability, and shielding users from storage subsystem details.



HDFS—Block Replication Architecture

In HDFS, unstructured data is represented in the form of data blocks. If there is a default size for the data blocks, it can be reset by the user or the administrator.

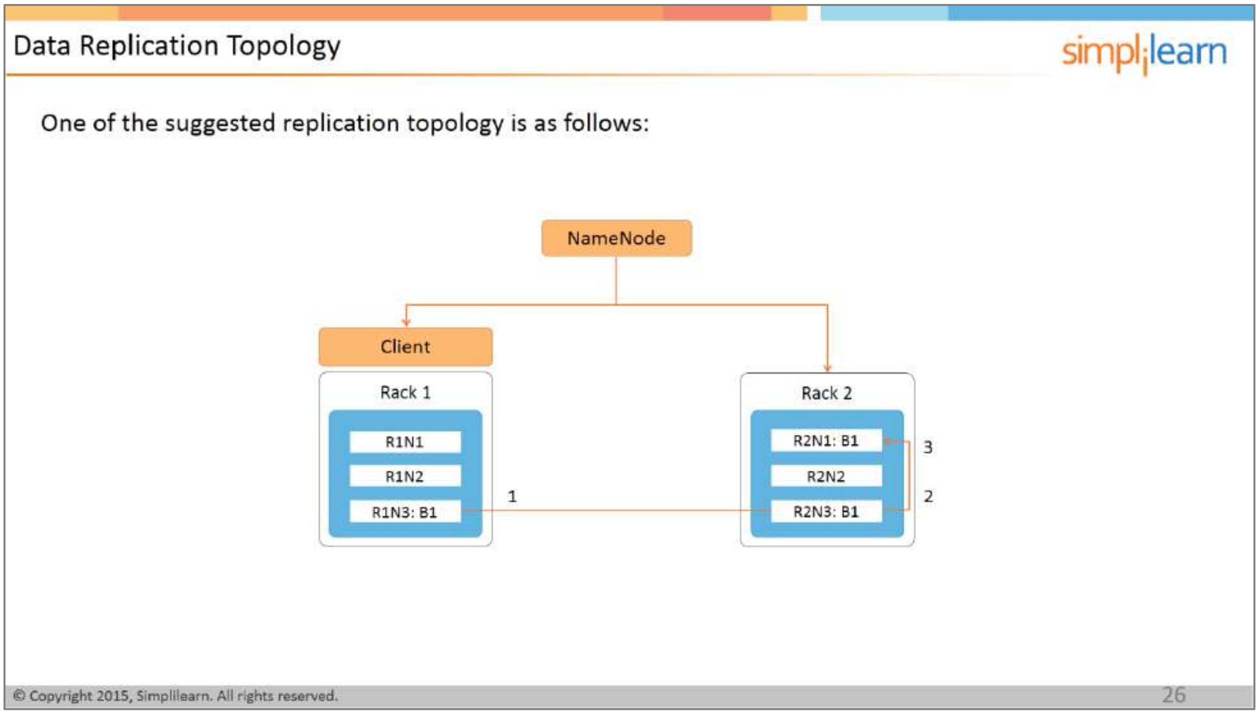
Block replication refers to the creation of replicas of a block in multiple DataNodes. Usually, the data is split in the form of parts, such as part-0 and part-1. HDFS performs block replication on multiple DataNodes, so that if any error exists on one DataNode server, the JobTracker service which is present in the NameNode server will resubmit the job to another DataNode server.



Replication Method

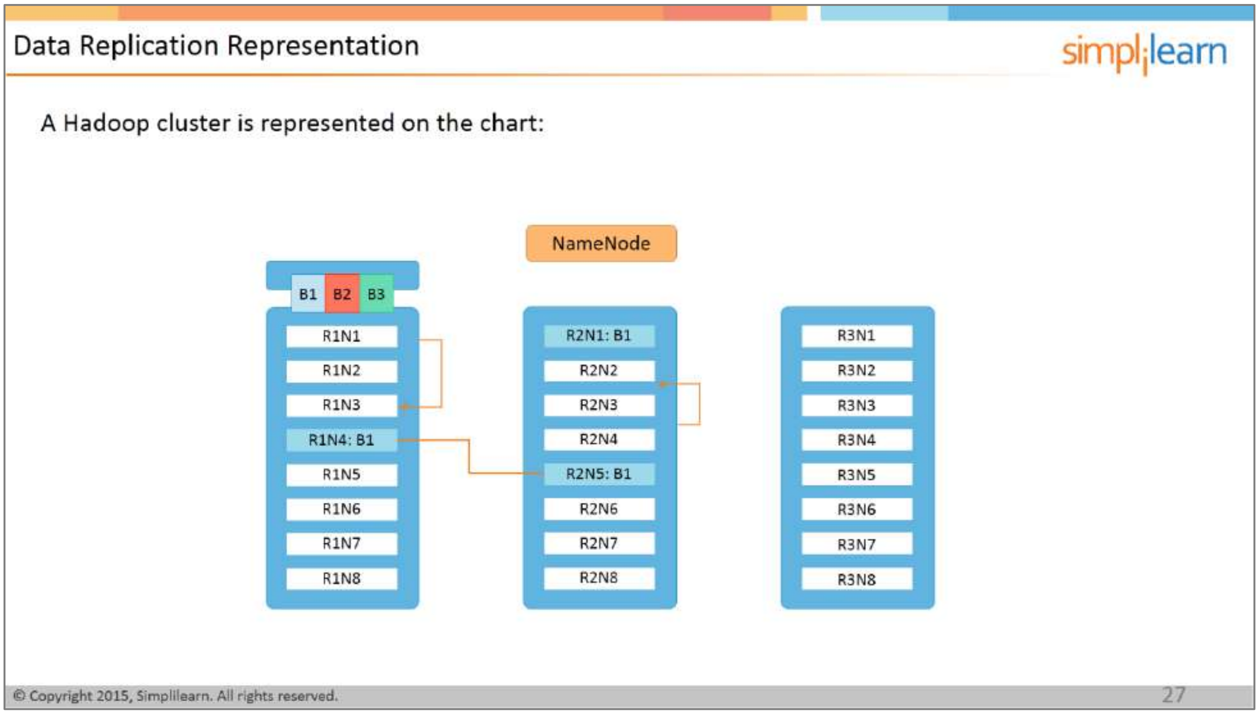
Each file is split into a sequence of blocks. Except the last, all blocks in the file are of the same size.

Blocks are replicated for fault-tolerance. The block replication factor is usually configured at the cluster level, but it can also be configured at the file level. The NameNode receives a *heartbeat* and a block report from each DataNode in the cluster. A block report contains all of the blocks on a DataNode. An application can also specify the number of replicas of the file needed, such as the replication factor of the file. This information is stored in the NameNode.



Data Replication Topology

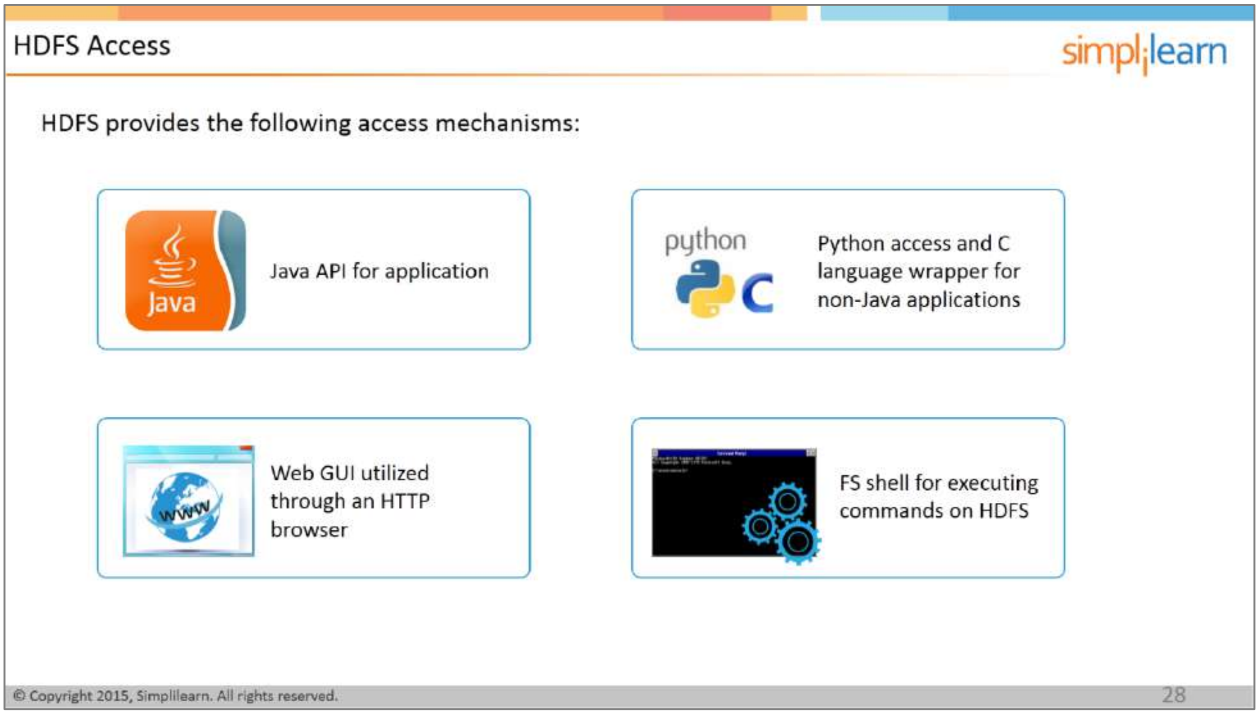
One of the suggested replication topologies could be having the first replica placed on the same node as the client. The second replica is placed on a rack different from the first rack. The third replica is placed on the same rack as the second but on a different node.



Data Replication Representation

On the chart shown, you can see a Hadoop cluster with three racks represented by dark blue pillars. Each rack contains multiple nodes represented by rectangular boxes. For example, you will see a box labeled R1N1 which represents Node 1 on Rack 1. To make it simple, each rack shown here has 8 nodes. At the top, you will see the NameNode shown in yellow.

Follow the path of a single data block – block B1, shown in purple on this chart. B1 is first written to Node 4 on Rack 1. A copy is then written on a different rack – Rack 2. More specifically, it is written to Node 5 on Rack 2. The third and final copy of the block is written to the same rack as the second copy – Rack 2 – but to a different node-- Node 1.



HDFS Access

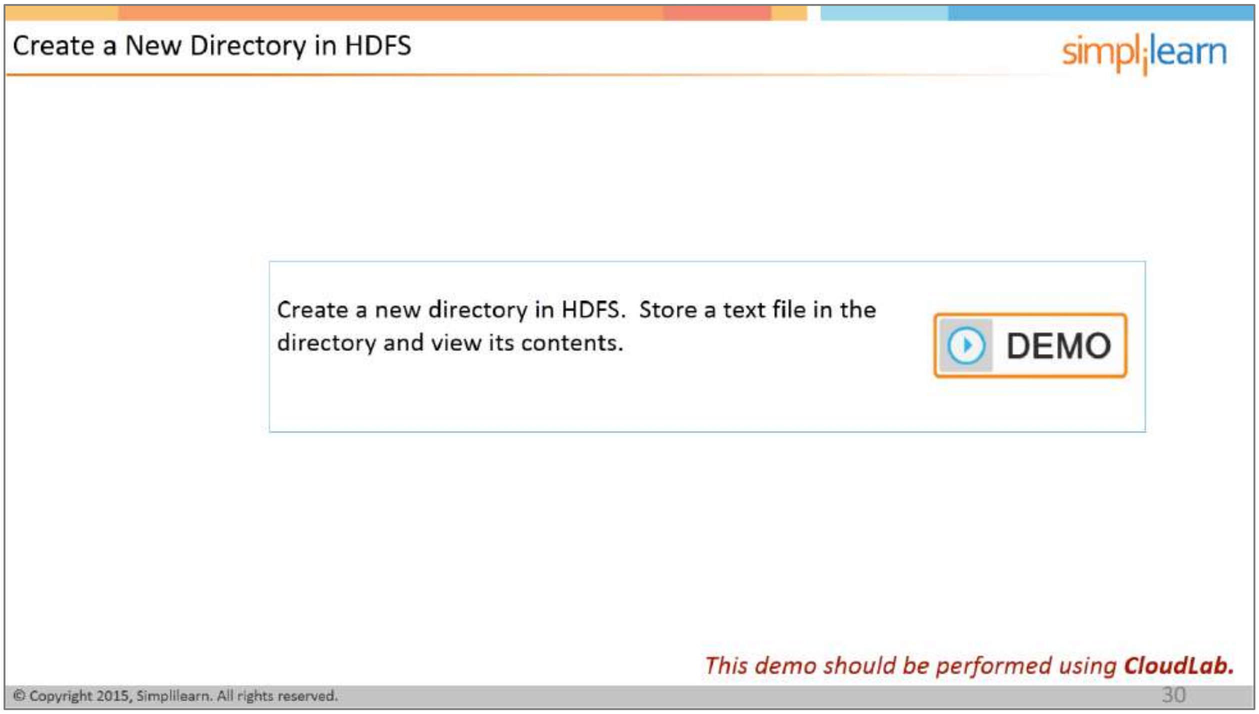
HDFS provides various access mechanisms. A Java API can be used for applications. There is also a Python and C language wrapper for non-Java applications. A Web GUI can also be utilized through an HTTP browser. An FS Shell is available for executing commands on HDFS.

Business Scenario

Olivia Tyler is the EVP of IT operations with Nutri Worldwide, Inc., and she has decided to use HDFS for storing Big Data. She will use HDFS shell to store the data in a Hadoop file system, and she will execute various commands on it.

The demos in the subsequent screens illustrate how to do basic command line operations in HDFS.

 29



Creating a New Directory in HDFS

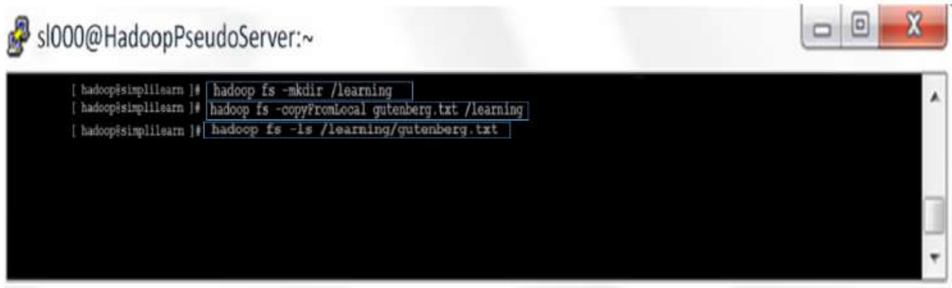
Let us do a quick recap of the steps performed:

1. Create a directory, named **learning**, in HDFS.
2. Copy the file **gutenberg.txt** to the directory created.
3. List the contents of the **learning** directory that are to be verified.
4. View the file contents using the required **cat** command.
5. Observe the file contents.

Spot the Error

Richard is a Hadoop developer in MCB technologies. He creates a new directory in HDFS and stores some files in it. When he tries to view the content of the file he gets an error message. Can you help him identify the error?

Spot the incorrect code.



Spot the Error

The incorrect code is: `hadoop fs -ls /learning/gutenberg.txt`

The correct code is: `[ hadoop@simplilearn ]# hadoop fs -cat /learning/gutenberg.txt`

QUIZ 1

What are the two major components of a Hadoop cluster?

a. Hadoop file system, NodeManager

b. MapReduce, Hadoop file system

c. ResourceManager, MapReduce

d. ApplicationMaster, ApplicationManager

 35

QUIZ 2

Which of the following services run in the Master node of Apache Hadoop in cluster mode (fully distributed mode)?

a. ResourceManager

b. ApplicationMaster

c. NodeManager, MapReduce

d. All of these

 37

QUIZ 3

Which are the single instance critical tasks?

a. NameNode, DataNode

b. ResourceManager, DataNode

c. NameNode

d. NameNode, Secondary NameNode

QUIZ 4

How is NameNode failure in non HA mode tackled?

a. Secondary NameNode is switched on as NameNode

b. Secondary NameNode automatically starts working as NameNode

c. From Secondary NameNode image backup, primary NameNode is recreated

d. Another NameNode in cluster with replication works as main NameNode

QUIZ 5

Which of the following statements best describes how a large (100 GB) file is stored in HDFS?

a. The file is replicated three times by default. Each copy of the file is stored on a separate DataNode.

b. The master copy of the file is stored on a single DataNode. The replica copies are divided into fixed-size blocks which are stored on multiple DataNodes.

c. The file is divided into fixed-size blocks which are stored on multiple DataNodes. Each block is replicated three times by default. Multiple blocks from the same file might reside on the same DataNode.

d. The file is divided into fixed-size blocks which are stored on multiple DataNodes. Each block is replicated three times by default. HDFS guarantees that different blocks from the same file are never on the same DataNode.

QUIZ 6

Which of the following describes how a client reads a file from HDFS?

a. The client queries the NameNode for the block location(s). The NameNode returns the block location(s) to the client. The client reads the data directly off the DataNode(s).

b. The client queries all DataNodes in parallel. The DataNode that contains the requested data responds directly to the client. The client reads the data directly off the DataNode.

c. The client contacts the NameNode for the block location(s).

d. The NameNode contacts the DataNode that holds the requested data block. Data is transferred from the DataNode to the NameNode and then from the NameNode to the client.

QUIZ 7

Which of the following are valid statements? (Choose two)

a. HDFS is optimized for storing a large number of files smaller than the HDFS block size.

b. HDFS has the characteristic of supporting a write once, read many data access model.

c. HDFS is a distributed file system that replaces ext3 or ext4 on Linux nodes in a Hadoop cluster.

d. HDFS is a distributed file system that runs on top of native OS file systems and is well-suited for storage of very large data sets.

QUIZ 8

The NameNode uses RAM for the following purpose:

a. To store the file contents in HDFS

b. To store filenames, list of blocks, and other meta information

c. To store the edits log that keeps track of changes in HDFS

d. To manage distributed read and write locks on files in HDFS

QUIZ 9

You need to move a file titled weblogs into HDFS. You are not allowed to copy the file. You know you have ample space on your DataNodes. Which action should you take to relieve this situation and store more files in HDFS?

a. Increase the block size on all current files in HDFS

b. Increase the block size on your remaining files

c. Decrease the block size on your remaining files

d. Increase the amount of memory for the NameNode

ANSWERS:

S.No.	Question	Answer & Explanation
1	What are the two major components of a Hadoop cluster?	b. The two major components of a Hadoop cluster are MapReduce and Hadoop File System.
2	Which of the following services run in the Master node of Apache Hadoop in cluster mode (fully distributed mode)?	a. ResourceManager service runs in the Master node of Apache Hadoop in YARN cluster mode.
3	Which are the single instance critical tasks?	c. There is only one NameNode that keeps block information like metadata for allocated storage on the data nodes in a non HA mode.
4	How is NameNode failure in non HA mode tackled?	c. NameNode failure in non HA mode is tackled by taking an image backup of the NameNode from Secondary NameNode and incorporating it into a new NameNode machine.
5	Which of the following statements best describes how a large (100 GB) file is stored in HDFS?	d. The file is divided into fixed-size blocks which are stored on multiple DataNodes. Each block is replicated three times by default. HDFS guarantees that different blocks from the same file are never on the same DataNode.
6	Which of the following describes how a client reads a file from HDFS?	d. The client contacts the NameNode for the block location(s). NameNode then queries the DataNodes for block locations. The DataNodes respond to the NameNode, and the NameNode redirects the client to the DataNode that holds the requested data block(s). The client then reads the data directly off the DataNode.

7	Which of the following are valid statements? (Choose two)	b. and d. HDFS has a characteristic of supporting a write once, read many data access model and a distributed file system that runs on top of native OS and is well-suited to storage of very large datasets.
8	The NameNode uses RAM for the following purpose:	b. NameNode uses RAM to store filenames, list of blocks, and other meta information.
9	You need to move a file titled weblogs into HDFS. You are not allowed to copy the file. You know you have ample space on your DataNodes. Which action should you take to relieve this situation and store more files in HDFS?	c. It is recommended to decrease the block size on your remaining files.

Case Study

Scenario | Analysis | Solution

XY Networks provides network security support to many organizations. It has system generated log files that are critical to security analysis and monitoring. These files are increasing in size and XY is running out of storage space. The company's expensive backup system is becoming obsolete. The cost estimate for system upgrade is approximately $5 million. Analysts suggest using Hadoop can reduce the storage cost by 90%.

54

Case Study

Scenario | Analysis | Solution

The company found that Hadoop is a popular solution for Big Data storage and processing. Hadoop has two components, HDFS for storage and MapReduce for processing.

Advantages of using HDFS for storage:

1. It can dump raw data logs into Hadoop.
2. Hadoop uses only hard disks for storage with RAID, which is inexpensive.
3. Developers can specify the number of replicas for each file.
4. HDFS maintains automatic backup for files.
5. It costs less than $1000 to store 1TB.

55

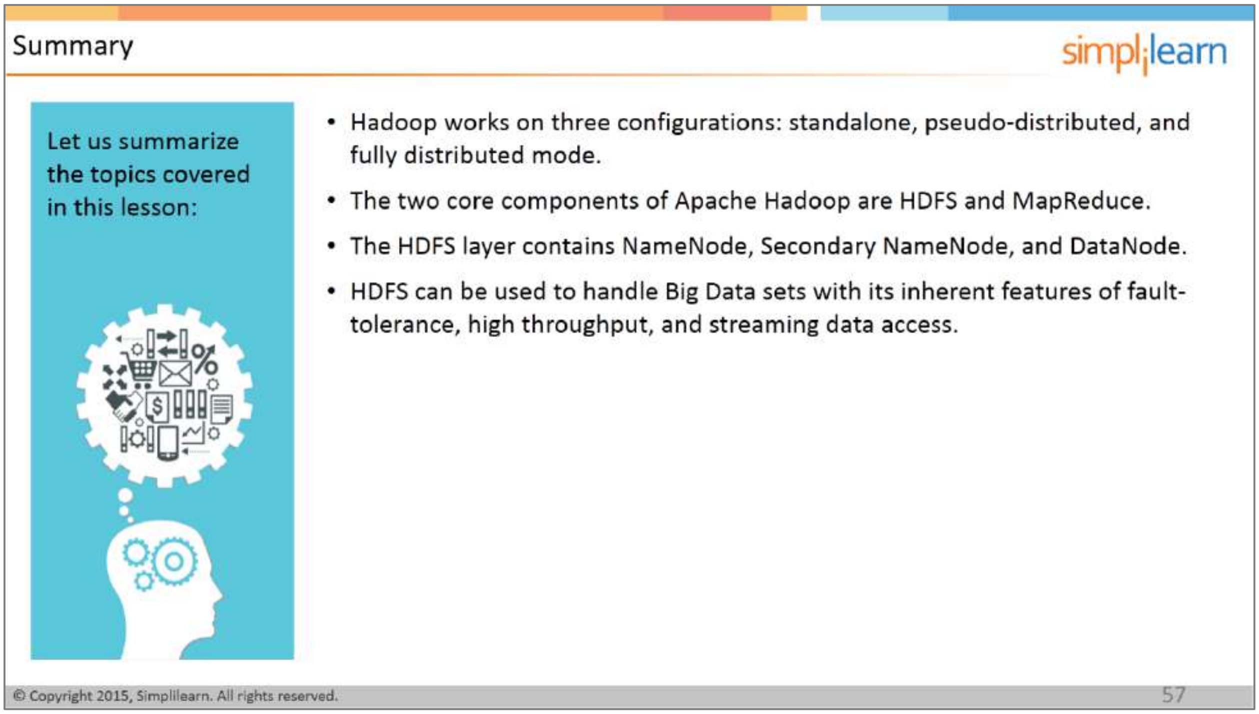
Case Study

Scenario Analysis Solution

Performed the following steps to explore HDFS storage in a cloud environment:

1. Consolidate some system log files.
2. Store the files in HDFS.
3. Check the files in HDFS and also the replication factor.
4. Increase the replication factor for critical files.
5. Recheck the files in HDFS to check the additional backup copies created.
6. Check the files through Hue.

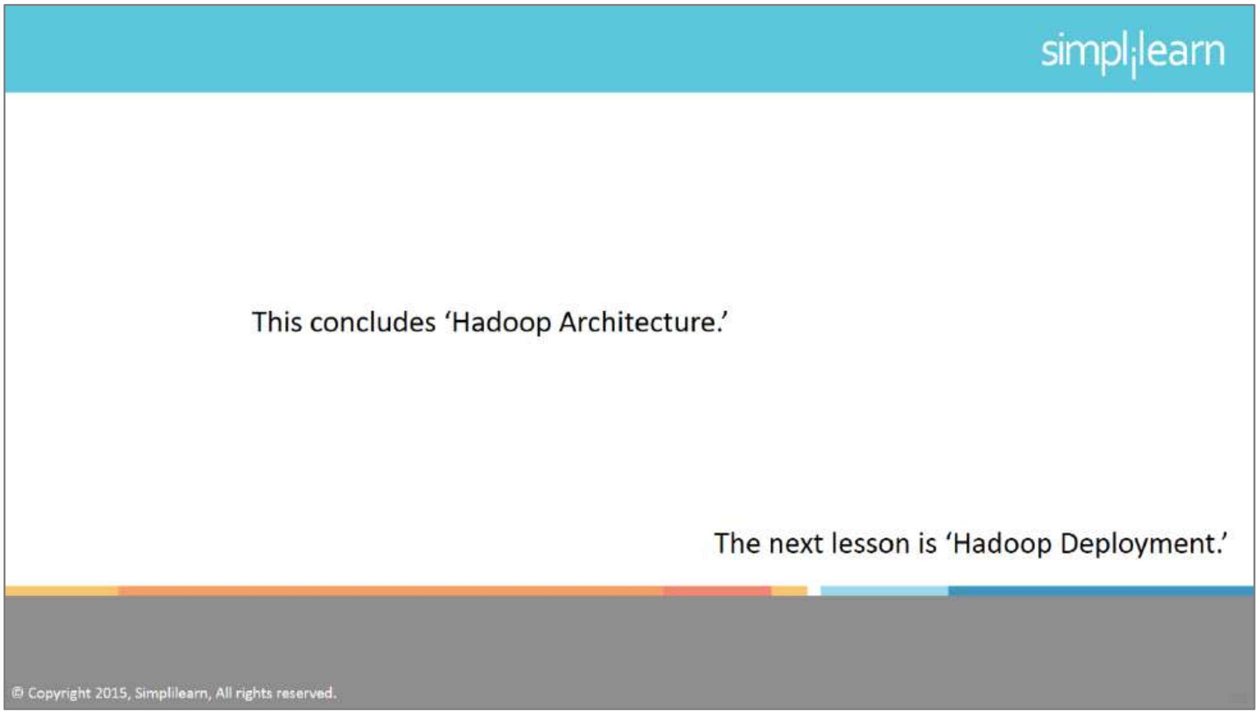
 56



Summary

Let us summarize the topics covered in this lesson:

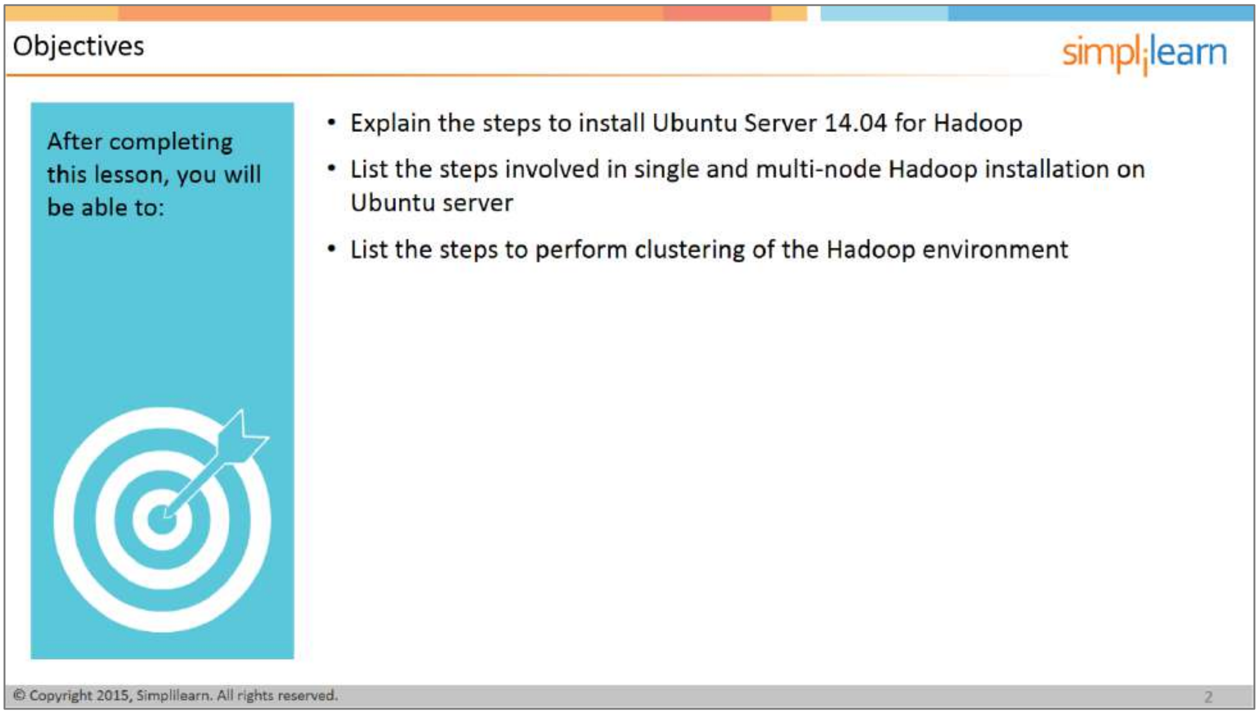
- Hadoop works on three configurations: standalone, pseudo-distributed, and fully distributed mode.
- The two core components of Apache Hadoop are HDFS and MapReduce.
- The HDFS layer contains NameNode, Secondary NameNode, and DataNode.
- HDFS can be used to handle Big Data sets with its inherent features of fault-tolerance, high throughput, and streaming data access.



Conclusion

This concludes 'Hadoop Architecture.' In the next lesson we will focus on 'Hadoop Deployment.'

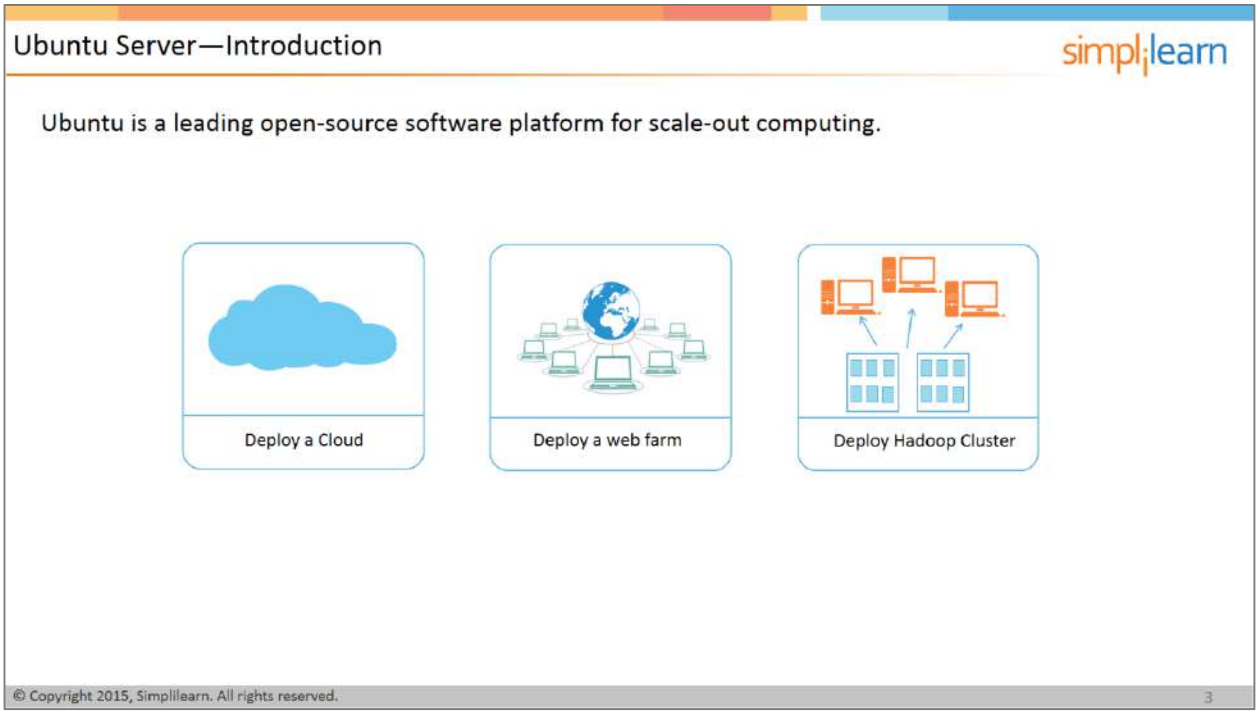
Lesson 3—Hadoop Deployment



Hadoop Deployment

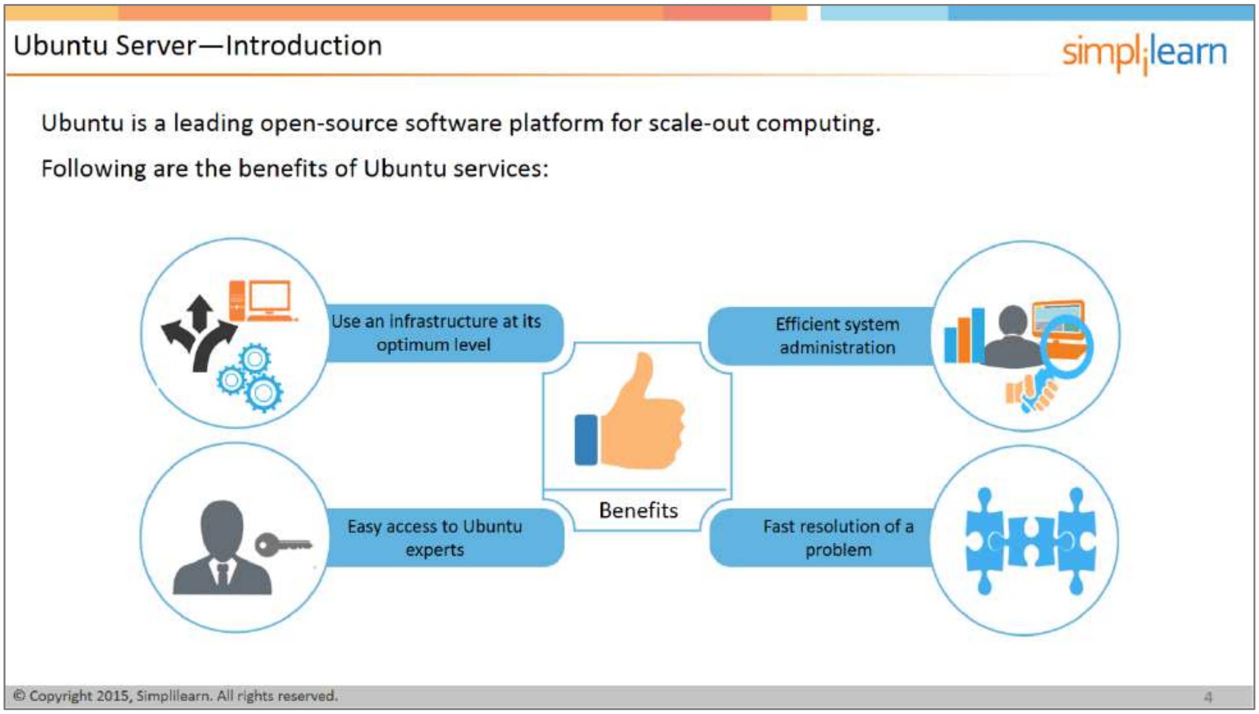
After completing this lesson, you will be able to:

- Explain the steps to install Ubuntu Server 14.04 for Hadoop
- List the steps involved in single and multi-node Hadoop installation on Ubuntu Server
- List the steps to perform clustering of the Hadoop environment



Ubuntu Server—Introduction

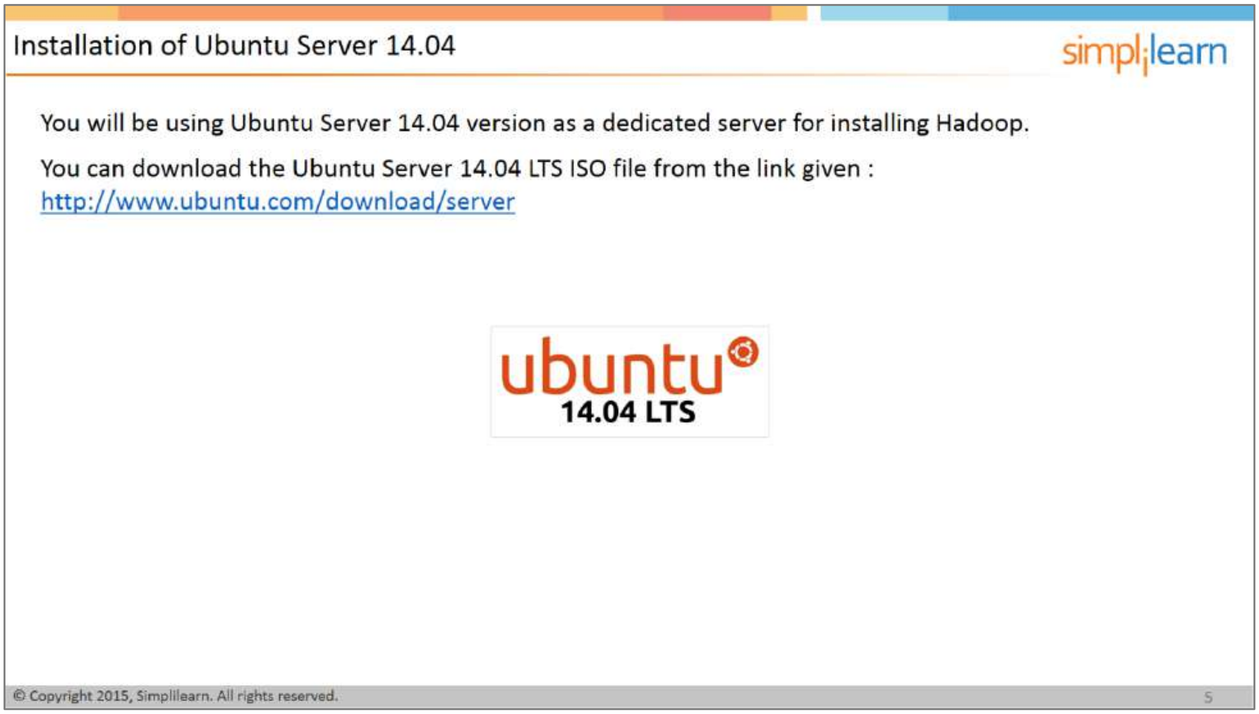
Ubuntu is a leading open-source software platform for scale-out computing. Ubuntu Server helps in the optimum utilization of infrastructure, irrespective of whether one wants to deploy a cloud, a web farm, or a Hadoop cluster.



Ubuntu Server—Introduction (contd.)

Following are the benefits of Ubuntu services:

- It has the required versatility and performance to utilize an infrastructure at its optimum level
- It ensures efficient system administration with landscape
- It provides easy access to Ubuntu experts
- It enables fast resolution of a problem



Installation of Ubuntu Server 14.04

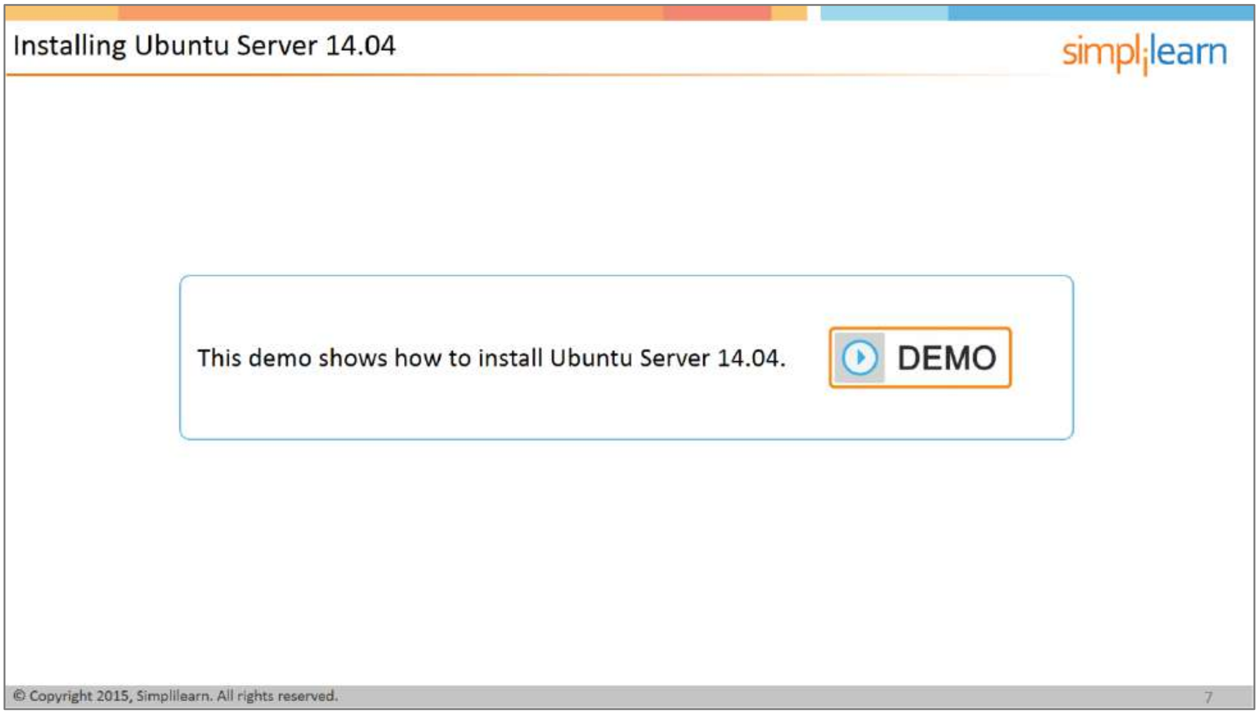
You will be using Ubuntu 14.04 version as a dedicated server for installing Hadoop. You can download the Ubuntu Server 14.04 LTS ISO file from the link: http://www.ubuntu.com/download/server.

Business Scenario

Nutri Worldwide, Inc. has adopted Hadoop for its operations. Olivia, the EVP at the company, has been assigned the task of setting up the infrastructure for Hadoop. Olivia's team has listed the following tasks to be performed:

- Installation of Ubuntu Server 14.04
- Installation of Hadoop in Ubuntu Server 14.04
- Performing clustering of the Hadoop environment

The demos in this lesson illustrate how to install Ubuntu Server 14.04, how to install Hadoop in Ubuntu Server 14.04, how to create a clone of Hadoop virtual machine, and how to perform the clustering of Hadoop environment.



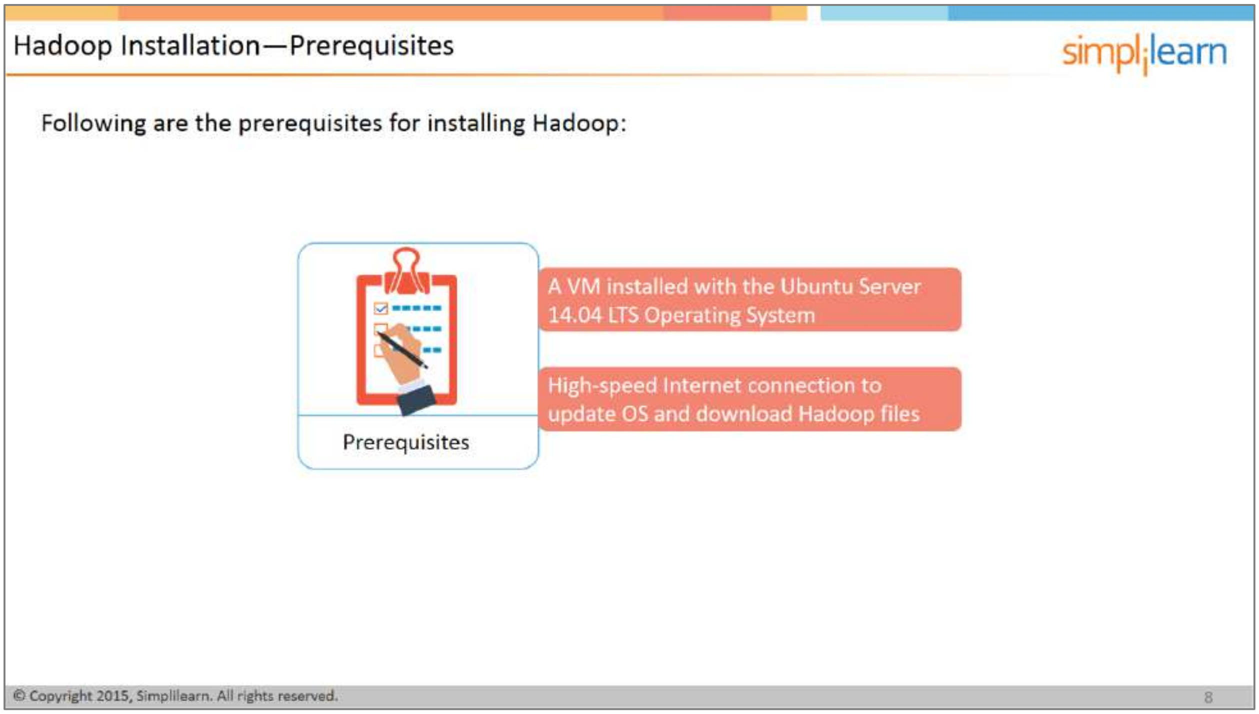
Installing Ubuntu Server 14.04

Let us summarize the steps in this demo:

This demo shows how to install Ubuntu Server 14.04.

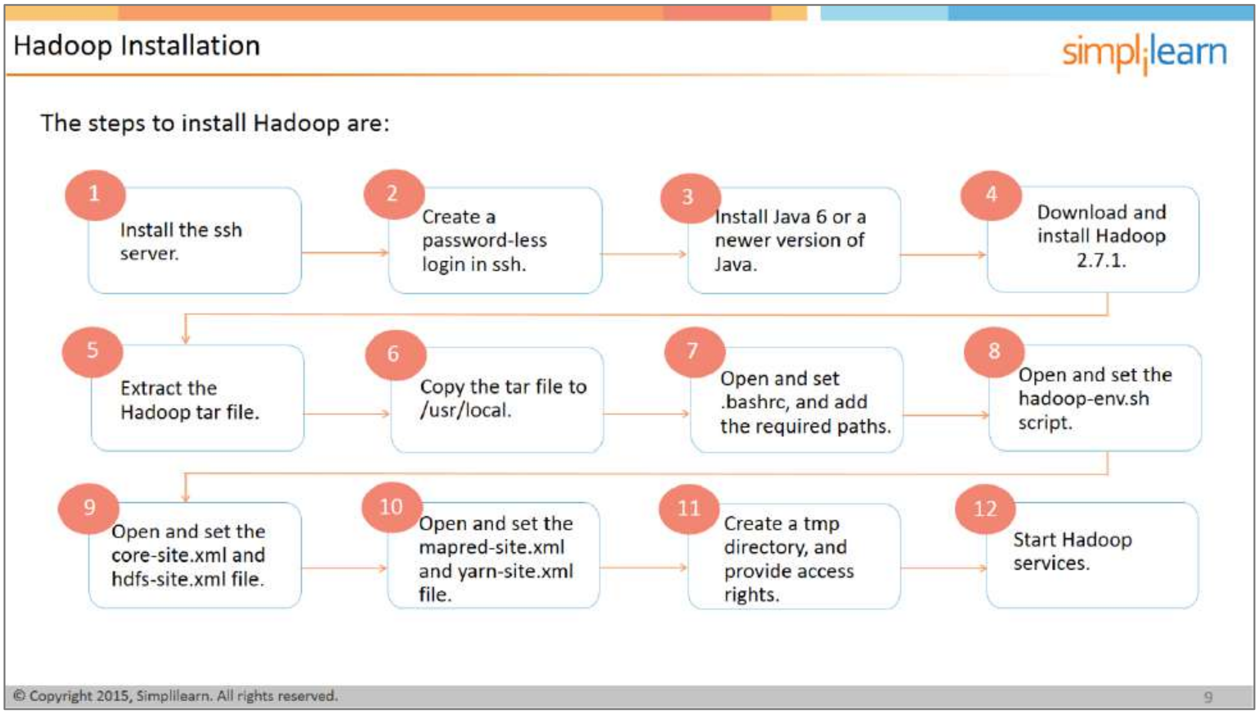
1. In this demo, let's look at an Ubuntu 14 installation in an Oracle Virtual Box VM. Click 'New' to begin.
2. The Create Virtual Machine window opens.
3. Type 'Ubuntu 14' as name of VM. Select Type as 'Linux.' And Version as 'Ubuntu 64bit.'
4. Set memory as 2GB and select 'create a Virtual Hard drive'.
5. The Create Virtual Hard Drive window opens. Select VMDk radio button and click Next.
6. You will find a new VM named Ubuntu 14 on the left pane. Select that VM and click 'Start' at the top.
7. The 'Select start-up disk' window opens.
8. Click the arrow button to select the disk image for Ubuntu.
9. Browse to the folder that containing 'ISO' image for Ubuntu.
10. Select the image and click Open.
11. Click 'Start' button. The Ubuntu 14 starting Virtual Machine window opens.
12. The Ubuntu 14– Oracle VM Virtual Box window opens.
13. Select the language as English.

14. Select Install Ubuntu Server.
15. Type the desired hostname in the Hostname: text field.
16. Enter the desired username under Username for your account.
17. Type the desired password under Choose a password for the new user.
18. Re-enter the password under Re-enter password to verify.
19. Press Enter to continue with the default settings of clock and disk Partitioning.
20. Select 'Guided– use entire disk and set up LVM'.
21. Select disk to partition. Use default option in most of the installation steps.
22. Type the HTTP proxy information.
23. Select Ubuntu upgrade method post installation.
24. Select 'Open SSH' from the list of software to install.
25. Install GRUB boot loader on a hard disk section appears.
26. Select <Continue> to re-boot the Ubuntu machine.
27. You will be prompted for User ID and Password.
28. You have successfully installed Ubuntu Server 14.04 LTS.



Hadoop Installation—Prerequisites

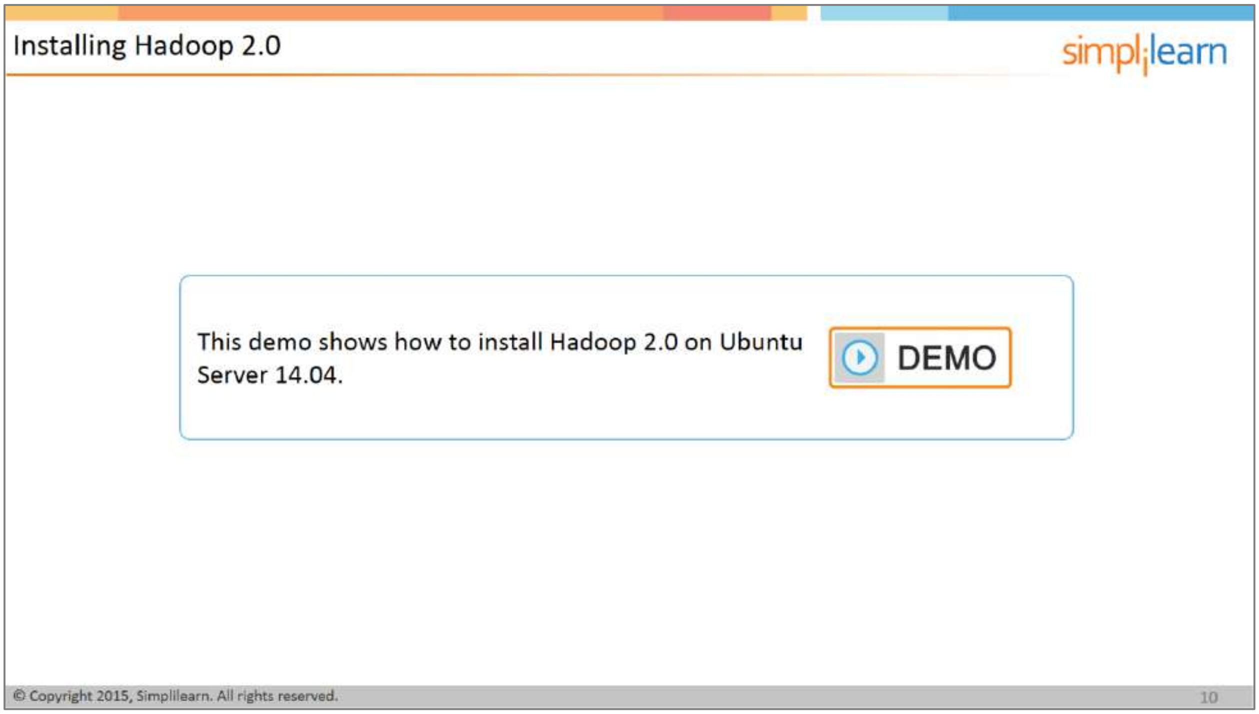
The prerequisites to install Hadoop are: A VM installed with the Ubuntu Server 14.04 LTS Operating System and high-speed Internet access to update the Operating System and to download the Hadoop files to the machine.



Hadoop Installation

Let’s install Hadoop version 2.0 in the Virtual Machine.

Follow the mentioned steps to successfully set up Hadoop in the Ubuntu Server 14.04 LTS.

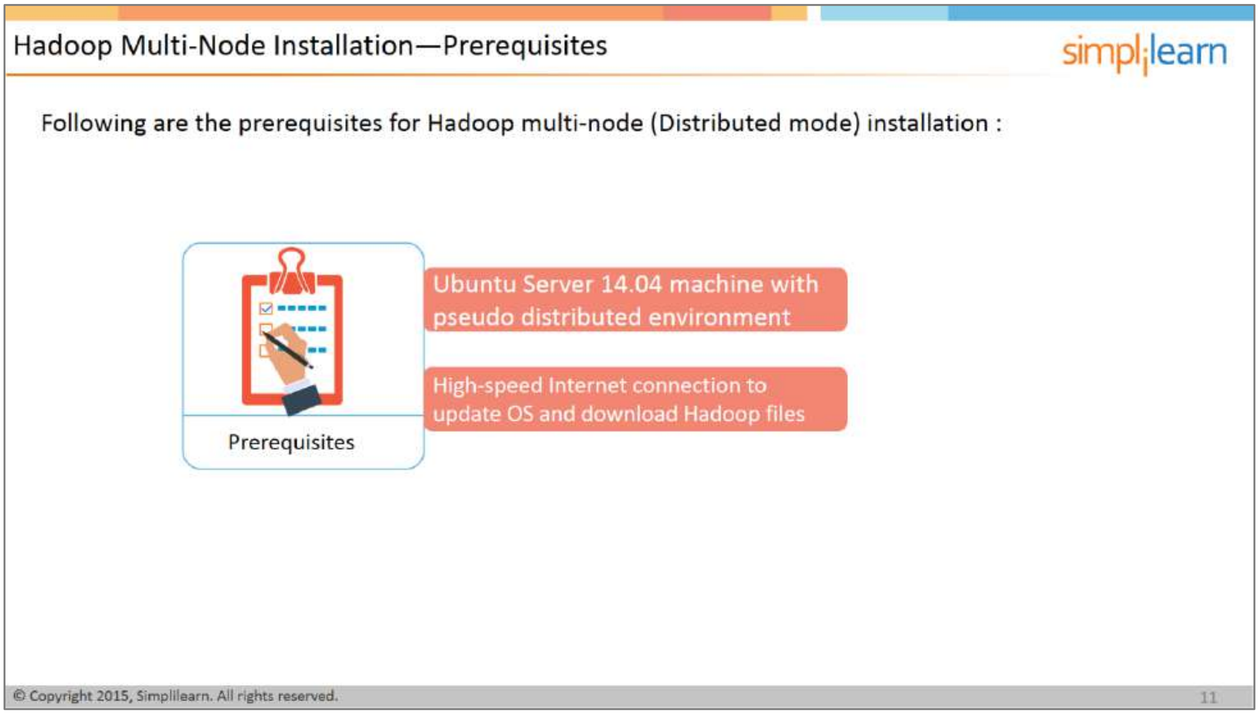


Installing Hadoop 2.0

This demo shows how to install Hadoop 2.0 in Ubuntu Server 14.04.

1. Before installing Hadoop, install pre-requisite software.
2. Update the apt repository to ensure the latest download references are available.
3. Ubuntu will update its apt-get repository definitions.
4. Install open JDK.
5. Verify the status of Java installation.
6. Install SSh server and rsync.
7. The pre-requisites software have been installed successfully.
8. Download the recent version of Hadoop package.
9. Once the package is downloaded, verify the download package.
10. Unpack the file using the command shown.
11. Create a soft link for Hadoop directory
12. Set environment variables for Hadoop in .bashrc file.
13. Enter JAVA _HOME path.
14. Add HADOOP_PREFIX to point Hadoop directory.
15. Add Hadoop's executables in the path. Add both bin and sbin Hadoop directory to the path.
16. Implement .bashrc changes using source command.
17. Run command 'hadoop' on shell prompt to verify the path settings.

18. The node use SSH key to communicate between them. To avoid password less communication, use SSH keys.
19. You will notice that SSH is now possible without any password.
20. Let us now start configuring Hadoop.
21. Verify the Hadoop sub directories at /usr/local/hadoop.
22. Edit configuration in ‘core-site.xml’, ‘hdfs-site.xml’, mapred-site.xml, and yarn-site.xml. Enter the configuration properties related to HDFS name node.
23. Also edit the hadoop-env.sh file.
24. Let us now initiate the name node and format. Start the DFS daemon and YARN daemon now.
25. You will notice various daemons running. You will also see the Resource Manager and Node Manager running along with the Name Node and Data Node.
26. Re-do HDFS directory listing.
27. You have successfully installed and configured Hadoop.

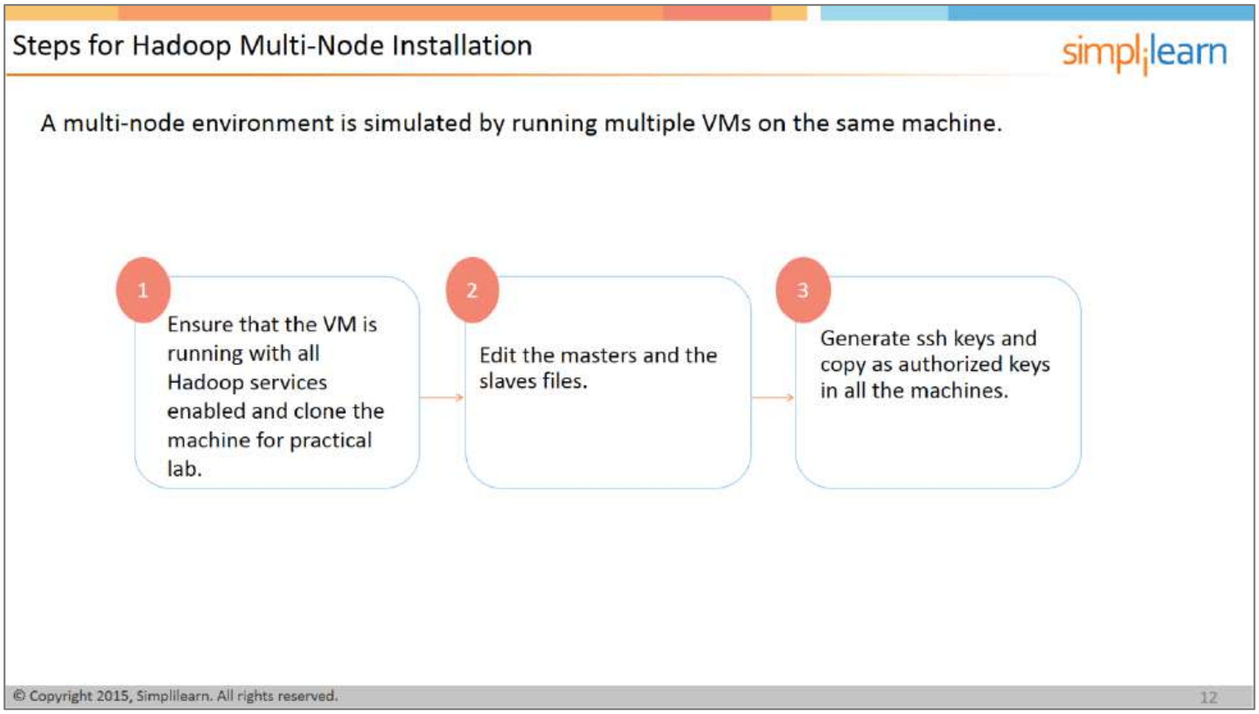


Hadoop Multi-Node Installation—Prerequisites

Let's now look at the Hadoop multi-node installation.

The Prerequisites to install Hadoop in distributed mode are as follows:

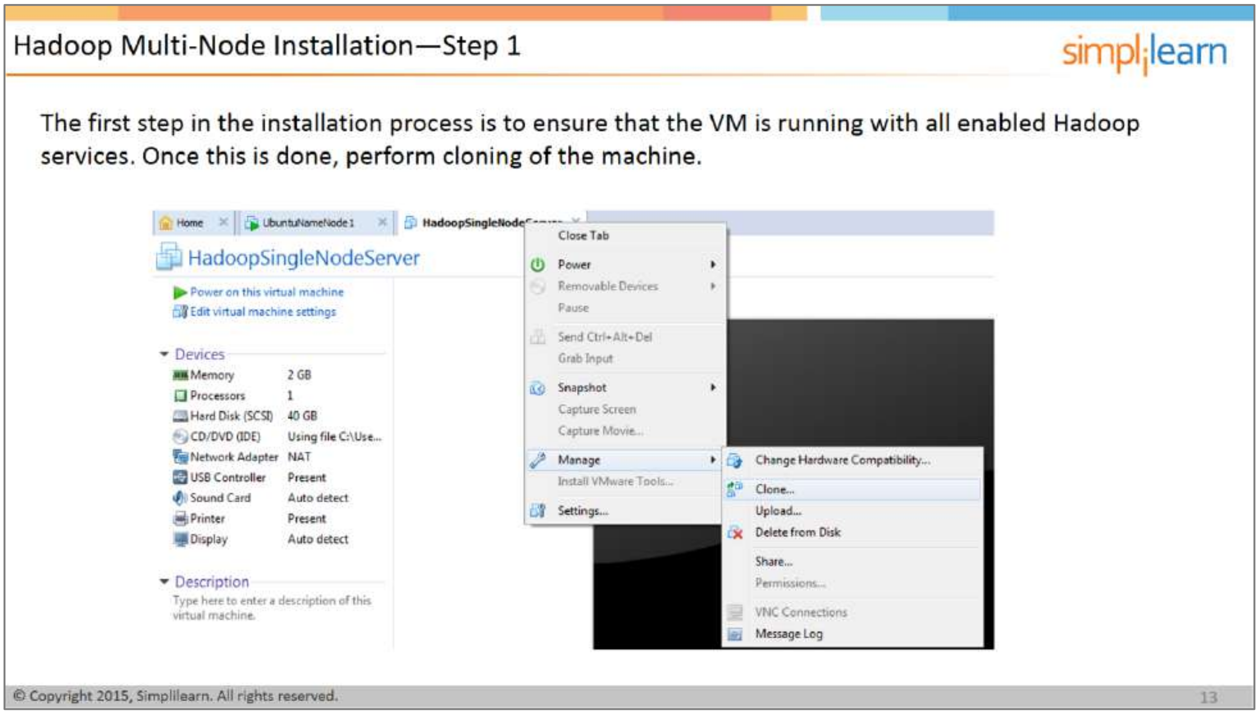
Ubuntu Server 14.04 machine with pseudo distributed environment high-speed Internet access to update and download the Hadoop files to the machine.



Steps for Hadoop Multi-Node Installation

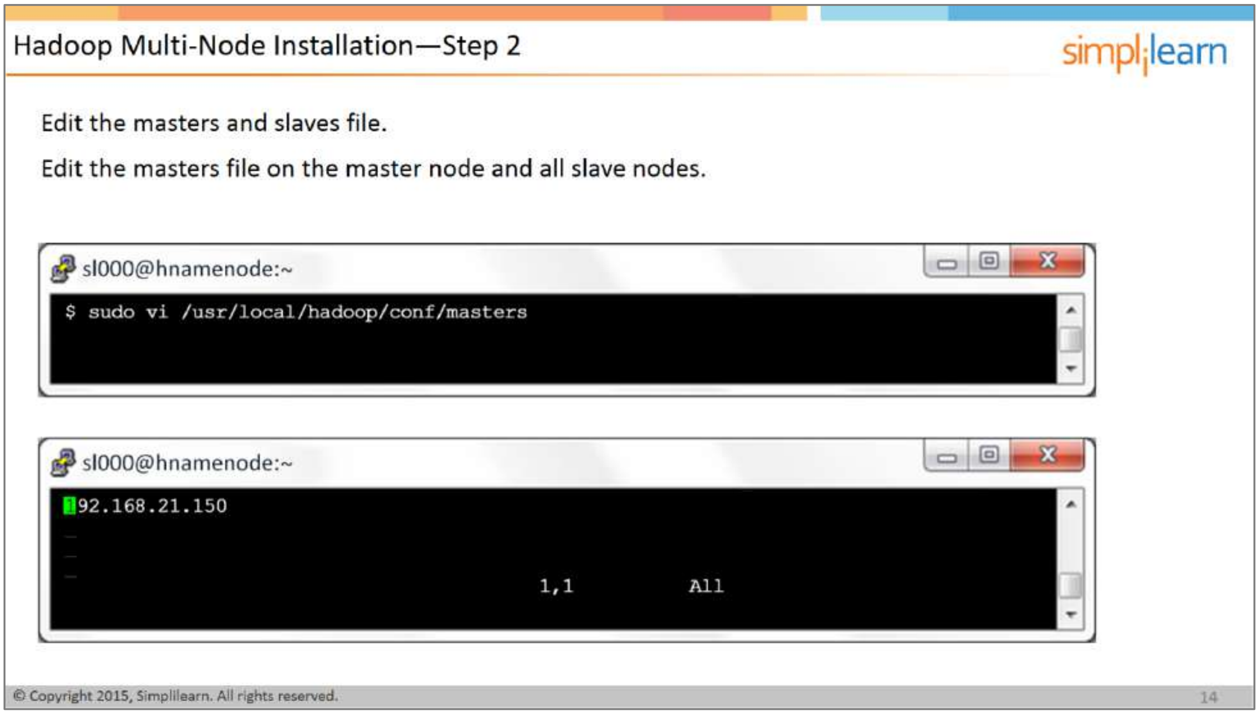
A multi-node environment is simulated by running multiple VMs on the same machine.

To install Hadoop 2.0 in distributed mode, you should follow the three steps given to successfully set up Hadoop in the Ubuntu Server 14.04 LTS.



Hadoop Multi-Node Installation—Step 1

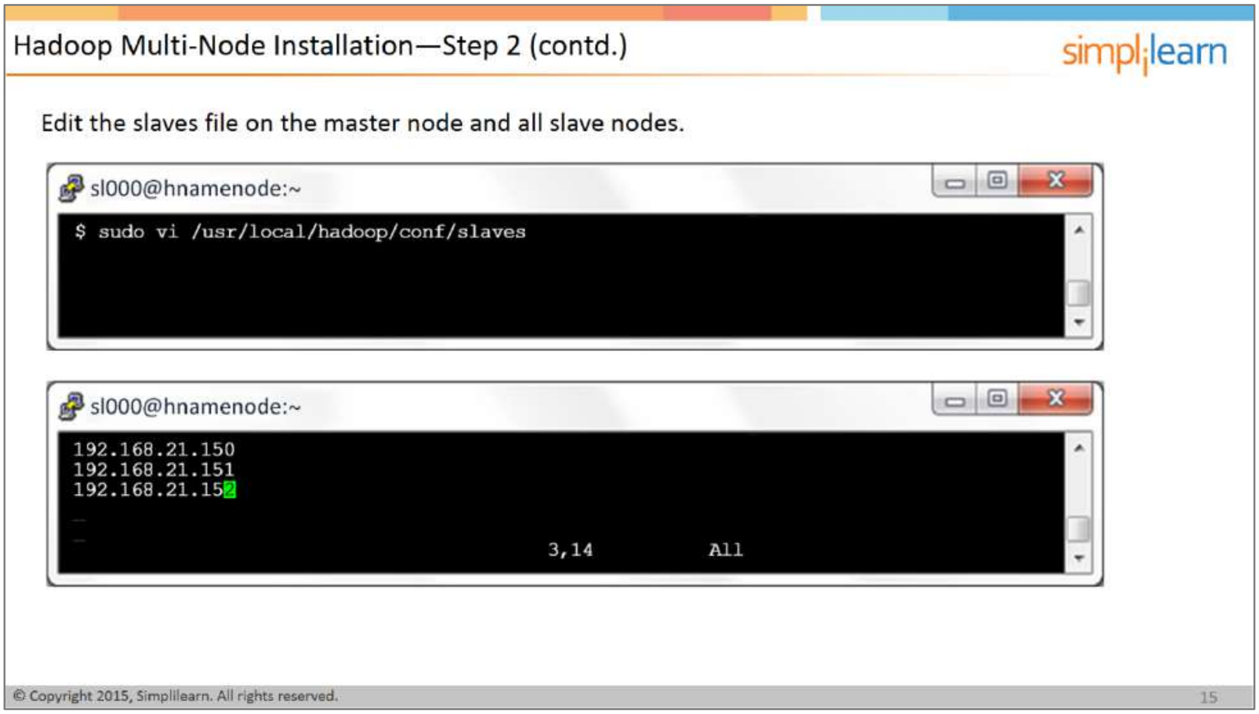
The first step is to ensure that the VM is running with all enabled Hadoop services. Once this is done, perform cloning of the machine. At least three machines are required.



Hadoop Multi-Node Installation—Step 2

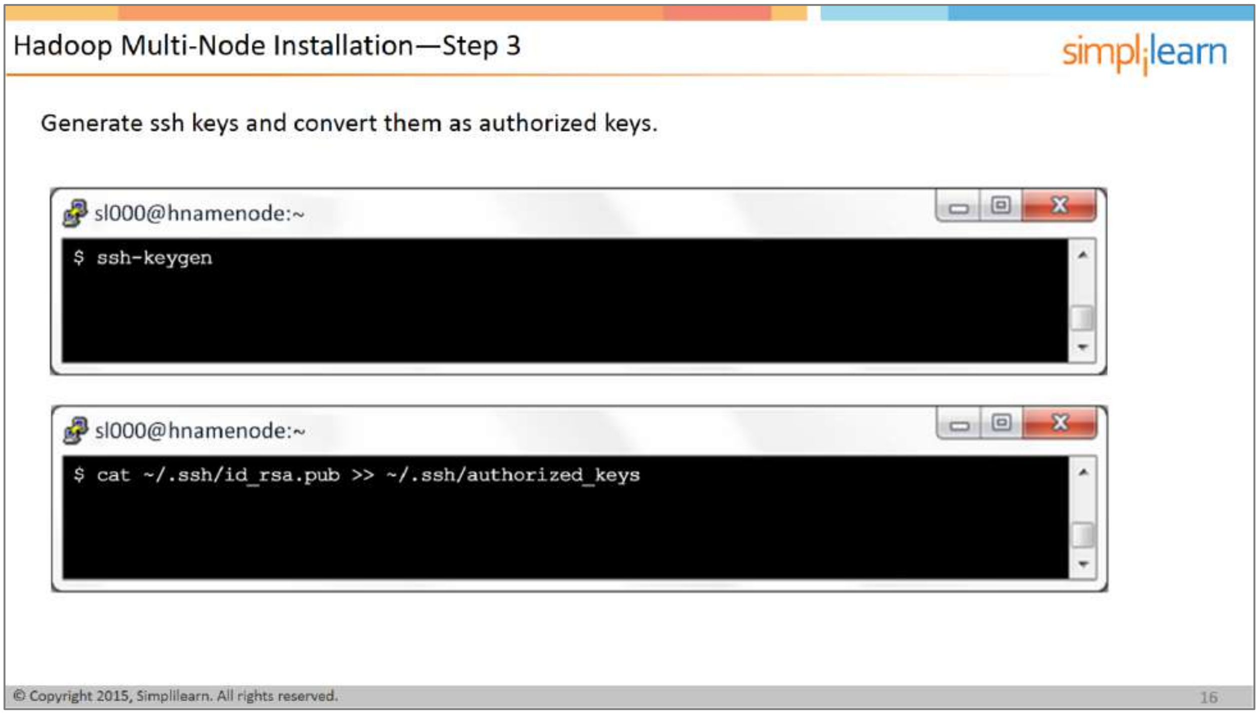
The second step is to edit the masters and slaves files.

The masters file is responsible for identifying the NameNode. Use the command displayed in the image to open the master file. Once the file is open, enter the IP address of the system that will act as the Secondary NameNode.



Hadoop Multi-Node Installation (contd.)

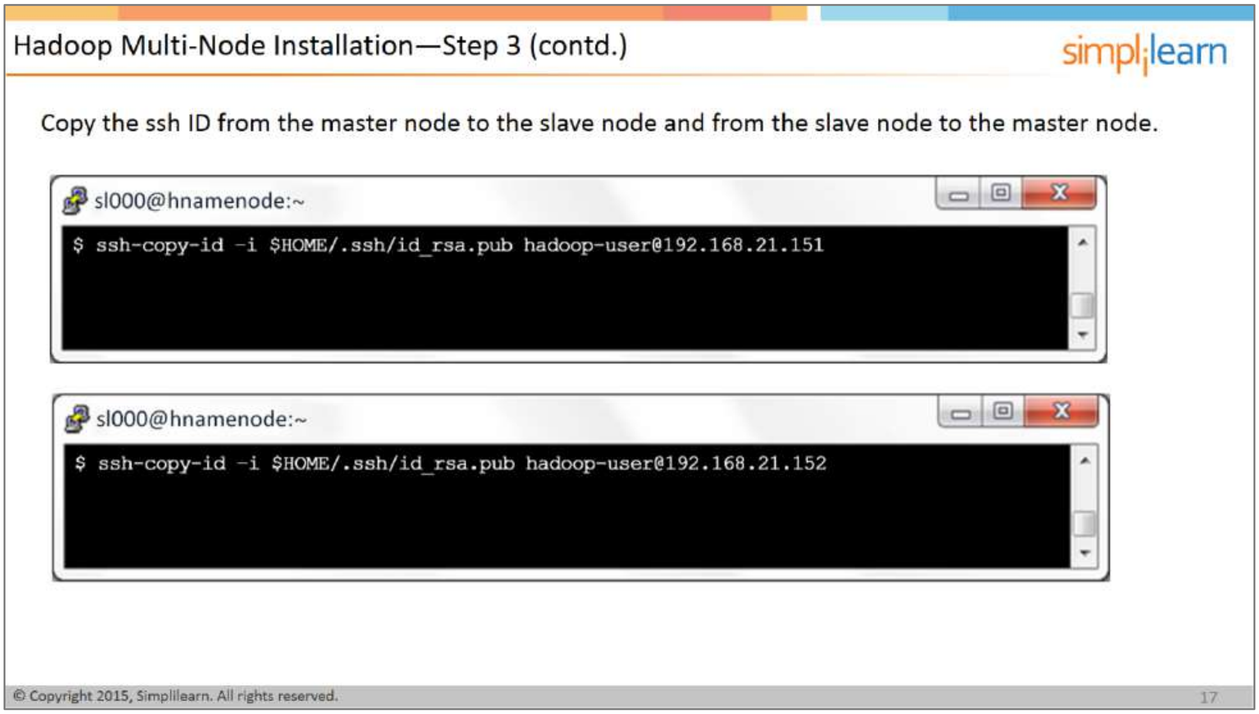
The slaves file is responsible for identifying the DataNode in the cluster. Use the command displayed in the first image to set up a slaves file. Once the file is open, enter the IP address of the system that will act as a DataNode and save the file. On each slave node, simply enter its own IP address or host name. There is no need to type the IP addresses or host names of other slaves.



Hadoop Multi-Node Installation—Step 3

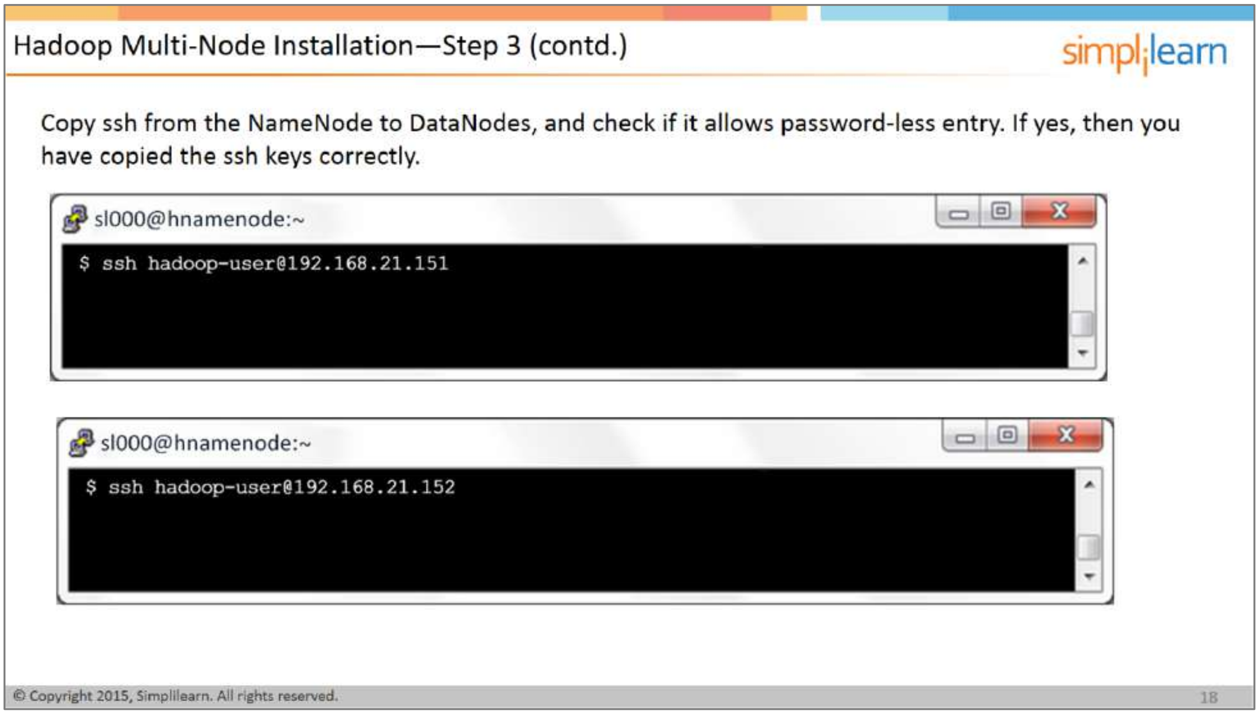
The third step is to generate ssh keys and copy them as authorized keys in all of the machines.

To set up the password-less login, create an ssh key initially. Use the command displayed in the first image to create an ssh key. Once you enter this command, you will be asked a number of questions. Press Enter after all questions to proceed. Once this is done, accept the generated RSA key as the authorized key. To do so, use the command displayed in the second image and press Enter.



Hadoop Multi-Node Installation—Step 3 (contd.)

Use the command displayed in the image to copy the authorized keys generated in the NameNode. Ensure that the command is executed for transferring the key in all of the nodes present in the cluster.

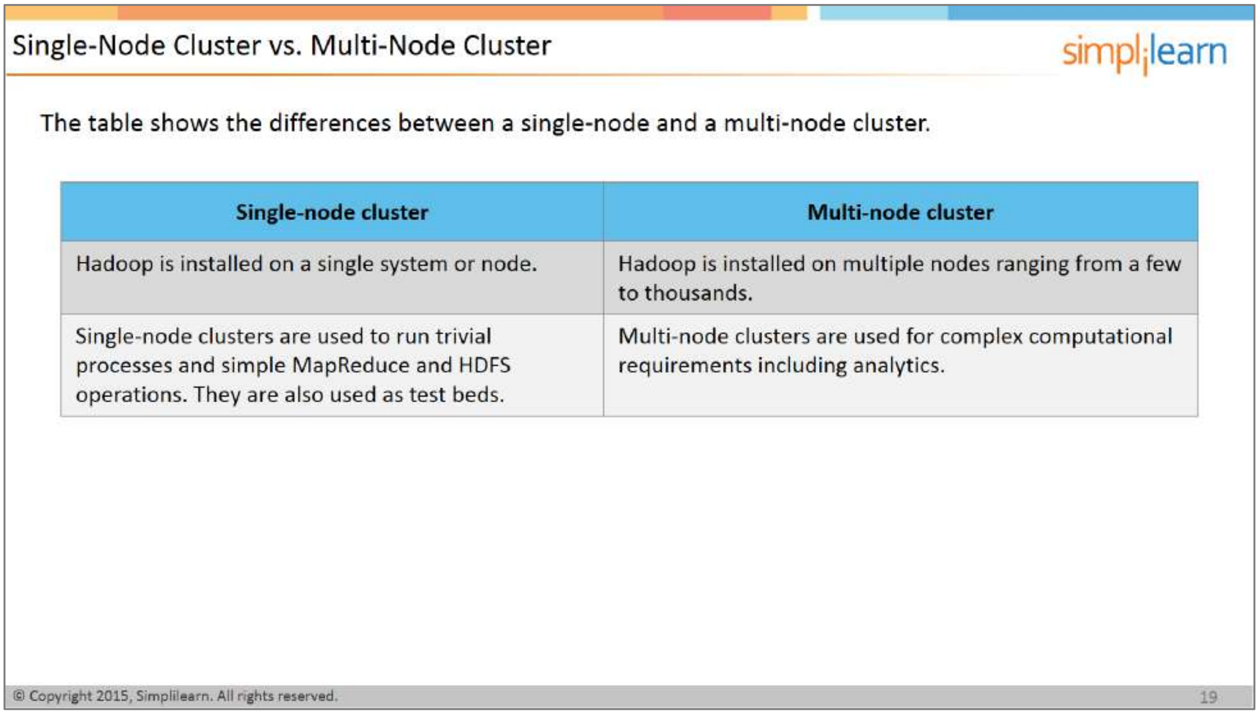


Hadoop Multi-Node Installation—Step 3 (contd.)

Copy ssh from the NameNode to DataNodes, and check if it allows password-less entry.

If yes, then you have copied the ssh keys correctly.

Perform the same steps from DataNodes to the NameNode by simply typing the command displayed in the second image.



Single-Node Cluster vs. Multi-Node Cluster

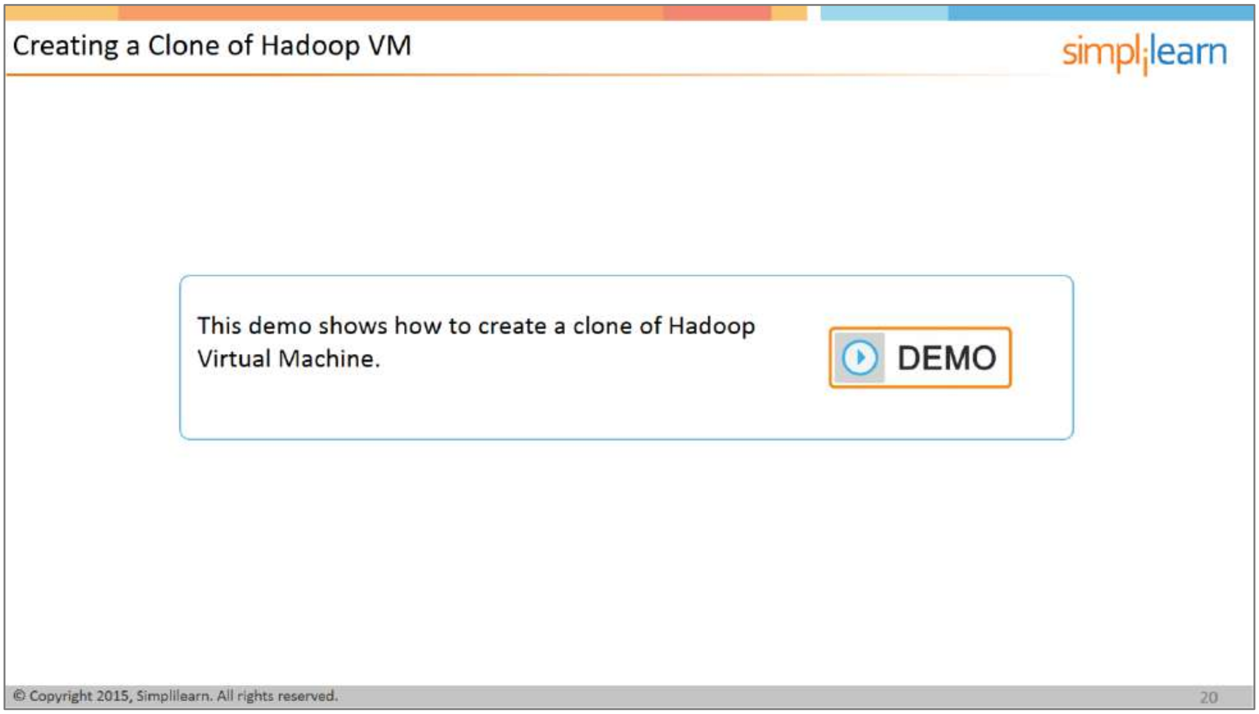
The table shows the differences between a single-node and a multi-node cluster.

Single-node cluster	Multi-node cluster
Hadoop is installed on a single system or node.	Hadoop is installed on multiple nodes ranging from a few to thousands.
Single-node clusters are used to run trivial processes and simple MapReduce and HDFS operations. They are also used as test beds.	Multi-node clusters are used for complex computational requirements including analytics.

19

Single-Node Cluster vs. Multi-Node Cluster

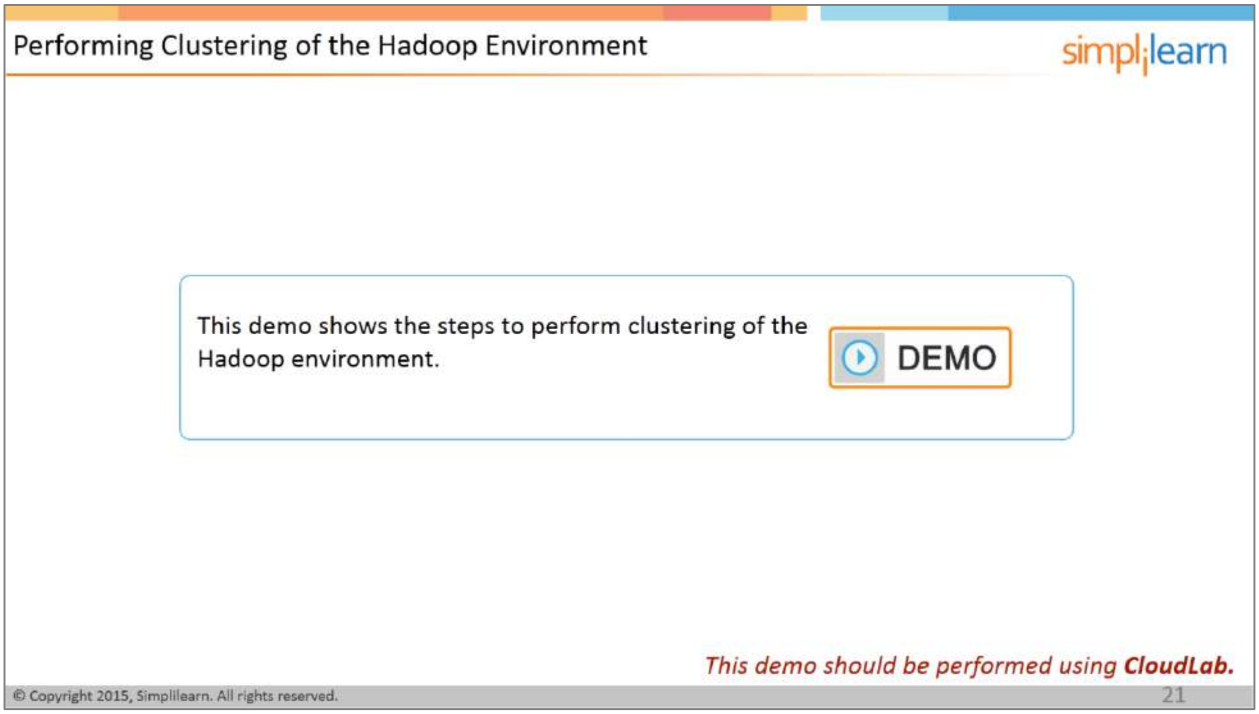
In a single node cluster, Hadoop is installed on a single system or node; in a multi-node cluster, however, Hadoop is installed on multiple nodes ranging from a few to thousands. Single node clusters are used to run trivial processes, simple MapReduce, and HDFS operations. They are also used as test beds. Multi-node clusters are used for complex computational requirements including analytics.



Creating a Clone of Hadoop VM

Let's summarize the steps of this demo as follows:

1. Click the VM menu.
2. Click the Clone menu item under the 'Manage' option.
3. The 'Clone Virtual Machine Wizard' window opens.
4. Click the Next button.
5. Click the Next button that follows.
6. Click create a full clone radio button and click **Next**.
7. Enter the clone virtual machine name and click **Finish**.
8. Click close from the **Clone Virtual Machine Wizard**.



Performing Clustering of the Hadoop Environment

Let's summarize the steps of this demo as follows:

This demo shows the steps to perform clustering of the Hadoop environment.

1. In this demo, we will create 2 node clusters with one NameNode and two DataNodes.
2. Open the master VM and go to the hadoop/conf directory.
3. List the content to observe various configuration files.
4. Open core-site.xml, hdfs-site.xml, and mapred-site.xml for editing in vi editor.
5. Enter dfs.replication value as 2 as we will be using two DataNodes in this demo.
6. Enter MapReduce job tracker IP address and port 9001.
7. Open the masters file and slaves file for editing. Enter IP address of the masters and slaves.
8. Switch to the slave terminal.
9. Check the IP address of the slave.
10. Go to the configuration directory.
11. Open core-site.xml for editing.
12. Ensure the IP Address of Namenode master is given and the Port is 9000.
13. Open hdfs-site.xml for editing.
14. Ensure dfs.replication value to 2 as you have two DataNodes.

15. Open mapred-site.xml and ensure the IP address of the task tracker is given and the Port is 9001.
16. Open the masters file for editing. Enter the IP addess of the master and save them.
17. Open the slave file. Enter the IP address of the slave files and save them.
18. Switch to master terminal. Generate ssh keys for password-less communication between nodes and press Enter.
19. Copy the SSH key from the master to the slave node.
20. SSH to slave node to confirm password-less entry.
21. Switch to slave terminal.
22. Generate SSH key and use the default options.
23. SSH key is generated.
24. Copy SSH Key from the slave file to the masters file.
25. Use the ssh-copy-id command with the master's IP address.
26. On the master node, start all daemons of Hadoop. Hadoop gets successfully started on master.
27. On the slave node, start all Hadoop daemons. Hadoop gets successfully started on slave.
28. On master and slave, list the contents of HDFS.
29. You will observe the same content on both master and slave.
30. Create a sample file.
31. Enter some content and save them.
32. Copy file to HDFS.
33. Use the Hadoop dfsadmin command.
34. Copy file to HDFS.
35. List the HDFS contents that are to be verified.
36. You will observe the new file.
37. Switch to slave and see the HDFS content listing.
38. When you see the file, you will observe that distributed file system is working properly on the 2 nodes.

Spot the Error

simplilearn

Michael has to create two node clusters with one NameNode and two DataNodes in his machine. He has edited all Hadoop files successfully. However, he faced a challenge while copying the SSH key from the master node to slave node. Can you assist Michael in identifying his error in the code?

Spot the incorrect code.

```
[ slave@simplilearn ]: vi core-site.xml
[ slave@simplilearn ]: vi hdfs-site.xml
[ slave@simplilearn ]: vi mapred-site.xml
[ slave@simplilearn ]: vi masters
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop-user/.ssh/id_rsa):
/home/hadoop-user/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop-user/.ssh/id_rsa.
Your public key has been saved in /home/hadoop-user/.ssh/id_rsa.pub.
The key fingerprint is:
00:ba:38:a2:a9:da:33:71:37:8d:56:e3:31:7c:ea:6f hadoop-user@hadoop-desk
[ master@simplilearn ]: cat ~/.ssh/id_rsa >> ~/.ssh/authorized_keys
[ master@simplilearn ]: ssh-cat-id -i $HOME/.ssh/id_rsa hadoop-user@192.168.197.134
```

 23

Spot the Error

simplilearn

The incorrect code is: `[ master@simplilearn ]: ssh-cat-id -i $HOME/.ssh/id rsa hadoop-user@192.168.197.134`

The correct code is: `ssh-copy-id -i $HOME/.ssh/id rsa hadoop-user@192.168.197.134`

 24

QUIZ 1

Which of the following commands will you use to archive?

a. gzip

b. tar

c. copy

d. cat

QUIZ 2

Which of the following commands helps to indicate a file or directory?

a. ls

b. ls -a

c. ls -l

d. man ls

QUIZ 3

Which of the following commands is used to start all Hadoop services?

a. start-dfs.sh

b. stop-dfs.sh

c. start-all.sh

d. stop-all.sh

 30

QUIZ 4

Which of the following commands is used to generate machine keys?

a. ssh

b. ssh-keygen

c. ssh-copy-id

d. jps

 32

QUIZ 5

Which file in Hadoop configuration is responsible for maintaining information about the NameNode?

a. core-site.xml

b. mapred-site.xml

c. hadoop-env.sh

d. masters file

QUIZ 6

Which of the following files in Hadoop configuration is responsible for maintaining information about the Application Master?

a. core-site.xml

b. mapred-site.xml

c. hadoop-env.sh

d. slaves file

simplilearn

QUIZ 7

Which of the following files is responsible for identifying the Secondary NameNode in a clustered environment?

a. masters

b. slaves

c. core-site.xml

d. mapred-site.xml

 38

simplilearn

QUIZ 8

Which of the following files is responsible for identifying the DataNode in a clustered environment?

a. masters

b. slaves

c. core-site.xml

d. mapred-site.xml

 40

simplilearn

QUIZ 9

Which of the following commands is responsible for copying the ssh ids from one machine to another?

a. ssh

b. ssh-keygen

c. ssh-copy-id

d. jps

 42

ANSWERS:

S.No.	Question	Answer & Explanation
1	What are the two major components of a Hadoop cluster?	b. The command tar stands for tape archive. This is not just for tape, but it is also for the disk file system.
2	Which of the following commands helps to indicate a file or directory?	c. The ls -l command helps to indicate a file or directory.
3	Which of the following commands is used to start all Hadoop services?	c. The start-all.sh command is used to start all Hadoop services.
4	Which of the following commands is used to generate machine keys?	b. The ssh-keygen is used to generate machine keys.
5	Which file in Hadoop configuration is responsible for maintaining information about the NameNode?	a. The core-site.xml file in Hadoop configuration is responsible for maintaining information about the NameNode.
6	Which of the following files in Hadoop configuration is responsible for maintaining information about the Application Master?	b. The mapred-site.xml file in Hadoop configuration is responsible for maintaining information about the Application Master.
7	Which of the following files is responsible for identifying the Secondary NameNode in a clustered environment?	a. The masters file is responsible for identifying the Secondary NameNode in a clustered environment.
8	Which of the following files is responsible for identifying the DataNode in a clustered environment?	b. The slaves file is responsible for identifying the DataNode in a clustered environment.
9	Which of the following commands is responsible for copying the ssh ids from one machine to another?	c. The command ssh-copy-id is responsible for copying the ssh ids from one machine to another.

Case Study

simplilearn

Scenario | Analysis | Solution

XY Invest provides investment advice to high net worth individuals and maintains stock market data of various exchanges. It handles stock market data critical for analysis and monitoring. This entails time-intensive storage of huge amounts of data, and vertical scaling is proving to be expensive. In its effort to find an alternate solution for storage, XY identifies that using Hadoop would reduce the company's cost and effort significantly.

 45

Case Study

simplilearn

Scenario | Analysis | Solution

The company does research on Hadoop and finds that it is a popular solution for Big Data storage and processing. Hadoop has two components, HDFS for storage and MapReduce for processing.

Advantages of using MapReduce:

1. It distributes processing so jobs can be completed really quickly.
2. It is highly fault tolerant, so even if some tasks fail, they will be automatically retried.
3. It moves processing to where the data is, reducing network traffic.
4. It is highly scalable, so you can add new machines to the cluster to handle increased load.
5. It can run on commodity-class hardware, significantly reducing costs.

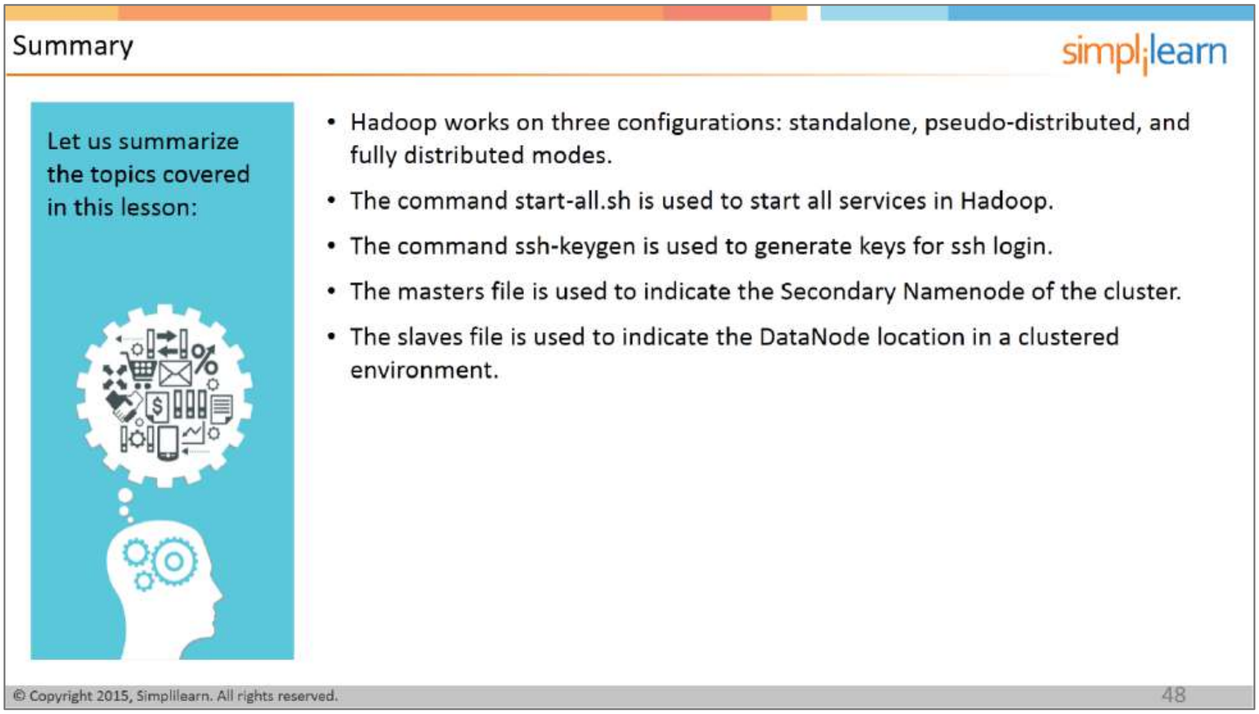
 46

Case Study

Scenario | Analysis | **Solution**

Perform the following steps to explore the Hadoop setup:

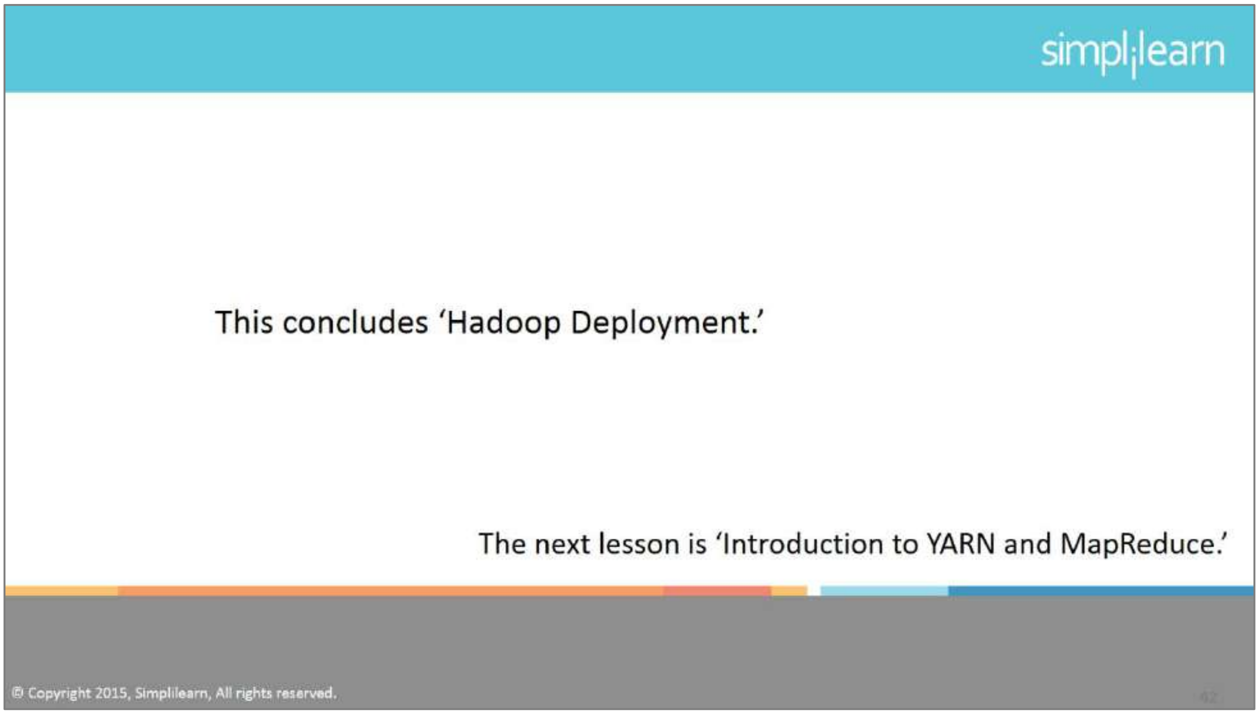
1. Get the log files from Hadoop programs into HDFS.
2. Run the *hadoop-example.jar* to process the word count.
3. Run the *hadoop-example.jar* to process the histogram.
4. Verify the number of map and reduce tasks.
5. Verify the distributed processing.
6. Verify the outputs.



Summarize

Let us summarize the topics covered in this lesson:

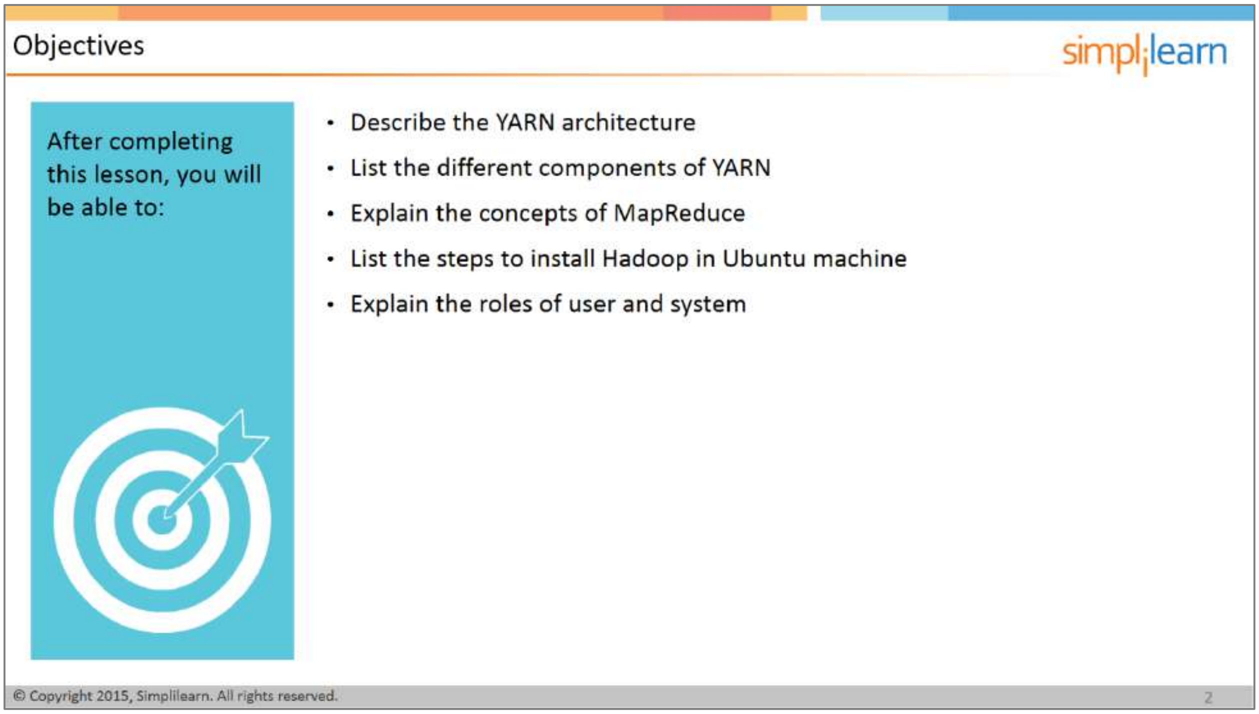
- Hadoop works on three configurations: standalone, pseudo-distributed, and fully distributed modes.
- The command start-all.sh is used to start all services in Hadoop.
- The command ssh-keygen is used to generate keys for ssh login.
- The masters file is used to indicate the Secondary NameNode of the cluster.
- The slaves file is used to indicate the DataNode location in a clustered environment.



Conclusion

This concludes 'Hadoop Deployment.' In the next lesson we will focus on 'Introduction to MapReduce.'

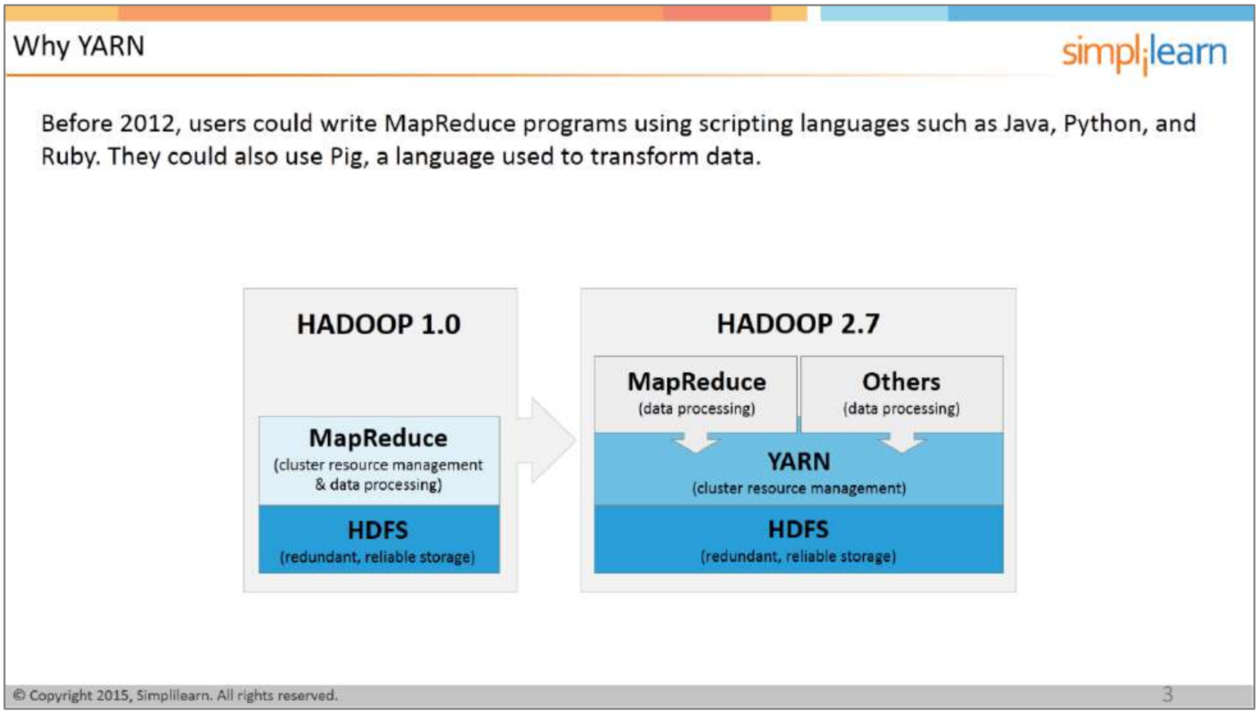
Lesson 4—Introduction to YARN and MapReduce



Introduction to YARN and MapReduce—Objectives

After completing this lesson, you will be able to:

- Describe the YARN architecture
- List the different components of YARN
- Explain the concepts of MapReduce
- List the steps to install Hadoop in Ubuntu machine
- Explain the roles of user and system



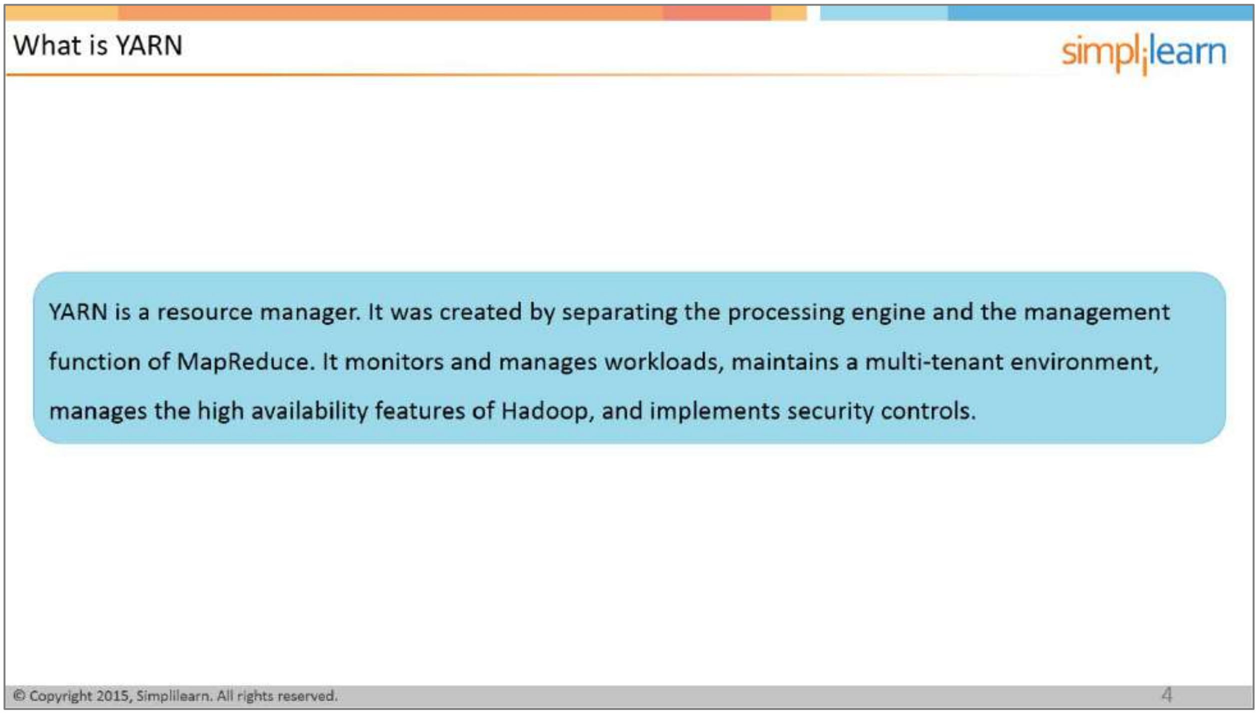
Why YARN

Before 2012, users could write MapReduce programs using scripting languages such as Java, Python, and Ruby. They could also use Pig, a language used to transform data. No matter what language was used, its implementation was dependent on the MapReduce processing model.

Hadoop version 2.0 was released in May 2012 with the introduction of 'Yet Another Resource Navigator,' popularly known as YARN.

YARN is called the Operating System of Hadoop. Significantly, we are not limited to work with the often latent MapReduce framework anymore, as it supports multiple processing models in addition to MapReduce, such as Spark.

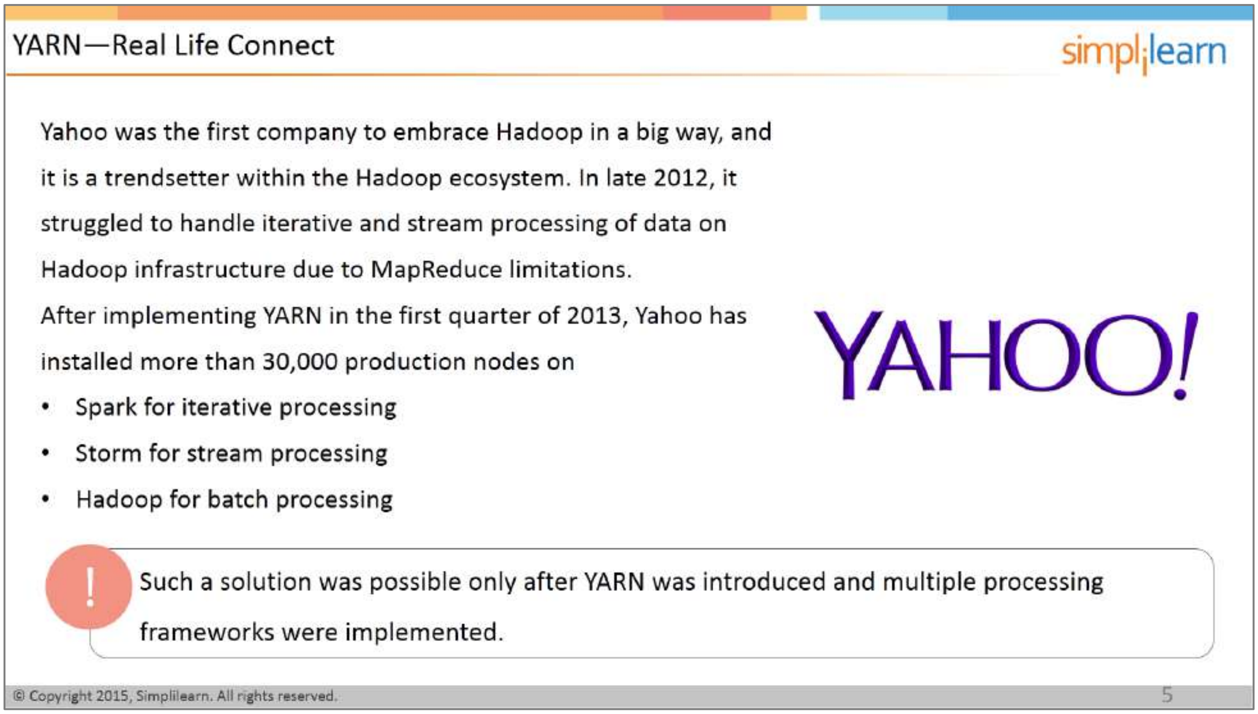
Other USPs of YARN are significant performance improvement and a flexible execution engine.



What is YARN

YARN is a resource manager. It was created by separating the processing engine and the management function of MapReduce. It monitors and manages workloads, maintains a multi-tenant environment, manages the high availability features of Hadoop, and implements security controls.

In a nutshell, Hadoop YARN is an attempt to take Apache Hadoop beyond MapReduce for data-processing.



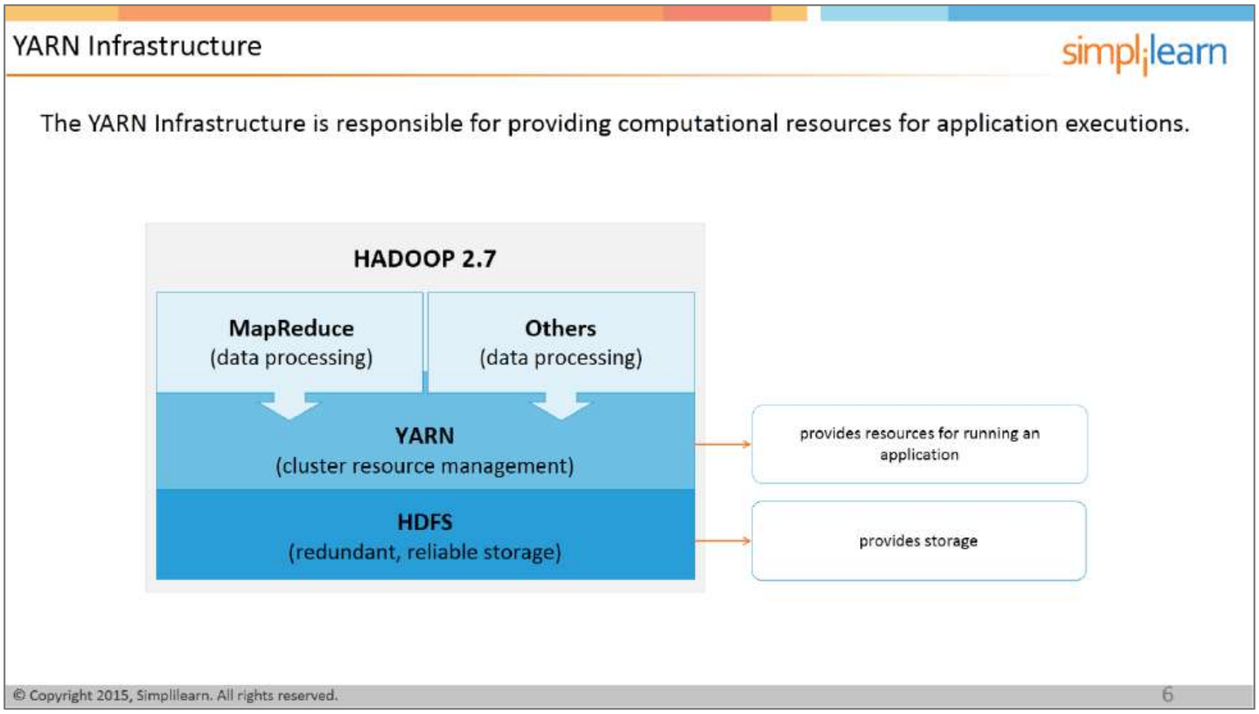
YARN—Real Life Connect

Before 2012, users could write MapReduce programs using scripting languages such as Java, Python, and Ruby. They could also use Pig, a language used to transform data. No matter what language was used, its implementation was dependent on the MapReduce processing model.

Hadoop version 2.0 was released in May 2012 with the introduction of 'Yet Another Resource Navigator,' popularly known as YARN.

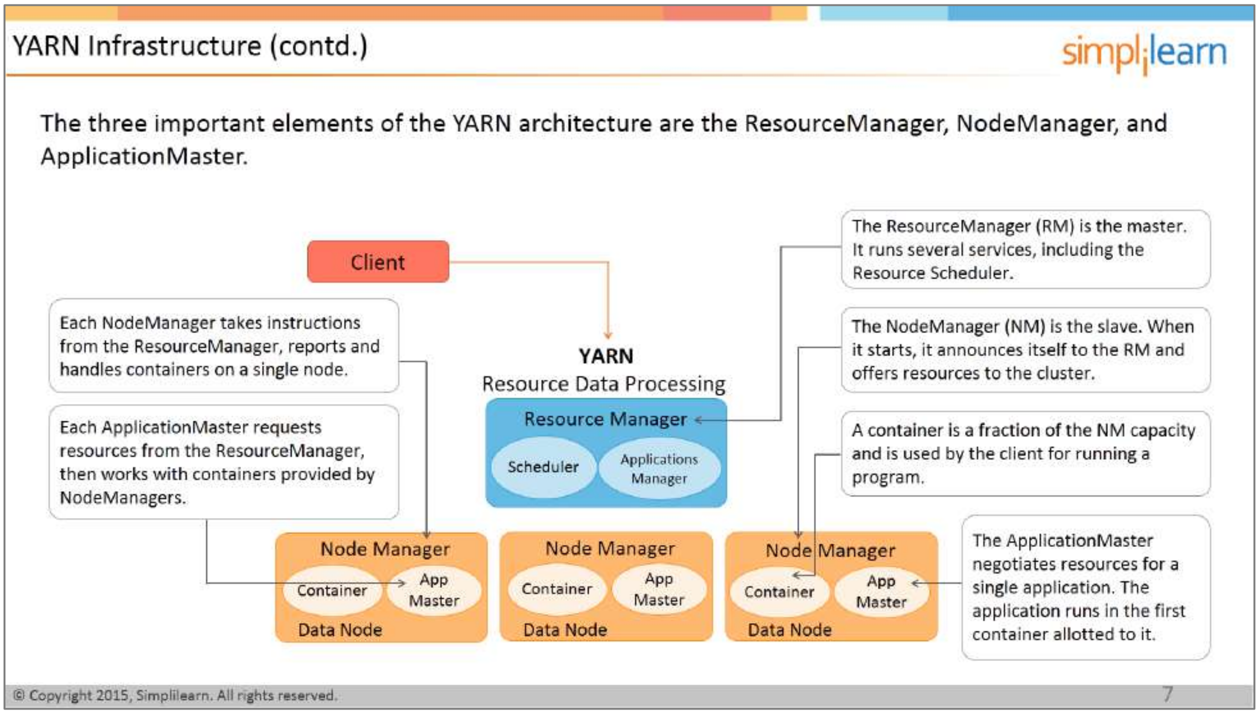
YARN is called the Operating System of Hadoop. Significantly, we are not limited to work with the often latent MapReduce framework anymore, as it supports multiple processing models in addition to MapReduce, such as Spark.

Other USPs of YARN are significant performance improvement and a flexible execution engine.



YARN Infrastructure

The YARN Infrastructure is responsible for providing the computational resources, such as, CPUs or memory needed for application executions. The YARN infrastructure and the HDFS federation are completely decoupled and independent: the former provides resources for running an application while the latter provides storage. The MapReduce framework is only one of the many possible frameworks which runs on top of YARN, although currently, it is the only one implemented. The fundamental idea of MRv2 is to split up the two major functionalities; resource management and job scheduling and monitoring, into separate daemons. There is a global ResourceManager and per-application ApplicationMaster.



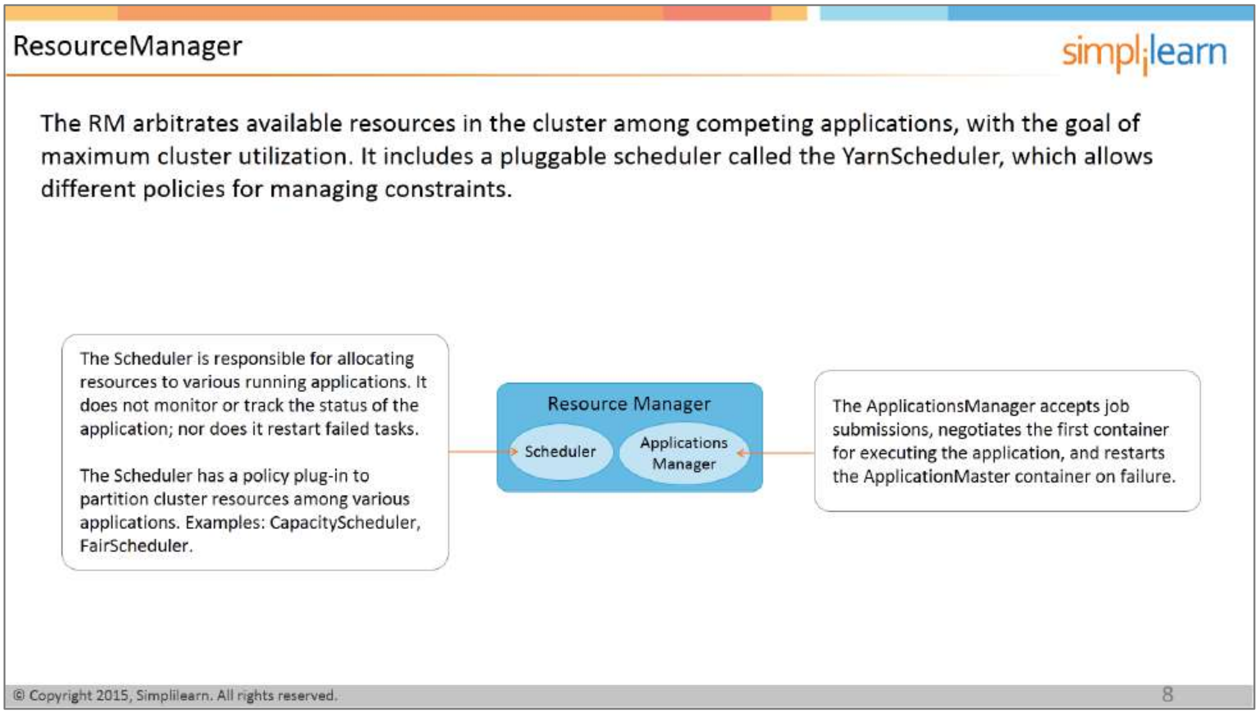
YARN Infrastructure (contd.)

The three important elements of the YARN architecture are:

- The ResourceManager, or RM, usually numbered one per cluster, is the master and knows where the slaves are located, referred to as Rack Awareness, and how many resources they have. The RM runs several services, the most important of which is the Resource Scheduler that decides how to assign the resources.
- The NodeManager, of which there can be many in one cluster, is the slave of the infrastructure. When it starts, it announces itself to the RM and periodically sends a heartbeat to the RM. Each NodeManager offers resources to the cluster, the resource capacity being the amount of memory and the number of vcores. At run-time, the Resource Scheduler decides how to use this capacity. A container is a fraction of the NodeManager capacity and it is used by the client for running a program.

Each NodeManager takes instructions from the ResourceManager, and reports and handles containers on a single node.

- The ApplicationMaster is a framework-specific process that negotiates resources for a single application, that is, a single job or a directed acyclic graph of jobs, which runs in the first container allocated for the purpose. Each ApplicationMaster requests resources from the ResourceManager, then works with containers provided by NodeManagers.



ResourceManager

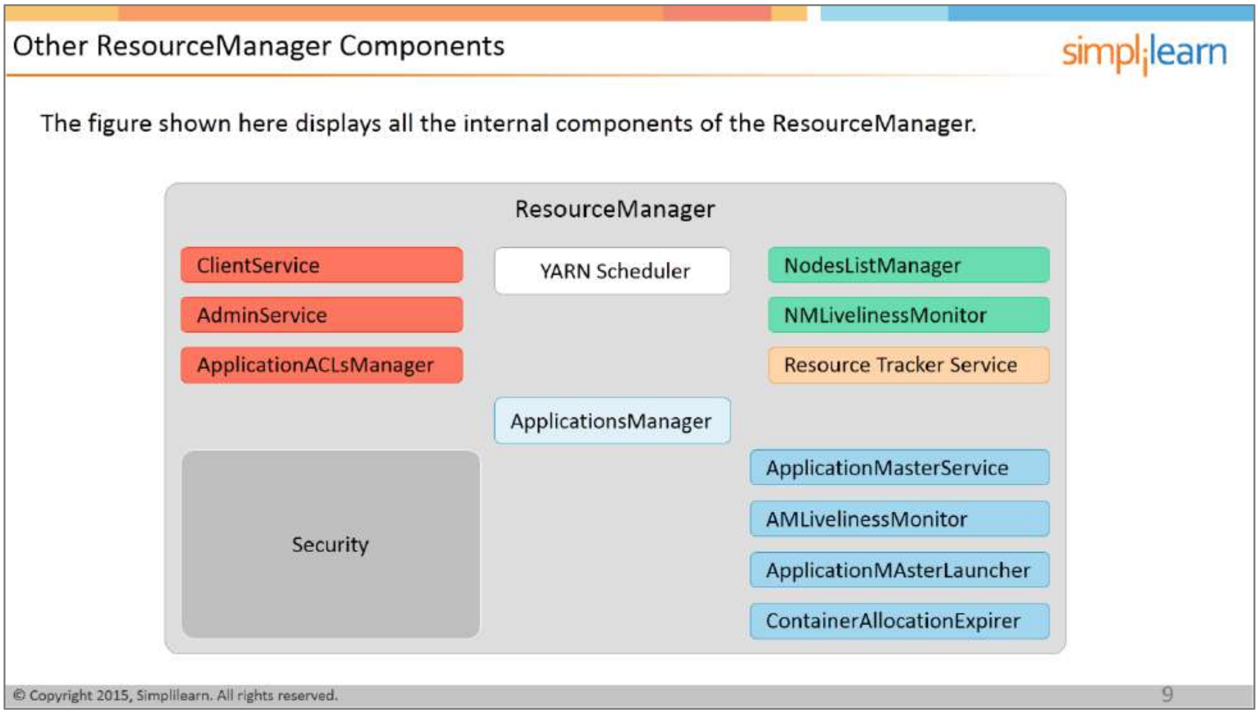
The RM arbitrates available resources in the cluster among competing applications, with the goal of maximum cluster utilization. It includes a pluggable scheduler called the YarnScheduler, which allows different policies for managing constraints such as capacity, fairness, and Service Level Agreements.

The ResourceManager has two main components: Scheduler and ApplicationsManager.

The Scheduler is responsible for allocating resources to various running applications depending on common constraints of capacities, queues, and so on. The Scheduler does not monitor or track the status of the application. Also, it does not ensure the restarting of tasks that failed either due to application failure or hardware failures. The Scheduler performs its function based on the resource requirements of the applications; it does so based on the abstract notion of a resource container which incorporates elements such as memory, CPU, disk, and network.

The Scheduler has a policy plug-in, which is responsible for partitioning the cluster resources among various queues and applications. The current MapReduce schedulers such as the CapacityScheduler and the FairScheduler are some examples of the plug-in.

The CapacityScheduler supports hierarchical queues to enable more predictable sharing of cluster resources.At the core of the ResourceManager is an interface called the ApplicationsManager, which maintains a list of applications that have been submitted, are currently running, or completed. The ApplicationsManager is responsible for accepting job-submissions, negotiating the first container for executing the application specific ApplicationMaster, and restarting the ApplicationMaster container on failure.



Other ResourceManager Components

The figure shown here displays all the internal components of the ResourceManager.

The ResourceManager communicates with application clients through an interface called the ClientService. A client can submit or terminate an application and gain information about the scheduling queue or cluster statistics through the ClientService.

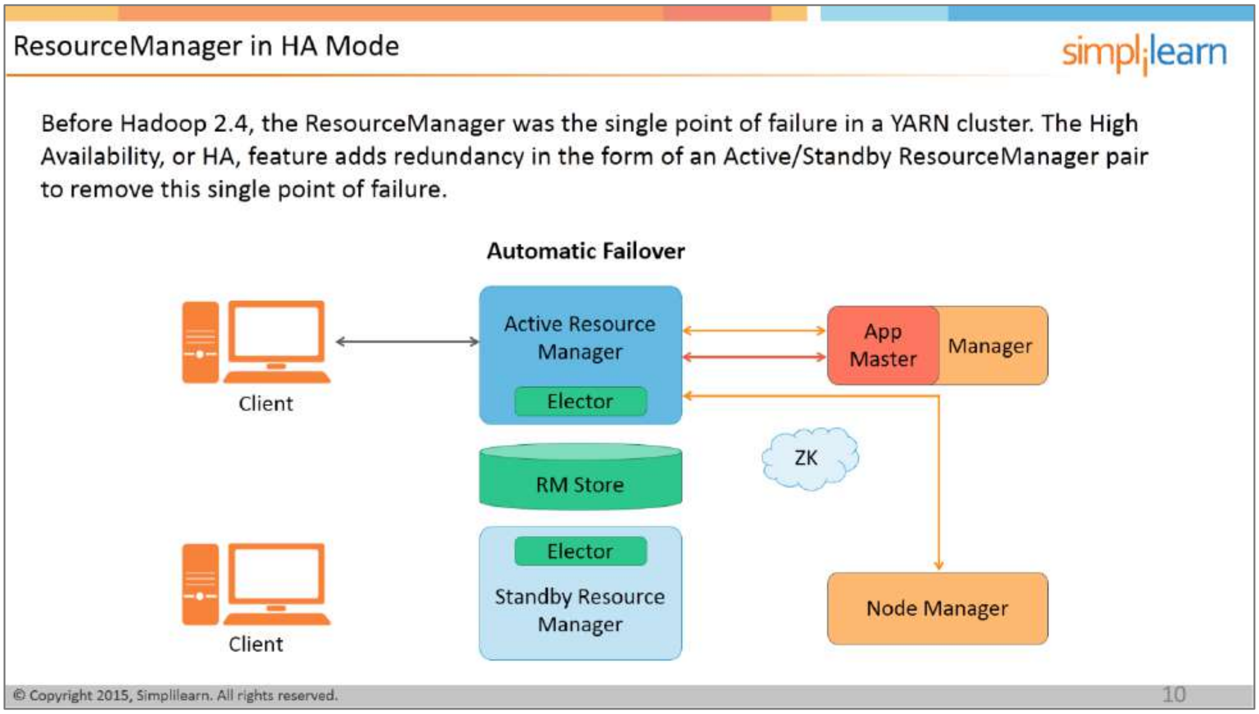
Administrative requests are served by a separate interface called the AdminService, through which operators can get updated information about cluster operation.

In parallel, the ResourceTrackerService receives node heartbeats from the NodeManager to track new or decommissioned nodes.

The NMLivelinessMonitor and NodesListManager keep an updated status of which nodes are healthy so that the scheduler and the ResourceTrackerService can allocate work appropriately.

The ApplicationMasterService manages ApplicationMasters on all nodes, keeping the scheduler informed.

The AMLivelinessMonitor keeps a list of ApplicationMasters and their last heartbeat times, to let the ResourceManager know what applications are healthy on the cluster. Any ApplicationMaster that does not heartbeat within a certain interval is marked as dead and re-scheduled to run on a new container.



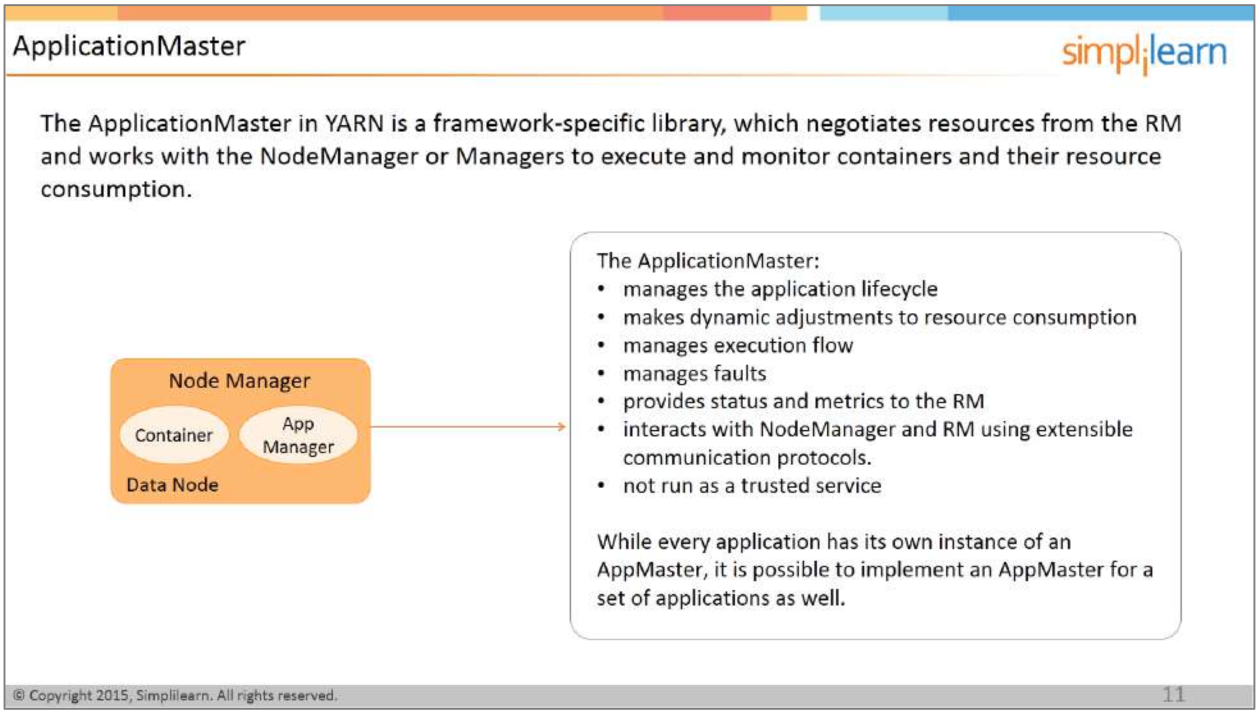
ResourceManager in HA Mode

Before Hadoop 2.4, the ResourceManager was the single point of failure in a YARN cluster. The High Availability, or HA, feature adds redundancy in the form of an Active/Standby ResourceManager pair to remove this single point of failure.

ResourceManager HA is realized through the Active/Standby architecture: at any point of time, one of the RMs is active, and one or more RMs are in Standby mode waiting to take over should anything happen to the Active. The trigger to transition-to-active comes from either the admin, through CLI, or where automatic failover is enabled, through the integrated failover-controller.

Let’s take a closer look at Automatic Failover.

The RMs have an option to embed the Zookeeper-based ActiveStandbyElector to decide which RM should be the Active. When the Active goes down or becomes unresponsive, another RM is automatically elected to be the Active. Note that, there is no need to run a separate ZKFC daemon, like in HDFS, because the ActiveStandbyElector embedded in RMs acts as a failure detector and a leader elector.



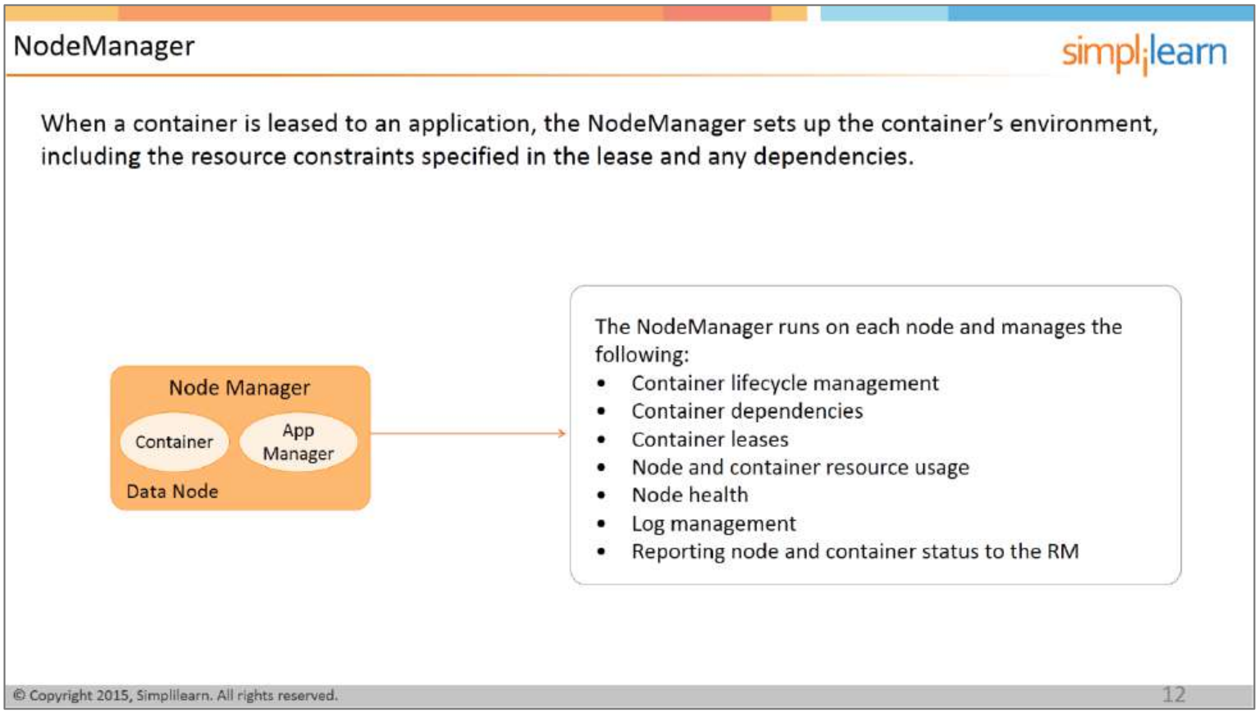
ApplicationMaster

The ApplicationMaster in YARN is a framework-specific library, which negotiates resources from the RM and works with the NodeManager or Managers to execute and monitor containers and their resource consumption.

While an application is running, the ApplicationMaster manages the application lifecycle, dynamic adjustments to resource consumption, execution flow, faults, and provides status and metrics.

The ApplicationMaster is architected to support a specific framework, and can be written in any language since its communication with the NodeManagers and the ResourceManager is done using extensible communication protocols. The ApplicationMaster can be customized to extend the framework or run any other code. Because of this the ApplicationMaster is not considered trustworthy, and is not run as a trusted service.

In reality, every application has its own instance of an ApplicationMaster. However, it’s feasible to implement an ApplicationMaster to manage a set of applications, for example, an ApplicationMaster for Pig or Hive to manage a set of MapReduce jobs.



NodeManager

When a container is leased to an application, the NodeManager sets up the container's environment, including the resource constraints specified in the lease and any dependencies, such as data or executable files.

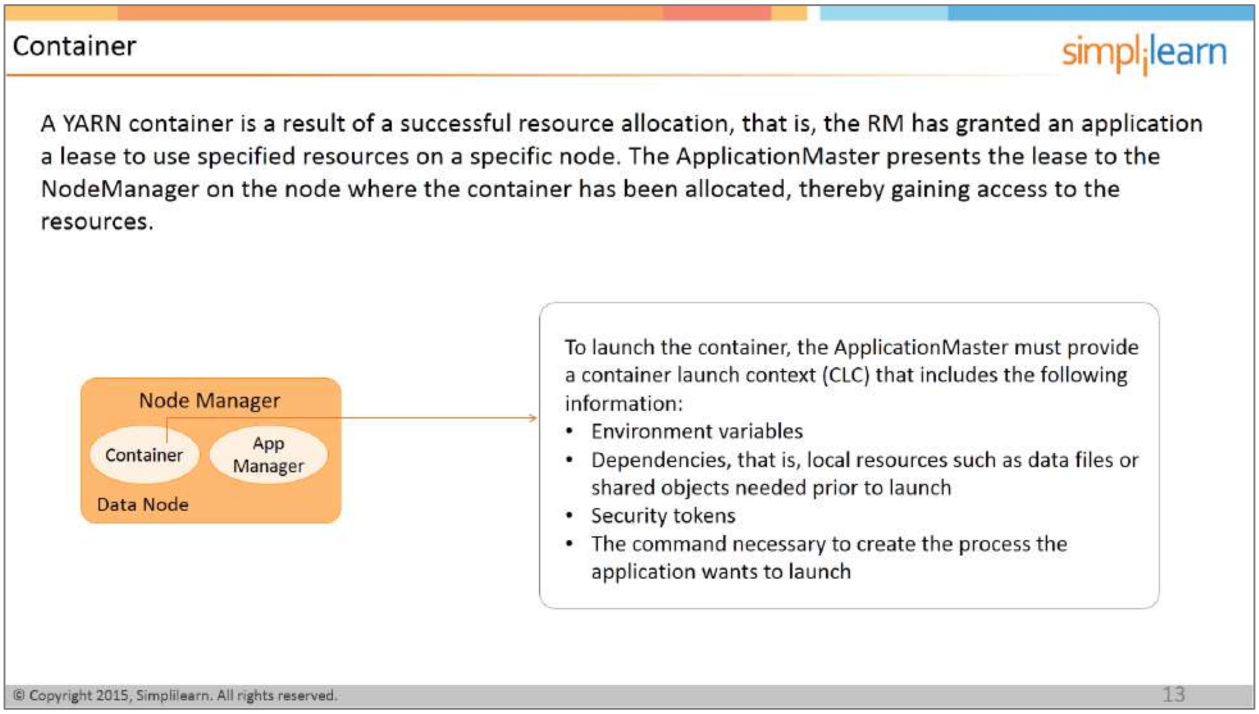
The NodeManager monitors the health of the node, reporting to the ResourceManager when a hardware or software issue occurs so that the scheduler can divert resource allocations to healthy nodes until the issue is resolved.

The NodeManager also offers a number of services to containers running on the node such as a log aggregation service.

The NodeManager runs on each node and manages the following:

- Container lifecycle management
- Container dependencies
- Container leases
- Node and container resource usage
- Node health
- Log management

Reporting node and container status to the ResourceManager.



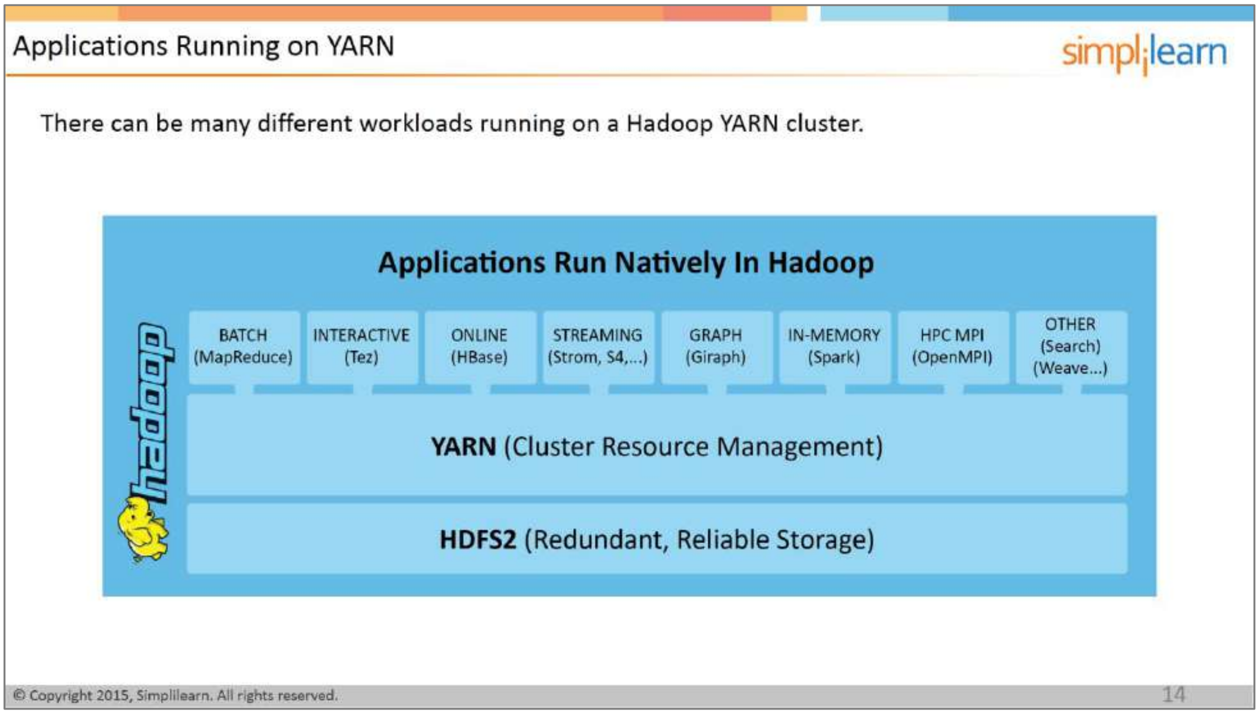
Container

A YARN container is a result of a successful resource allocation, meaning that the ResourceManager has granted an application a lease to use a specific set of resources in certain amounts on a specific node. The ApplicationMaster presents the lease to the NodeManager on the node where the container has been allocated, thereby gaining access to the resources.

To launch the container, the ApplicationMaster must provide a container launch context or CLC that includes the following information:

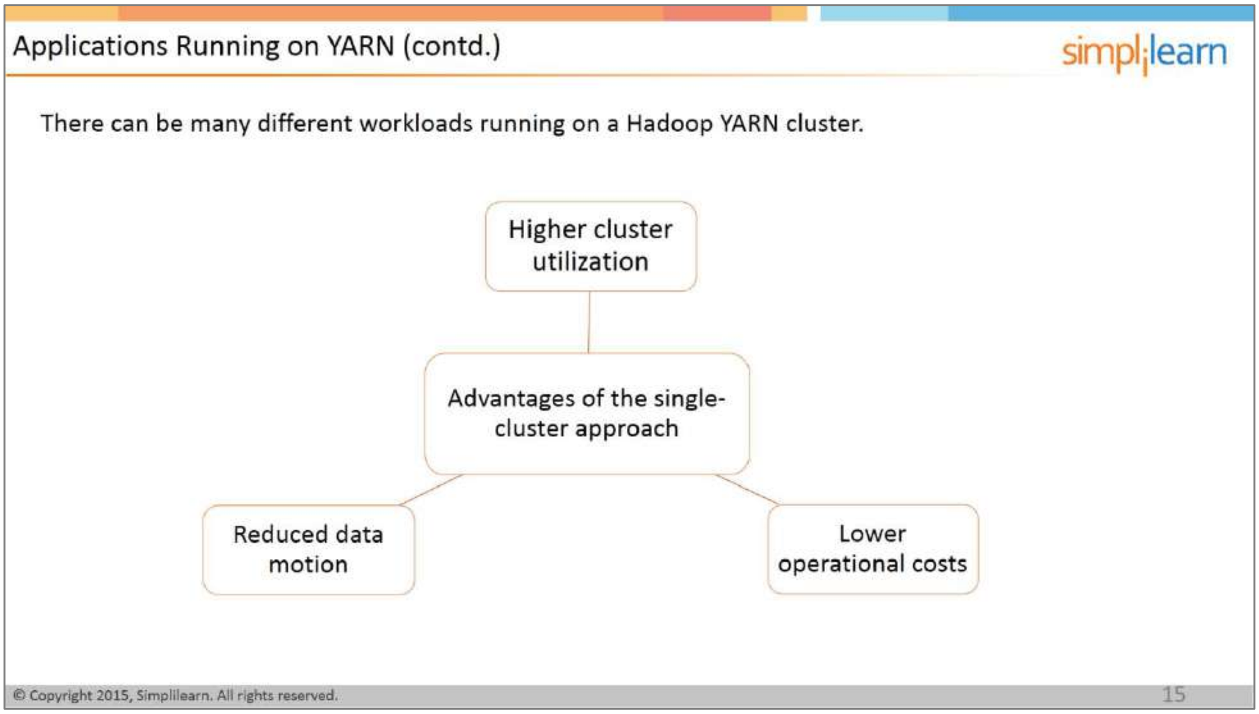
- Environment variables
- Dependencies, that is, local resources such as data files or shared objects needed prior to launch
- Security tokens
- The command necessary to create the process the application wants to launch

The CLC makes it possible for the ApplicationMaster to use containers to run a variety of different kinds of work, from simple shell scripts to applications to virtual machines.



Applications Running on YARN

Owing to YARN's generic approach, a Hadoop YARN cluster running many different workloads is now a possibility. This means a single Hadoop cluster in your data center can run MapReduce, Giraph, Storm, Spark, Tez/Impala, MPI, and more.

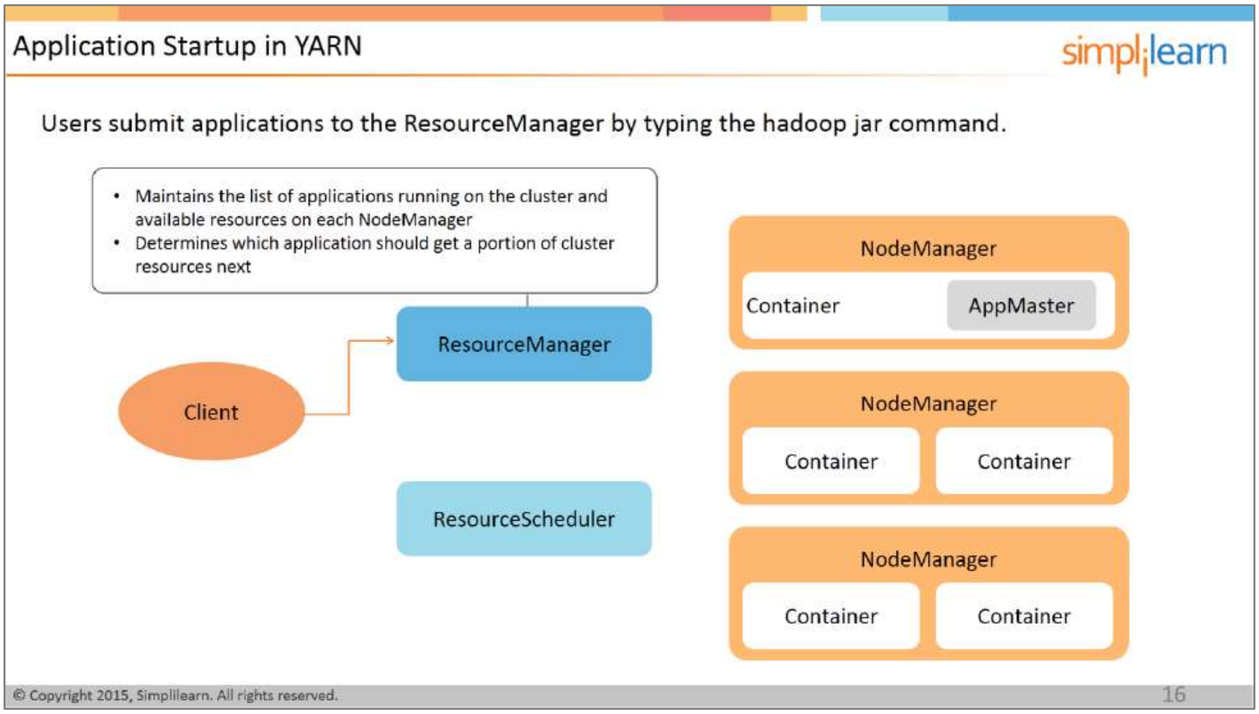


Applications Running on YARN (contd.)

The single-cluster approach obviously provides a number of advantages, including:

- Higher cluster utilization, whereby resources not used by one framework could be consumed by another
- Lower operational costs, because only one "do-it-all" cluster needs to be managed and tuned

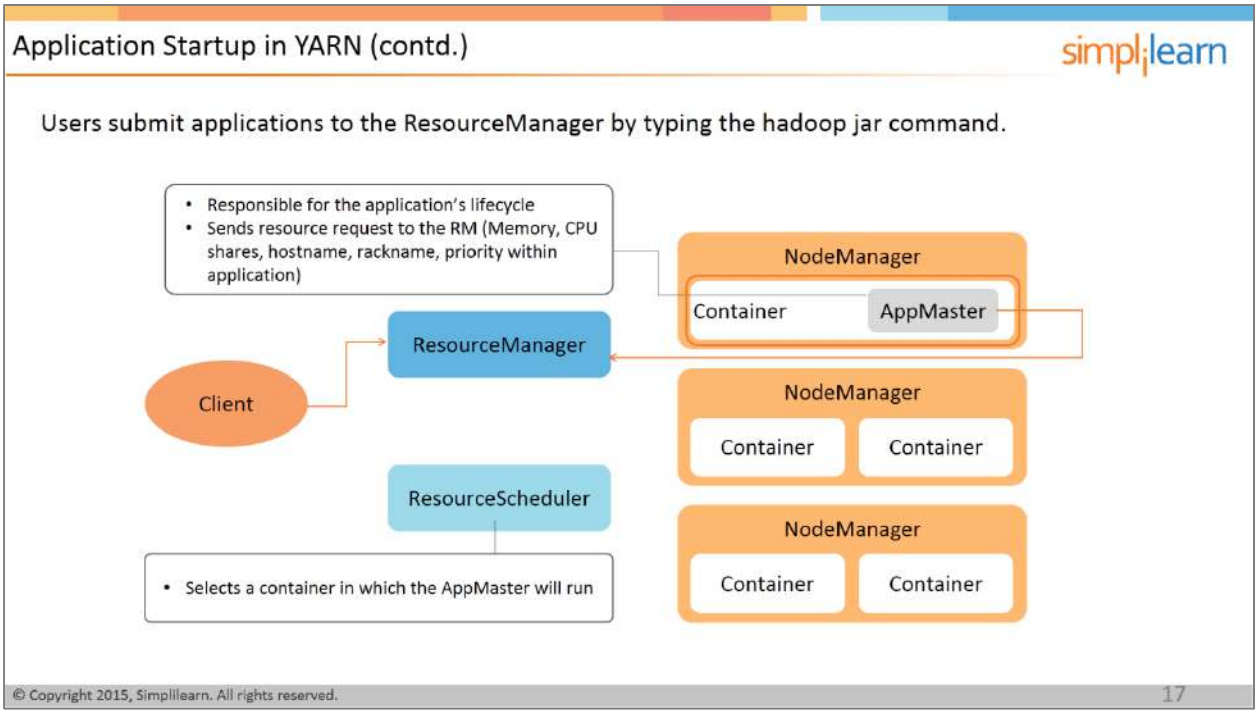
Reduced data motion, as there's no need to move data between Hadoop YARN and systems running on different clusters of machines



Application Startup in YARN

Suppose users submit applications to the ResourceManager by typing the hadoop jar command.

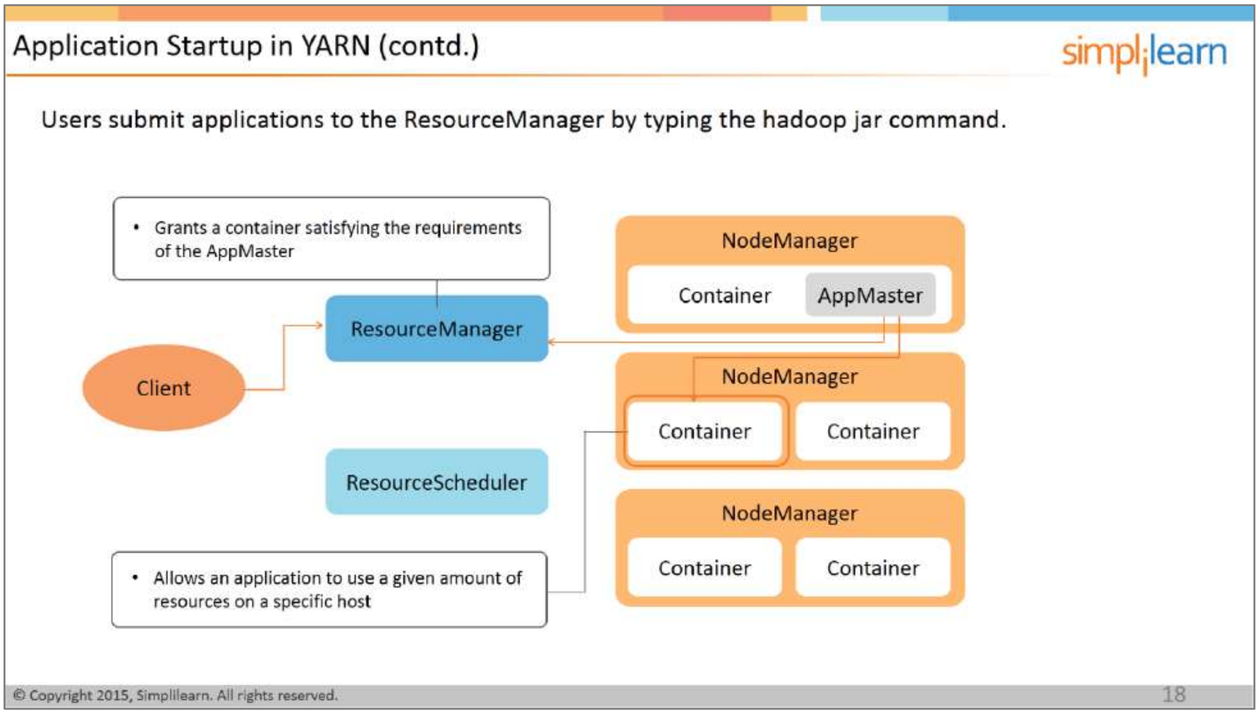
The ResourceManager maintains the list of applications running on the cluster and available resources on each live NodeManager. The ResourceManager determines which application should get a portion of cluster resources next. The decision is subject to many constraints, such as, queue capacity, ACLs, and fairness.



Application Startup in YARN (contd.)

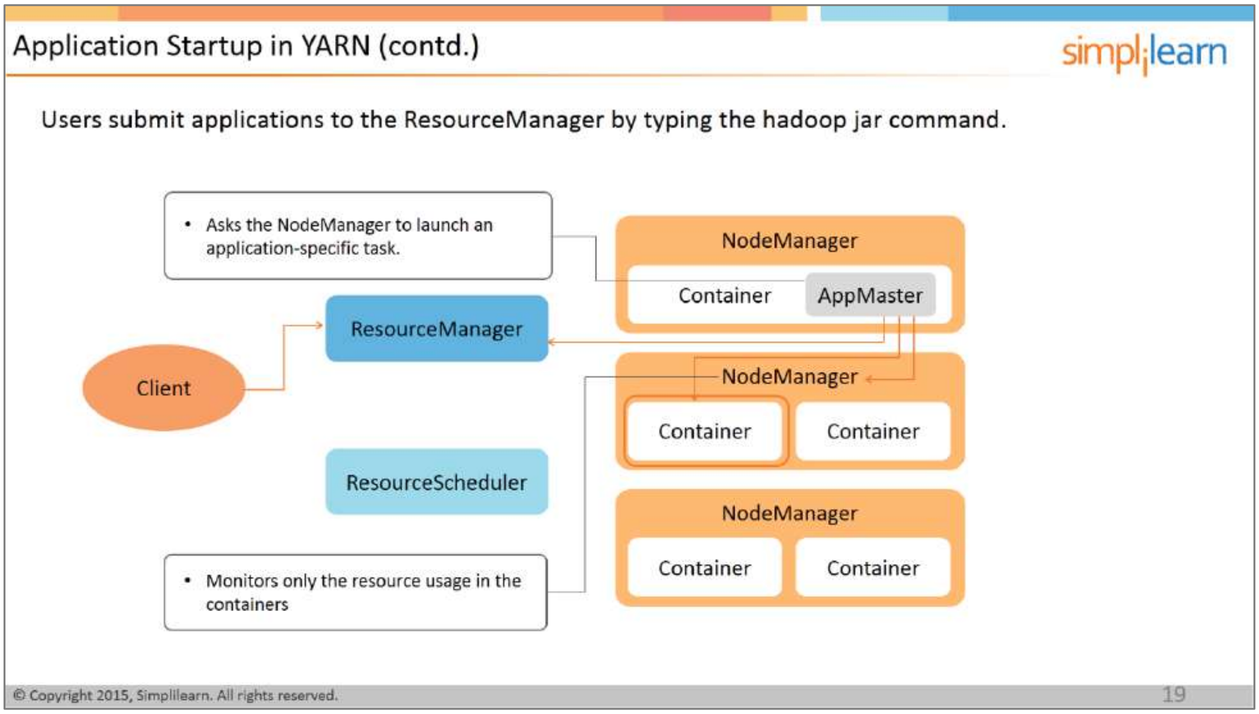
When the ResourceManager accepts a new application submission, one of the first decisions the Scheduler makes is selecting a container in which the ApplicationMaster will run. After the ApplicationMaster is started, it is responsible for the entire life cycle of this application. First, it sends resource requests to the ResourceManager to ask for containers to run the application's tasks. A resource request is simply a request for a number of containers that satisfy resource requirements, such as:

- Amount of resources, expressed as megabytes of memory and CPU shares
- Preferred location, specified by hostname, rackname, or star to indicate no preference
- Priority within this application, and not across multiple applications



Application Startup in YARN (contd.)

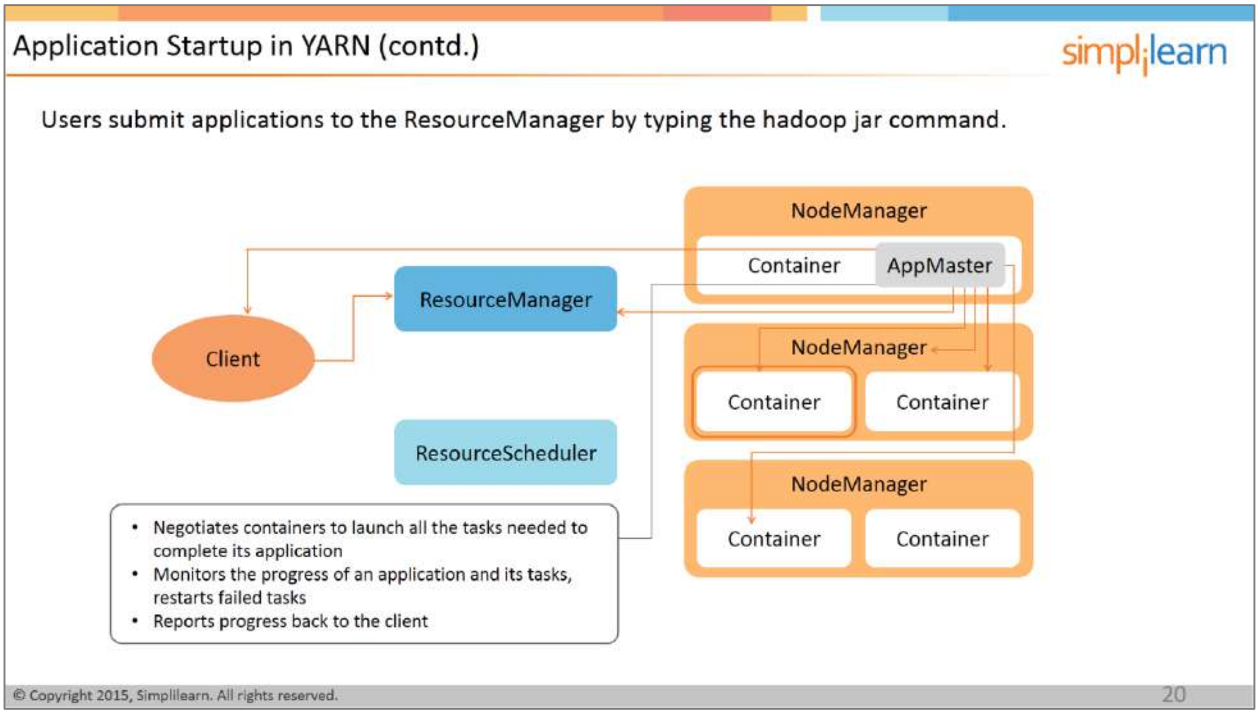
The ResourceManager grants a container, expressed as container ID and hostname, which satisfies the requirements of the ApplicationMaster. A container allows an application to use a given amount of resources on a specific host.



Application Startup in YARN (contd.)

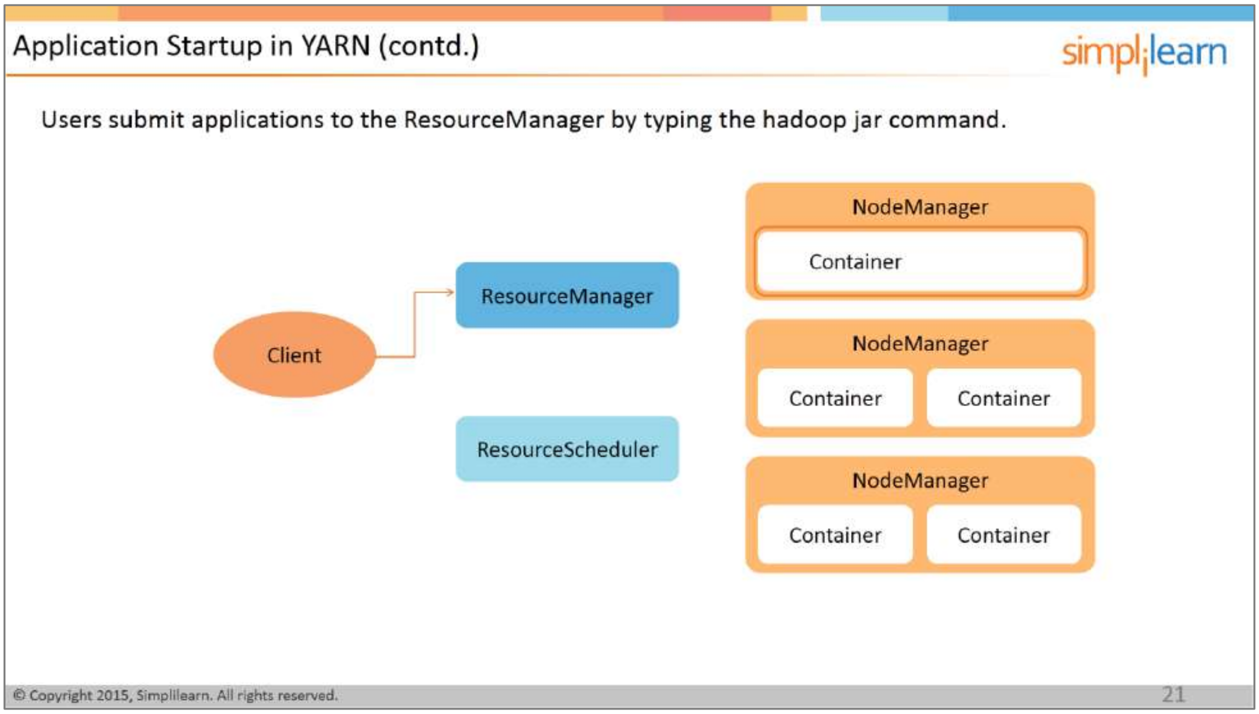
After a container is granted, the ApplicationMaster asks the NodeManager, managing the host on which the container was allocated to use these resources, to launch an application-specific task. This task can be any process written in any framework, such as a MapReduce task or a Giraph task.

The NodeManager does not monitor tasks; it only monitors the resource usage in the containers, for example, it kills a container if it consumes more memory than initially allocated.



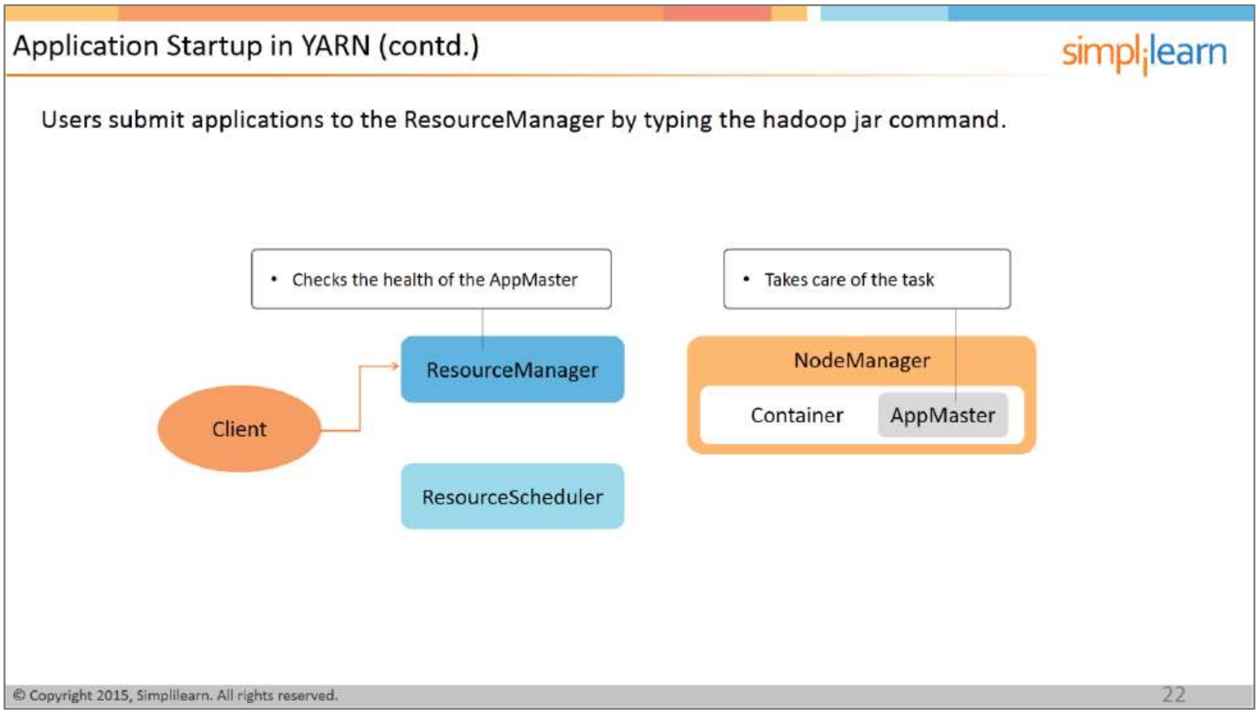
Application Startup in YARN (contd.)

Throughout its life, the ApplicationMaster negotiates containers to launch all of the tasks needed to complete its application. It also monitors the progress of an application and its tasks, restarts failed tasks in newly requested containers, and reports progress back to the client that submitted the application.



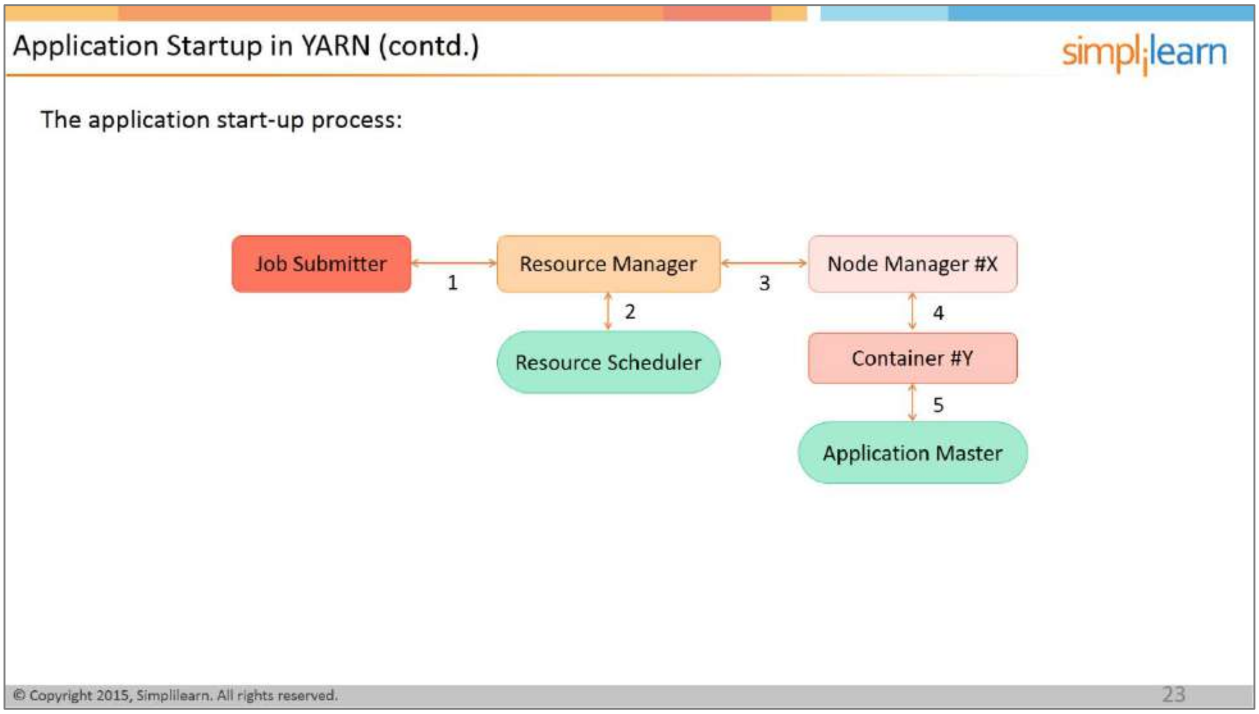
Application Startup in YARN (contd.)

After the application is complete, the ApplicationMaster shuts itself down and releases its own container.



Application Startup in YARN (contd.)

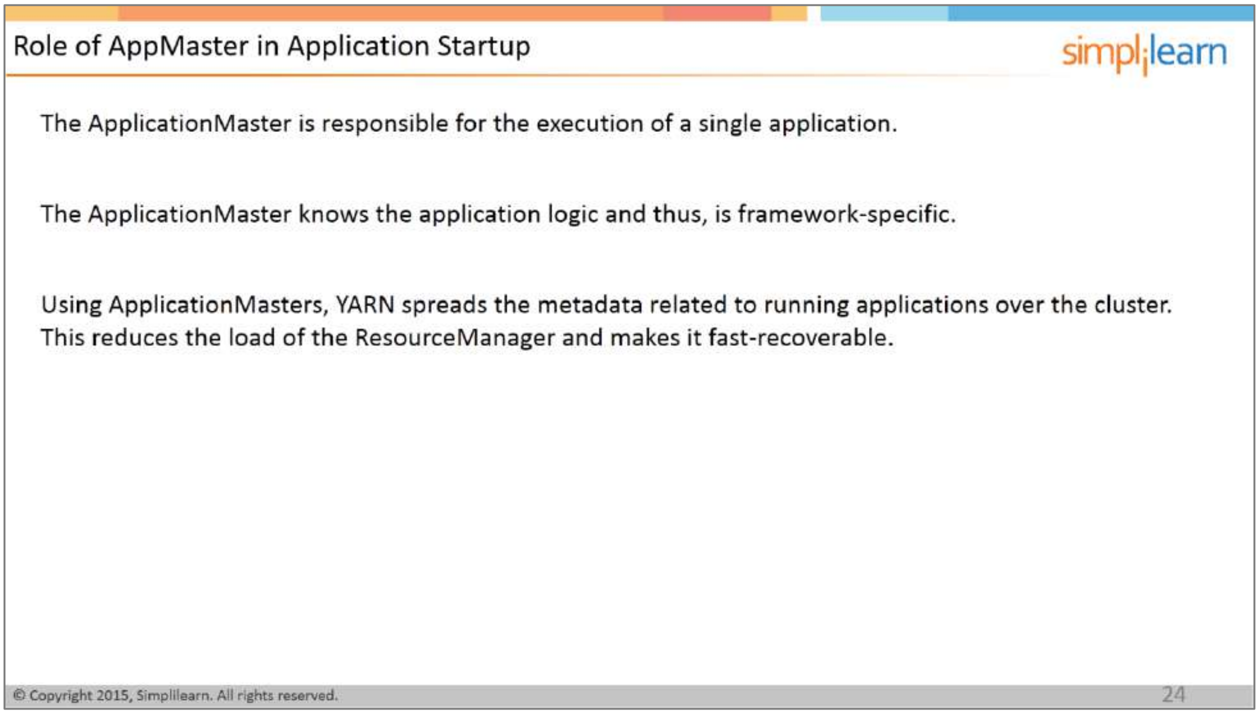
Though the ResourceManager does not perform any monitoring of the tasks within an application, it checks the health of the ApplicationMasters. If the ApplicationMaster fails, it can be restarted by the ResourceManager in a new container. In short, the ResourceManager takes care of the ApplicationMasters, while the ApplicationMasters takes care of tasks.



Application Startup in YARN (contd.)

The application startup can be summarized as follows:

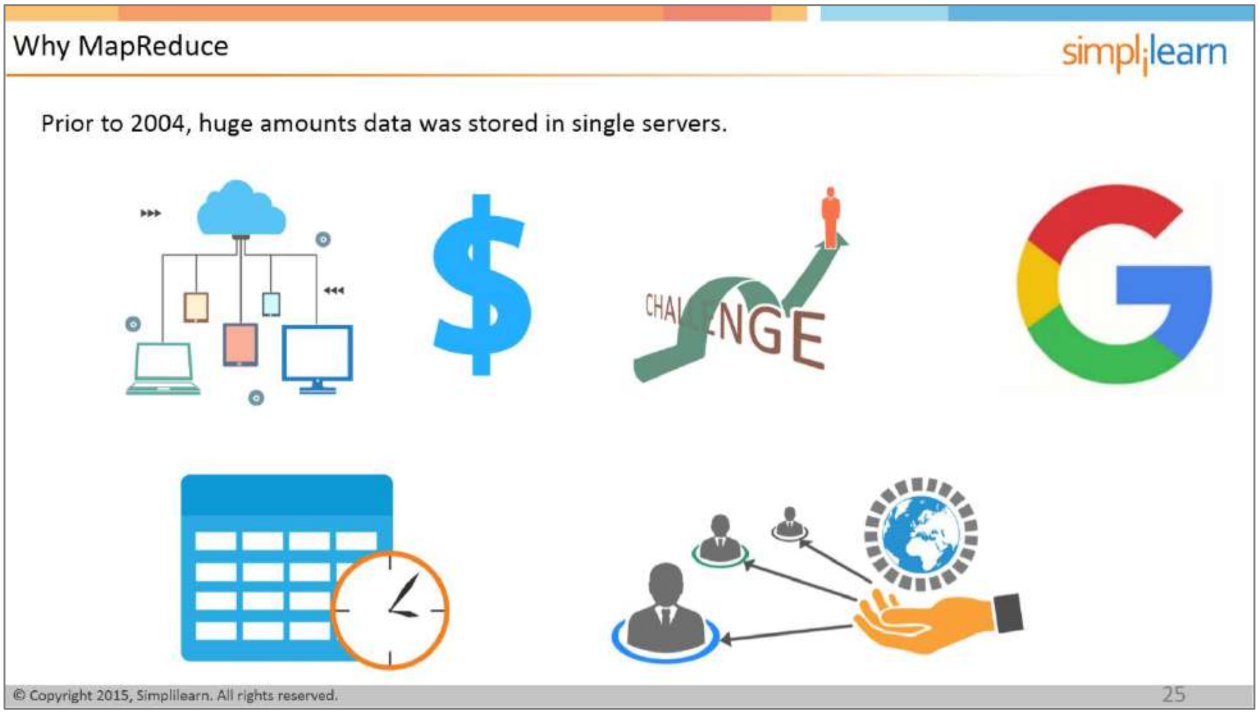
- a client submits an application to the Resource Manager
- the ResourceManager allocates a container
- the ResourceManager contacts the related NodeManager
- the NodeManager launches the container
- the Container executes the Application Master



Role of AppMAster in Application Startup

The ApplicationMaster is responsible for the execution of a single application. It asks the Resource Scheduler for containers and executes specific programs, example the main of a Java class, on the allocated containers. The ApplicationMaster knows the application logic and thus, is framework-specific. The MapReduce framework provides its own implementation of an ApplicationMaster.

The ResourceManager is a single point of failure in YARN. Using ApplicationMasters, YARN spreads the metadata related to running applications over the cluster. This reduces the load of the ResourceManager and makes it fast-recoverable.

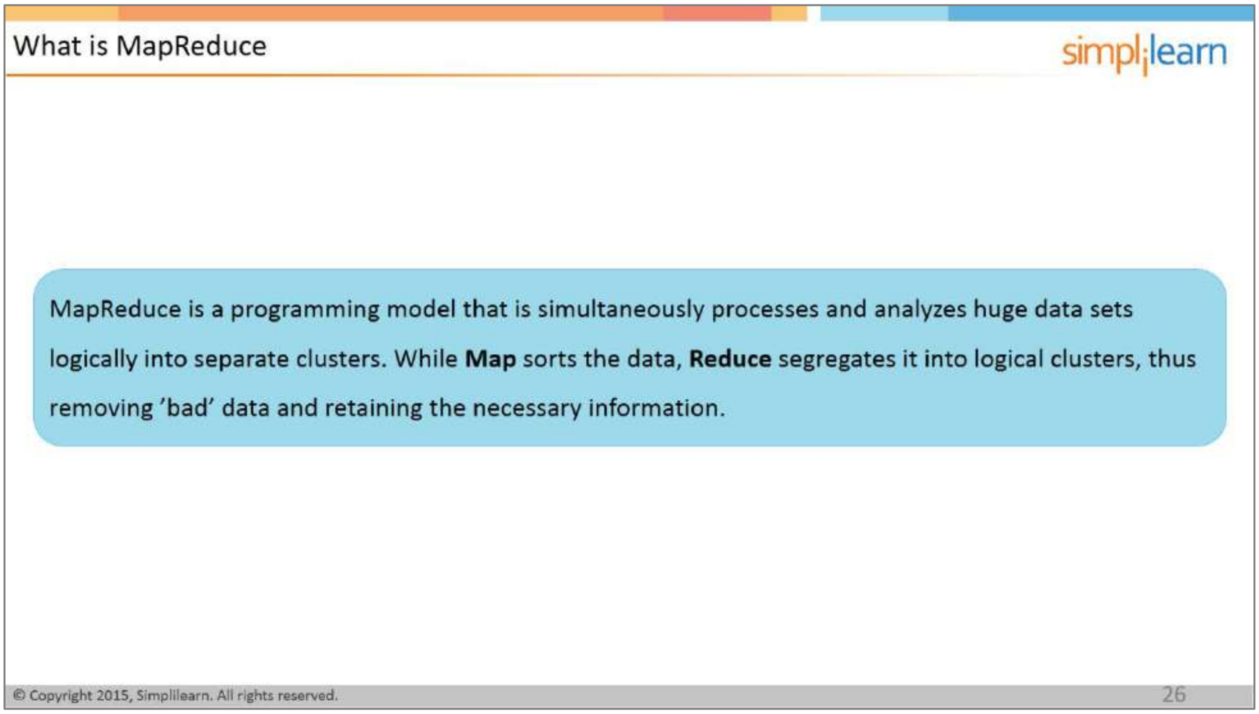


Why MapReduce

Prior to 2004, huge amounts data was stored in single servers. If any program would run a query for data stored in multiple servers, logical integration of search results & analyses of data was a nightmare. Not to mention the massive efforts and expenses were involved. The threat of data loss, challenge of data backup, and reduced scalability resulted in snowballing the issue.

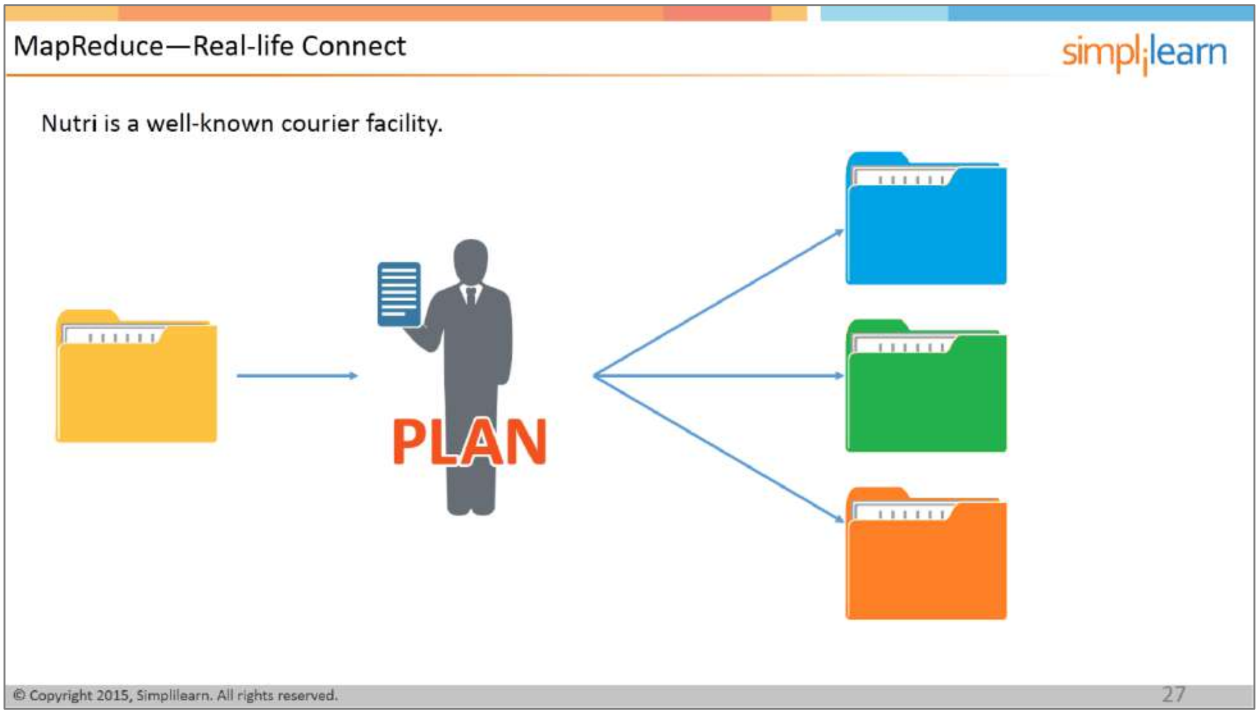
To counter this, Google introduced MapReduce in December 2004. With this, analysis of data sets that would have taken 8-10 days was done probably in less than 10 minutes. Queries could run simultaneously on multiple servers, and now logically integrate search results and analyze data in real-time.

The USP of MapReduce is its fault-tolerance and scalability.



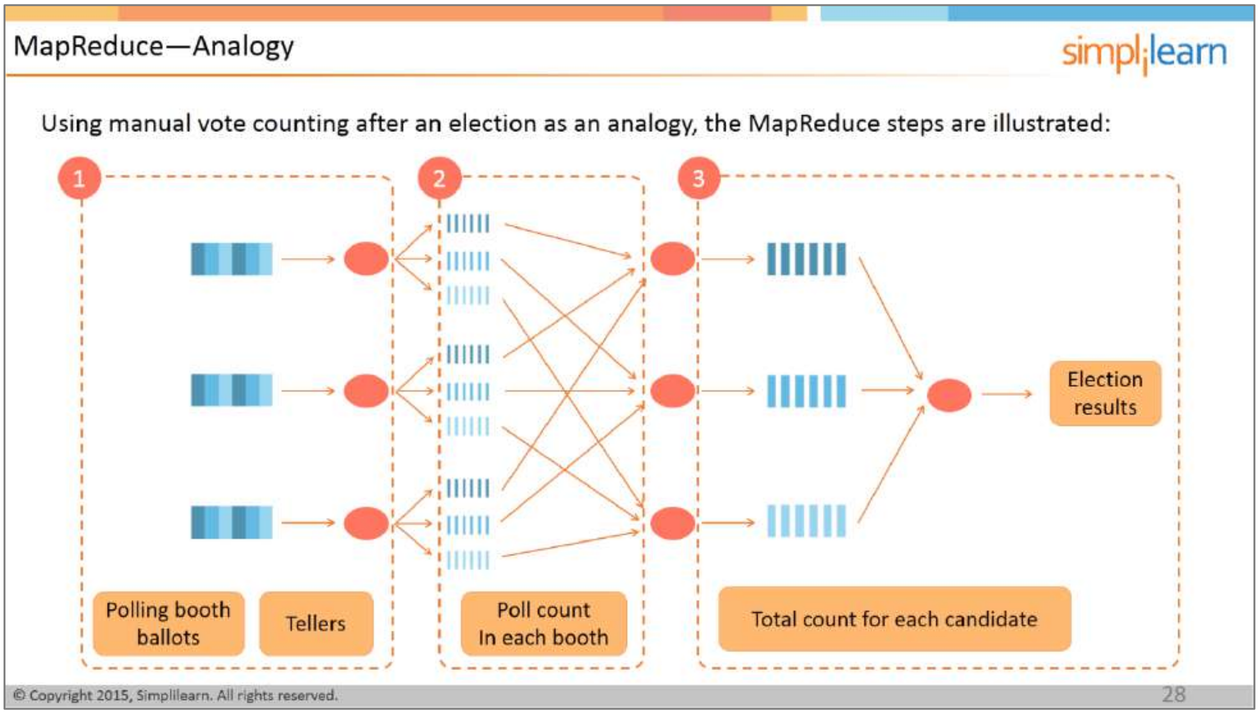
What is MapReduce

MapReduce is a programming model that is simultaneously processes and analyzes huge data sets logically into separate clusters. While **Map** sorts the data, **Reduce** segregates it into logical clusters, thus removing 'bad' data and retaining the necessary information.



MapReduce—Real-life Connect

Nutri is a well-known courier facility. It transports documents across the globe. When the staff receive a courier, they color code it based on the country to be transported. The dispatch staff then segregates the courier by the tagged color code. Hence, the reception functions as "Map," and the dispatch team as "Reduce."



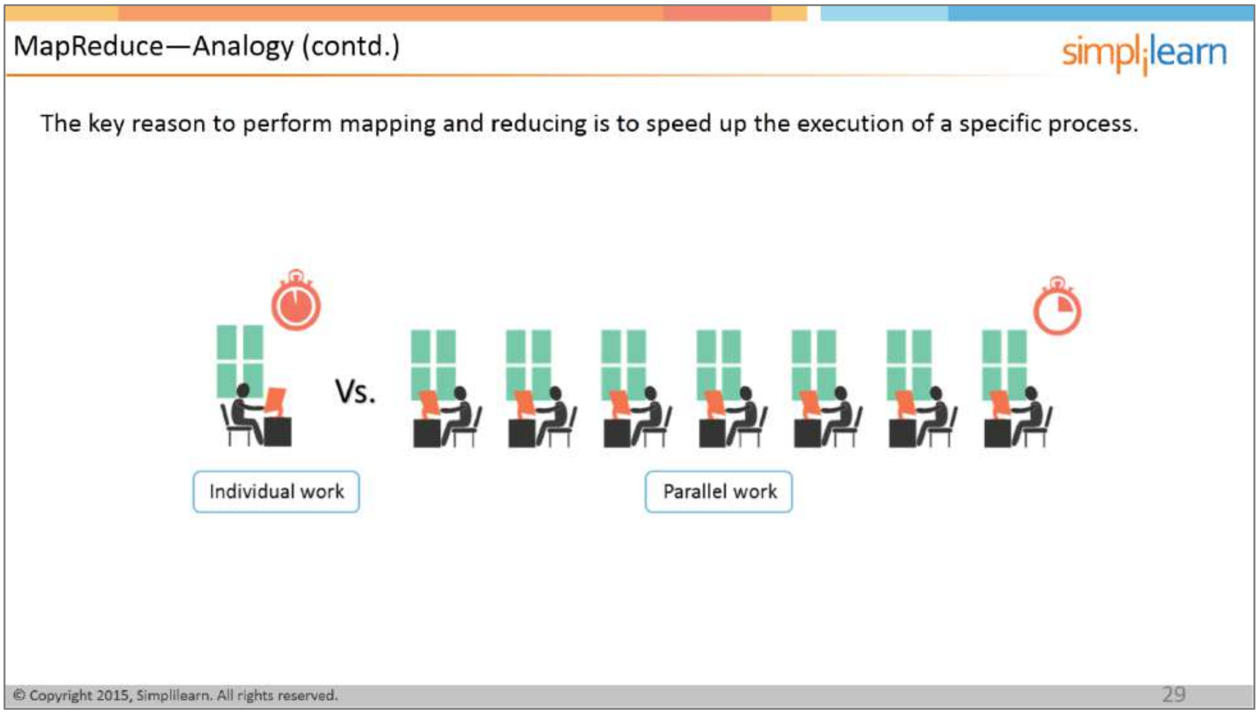
MapReduce Analogy

The MapReduce steps are listed here that represent manual vote counting after an election as an analogy.

Step 1: Each poll booth's ballot papers are counted by a teller. This is a pre-MapReduce step called input splitting.

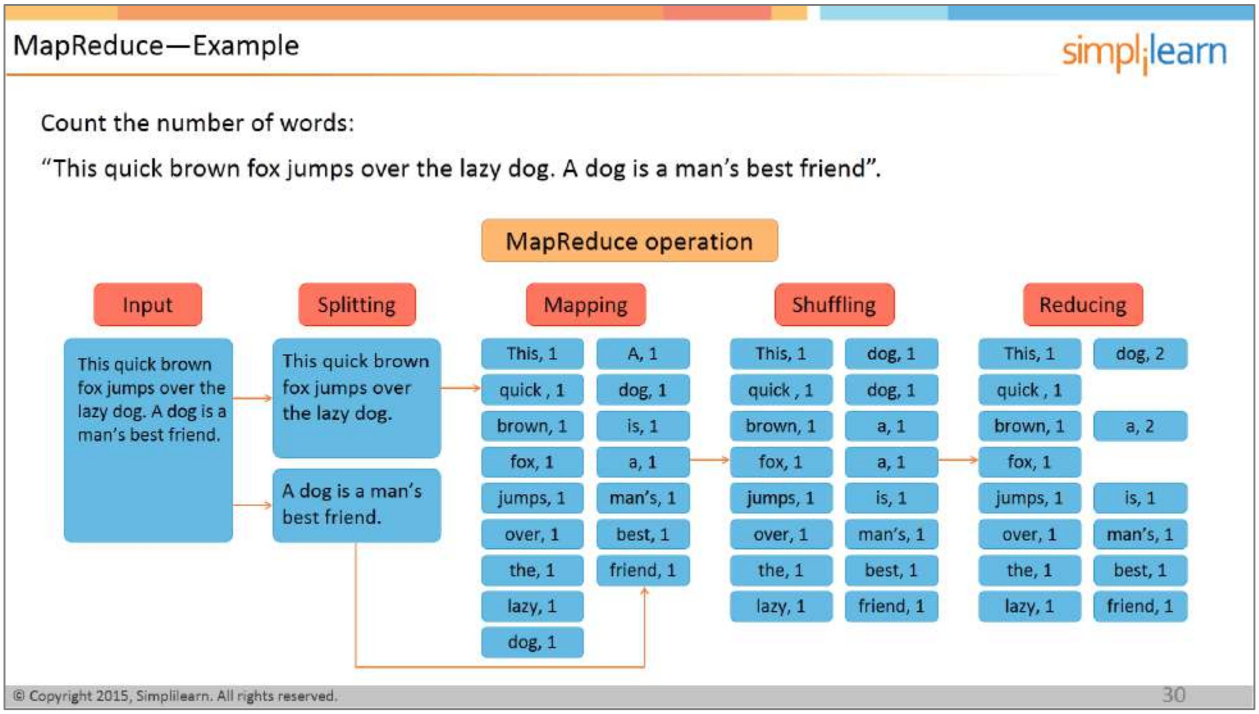
Step 2: Tellers of all booths count the ballot papers in parallel. As multiple tellers are working on a single job, the job execution time will be faster. This is called the Map method.

Step 3: The ballot count of each booth under the assembly and parliament seat positions is found, and the total count for the candidates is generated. This is known as the Reduce method. Thus, map and reduce help to execute the job more quickly than can be done using individual counting.



MapReduce—Analogy (contd.)

As we have seen previously, the analogy of vote counting helps in understanding the usefulness of MapReduce. The key reason to perform mapping and reducing is to speed up the job execution of a specific process. This can be done by splitting a process into a number of tasks, thus enabling parallelism. If one person counts all of the ballot papers or waits for another to finish the ballot count, it could take a month to receive the election results. When many people count the ballots simultaneously, the results are obtained in one or two days. This is how MapReduce works.



MapReduce—Example

The MapReduce operation is explained using a real-time problem. The job is to perform a word count of the given paragraph. The paragraph is, "This quick brown fox jumps over a lazy dog. A dog is a man's best friend."

The MapReduce process consists of input, splitting, mapping, shuffling, and reducing phases.

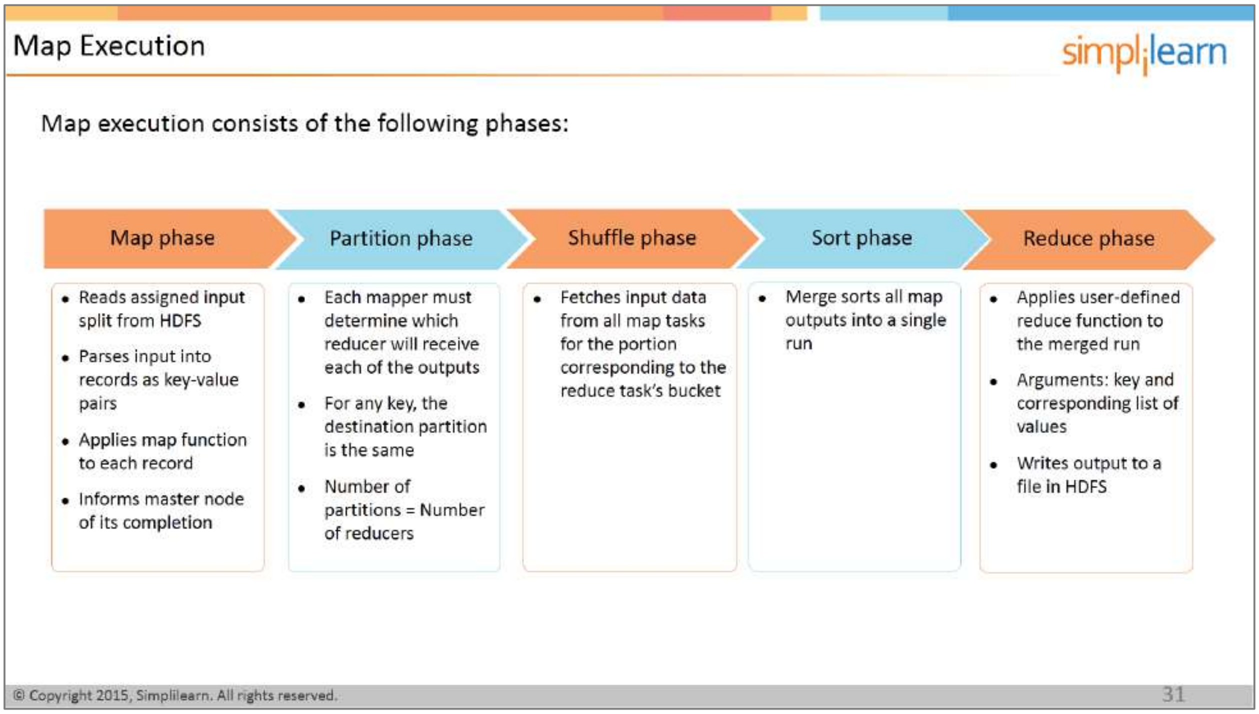
The Input phase refers to providing data for which the MapReduce process is to be performed. The paragraph is used as the input here.

The Splitting phase refers to converting a job submitted by the client into a number of tasks. In this example, the job is split into two tasks.

The Mapping phase refers to generating a key-value pair for the input. Since this example is about counting words, the sentence is split into words by using the substring method to generate words from lines. The Mapping phase will ensure that the words generated are converted into keys, and a default value of one is allotted to each key.

The Shuffling phase refers to sorting the data based on the keys. As shown, the words are sorted in ascending order.

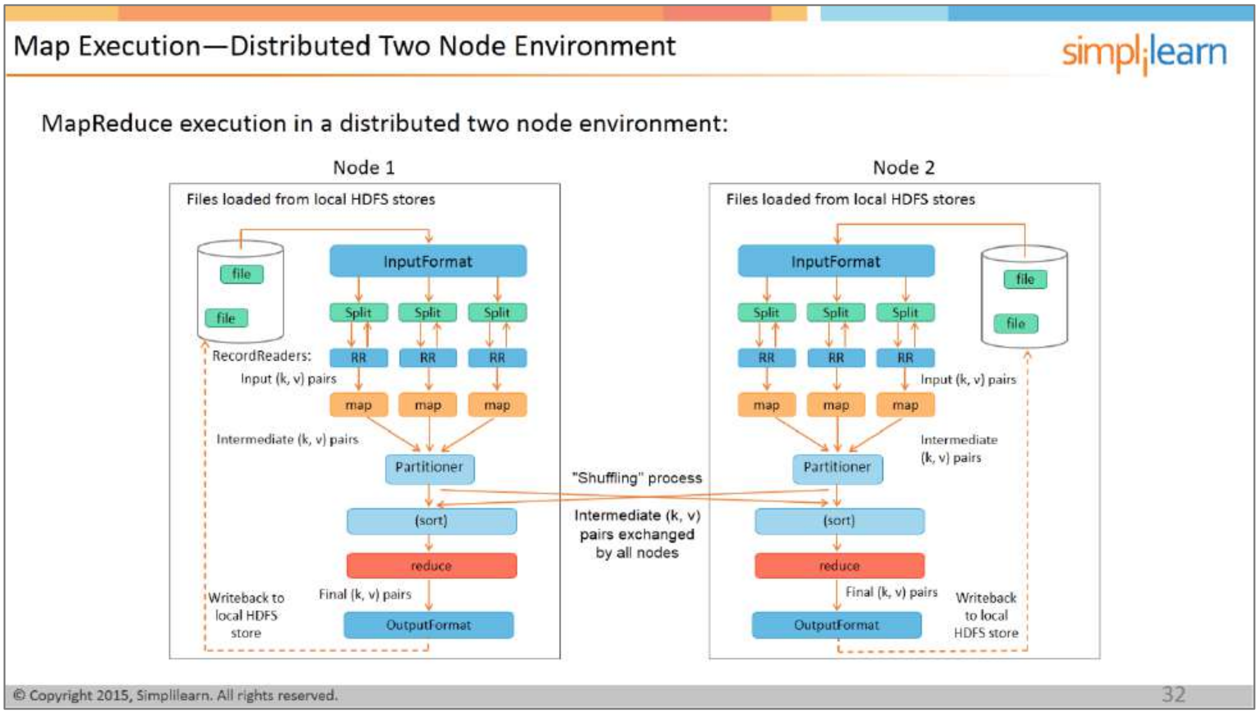
The last phase is the Reducing phase. In this phase, the data is reduced based on the repeated keys by incrementing the value. The word "dog" and the letter "a" are repeated. Therefore, the reducer will delete the key and increment the value depending on the number of occurrences of the key. This is how the MapReduce operation is performed.



Map Execution

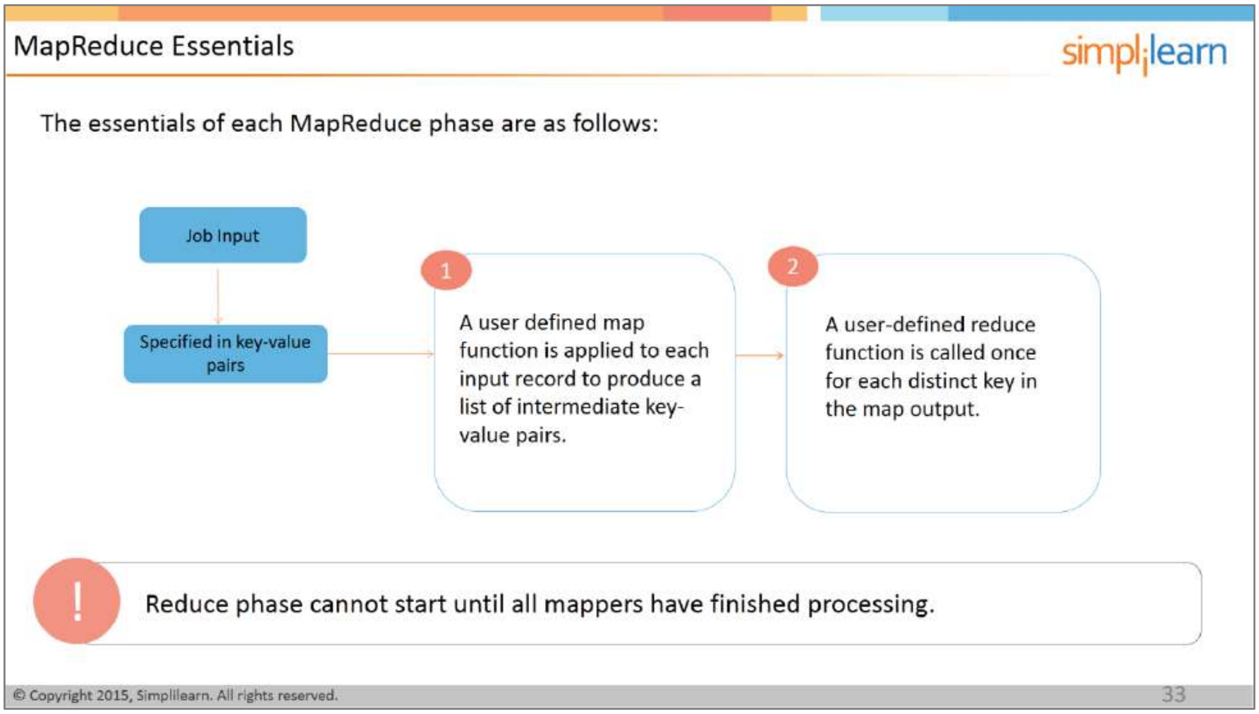
Map execution consists of five phases: map phase, partition phase, shuffle phase, sort phase, and reduce phase.

- **Map Phase:** In the map phase, the assigned input split is read from HDFS, where a split could be a file block by default. Furthermore, input is parsed into records as key-value pairs. The map function is applied to each record to return zero or more new records. These intermediate outputs are stored in the local file system as a file. They are sorted first by bucket number and then by key. At the end of the map phase, information is sent to the master node of its completion.
- **Partition phase:** In the partition phase, each mapper must determine which reducer will receive each of the outputs. For any key, regardless of which mapper instance generated it, the destination partition is the same. Note that the number of partitions will be equal to the number of reducers.
- **Shuffle Phase:** In the shuffle phase, input data is fetched from all map tasks for the portion corresponding to the reduce task's bucket.
- **Sort Phase:** In the sort phase, a merge-sort of all map outputs occurs in a single run.
- **Reduce Phase:** In the reduce phase, a user-defined reduce function is applied to the merged run. The arguments are a key and corresponding list of values. The output is written to a file in HDFS.



Map Execution—Distributed Two-Node Environment

The mappers on each of the nodes are assigned an input split of blocks. Based on the input format, the Record Reader reads the split as a key-value pair. The map function is applied to each record to return zero or more new records. These intermediate outputs are stored in the local file system as a file. Thereafter, a partitioner assigns the records to a reducer. In the shuffling phase, the intermediate key-value pairs are exchanged by all nodes. The key-value pairs are then sorted by applying the key and reduce function. The output is stored in HDFS based on the specified output format.



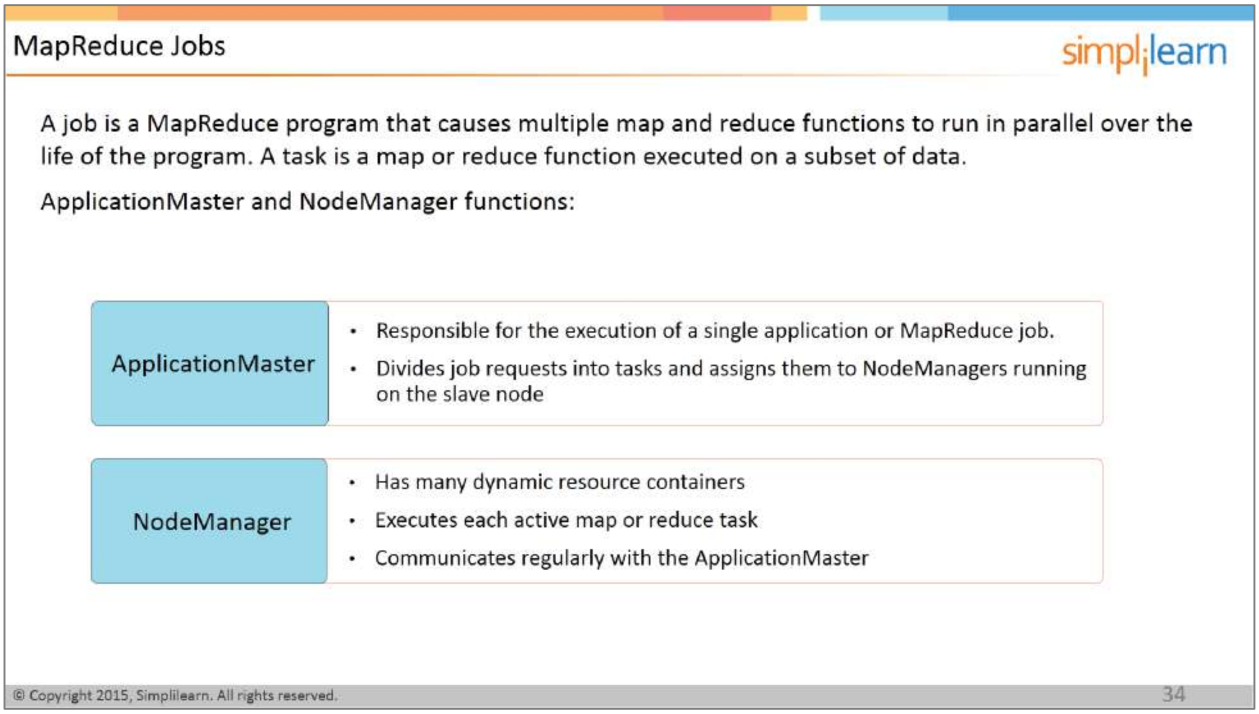
MapReduce Essentials

The essentials of each MapReduce phase are shown.

The job input is specified in key-value pairs.

Each job consists of two stages. First, a user defined map function is applied to each input record to produce a list of intermediate key-value pairs. Second, a user-defined reduce function is called once for each distinct key in the map output. Then the list of intermediate values associated with that key is passed.

The number of reduce tasks can be defined by the users. Each reduce task is assigned a set of record groups which are intermediate records corresponding to a group of keys. For each group, a user-defined reduce function is applied to the record values in that group. The reduce tasks read from every map task, and each read returns the record groups for that reduce task. Note that the reduce phase cannot start until all mappers have finished processing.



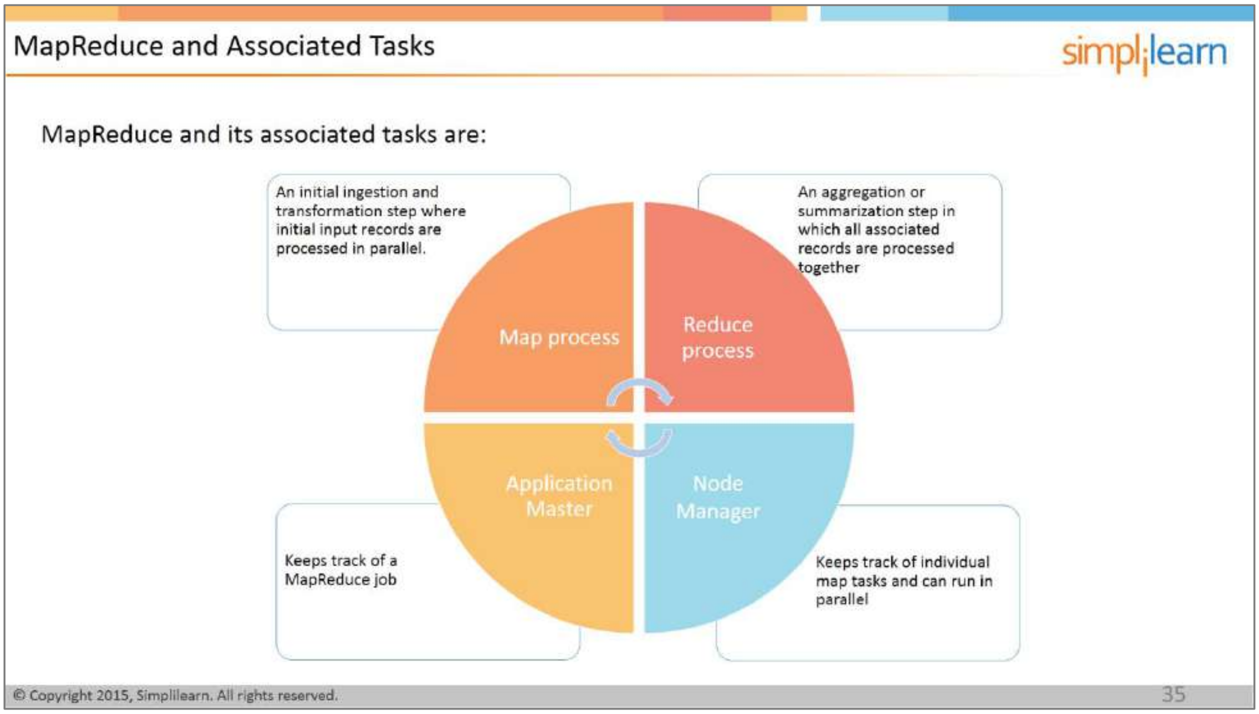
MapReduce Jobs

A job is a full MapReduce program which typically causes multiple map and reduce functions to be run in parallel over the life of the program. Many copies of map and reduce functions are forked for parallel processing across the input data set. A task is a map or reduce function executed on a subset of data.

With this understanding of "job" and "task," the ApplicationMaster and NodeManager functions become easy to comprehend.

The ApplicationMaster is responsible for the execution of single application or MapReduce job. It divides the job requests into tasks and assigns those tasks to Node Managers running on the slave node.

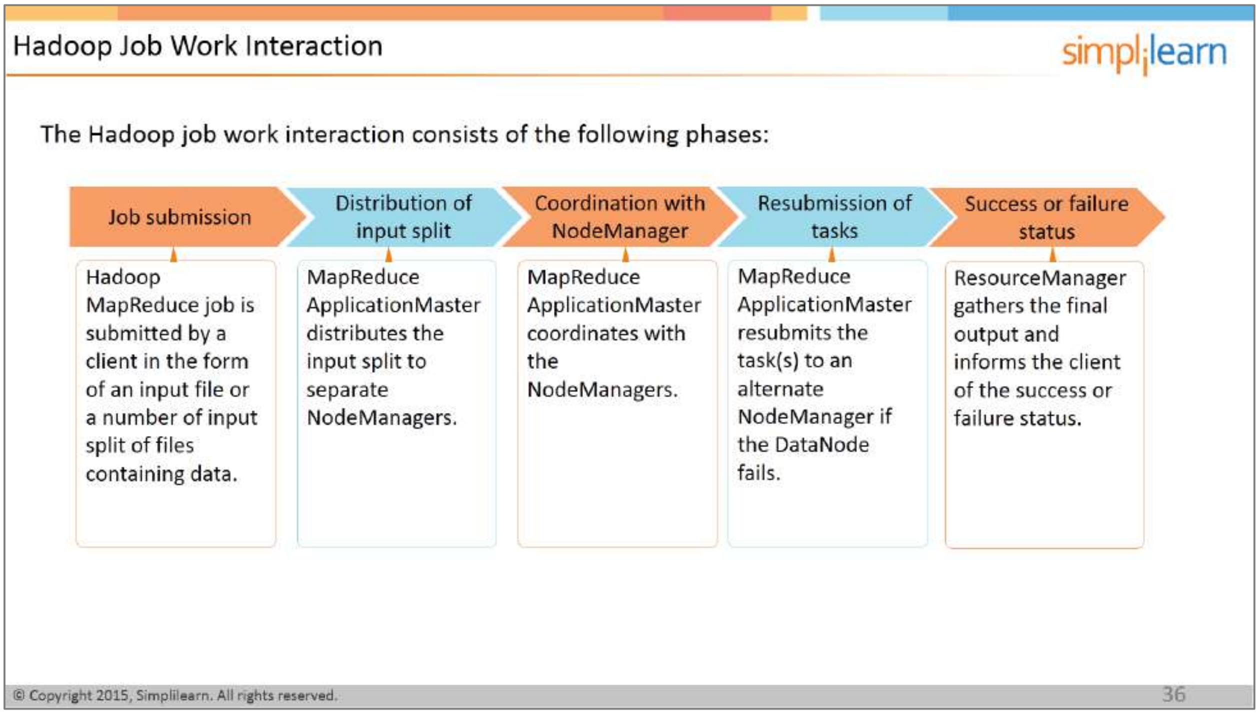
The NodeManager has a number of dynamically created resource containers. The size of a container depends on the amount of resources it contains, such as memory, CPU, disk, and network IO. It executes map and reduce task by launching these containers when instructed by the MapReduce Application Master.



MapReduce and Associated Tasks

MapReduce and associated tasks are listed here. The Map process is an initial step to process individual input records in parallel. The reduce process is all about summating the output with a defined goal as coded in business logic. NodeManager keeps track of individual map tasks and can run in parallel. A map job runs as part of container execution by NodeManager on a particular DataNode.

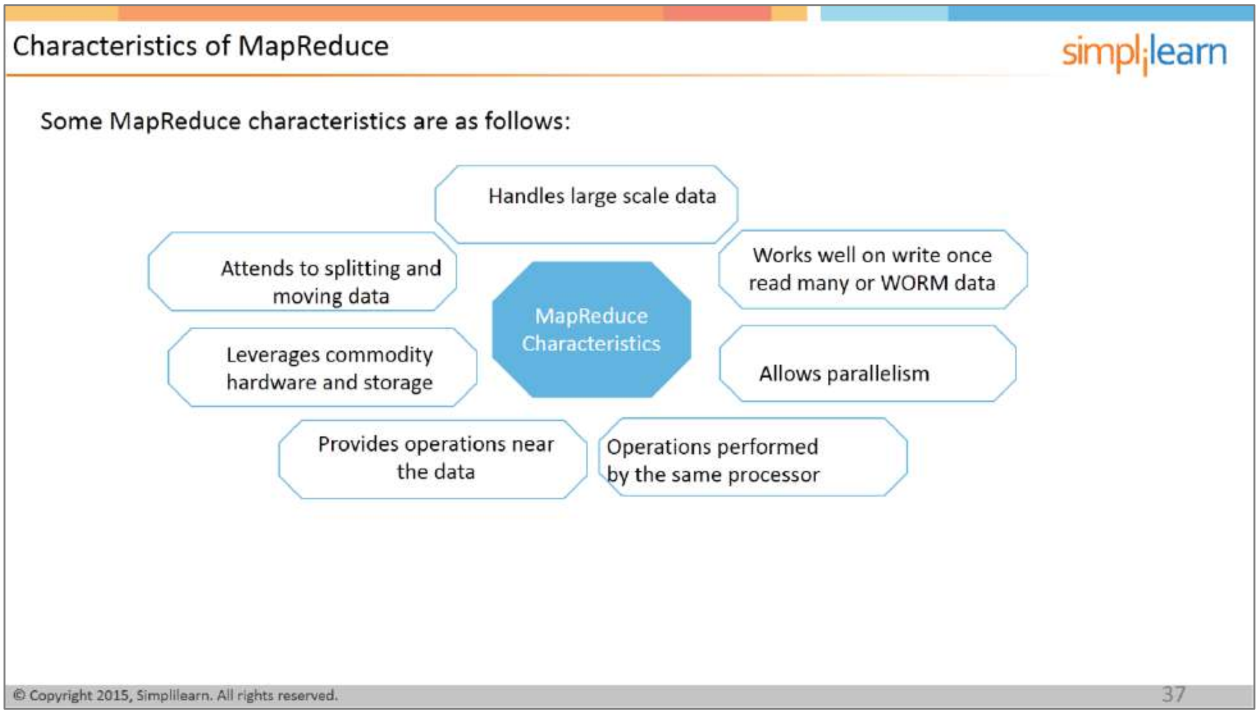
The ApplicationMaster keeps track of a MapReduce job.



Hadoop Job Work Interaction

The flow diagram represents the Hadoop Job work interaction.

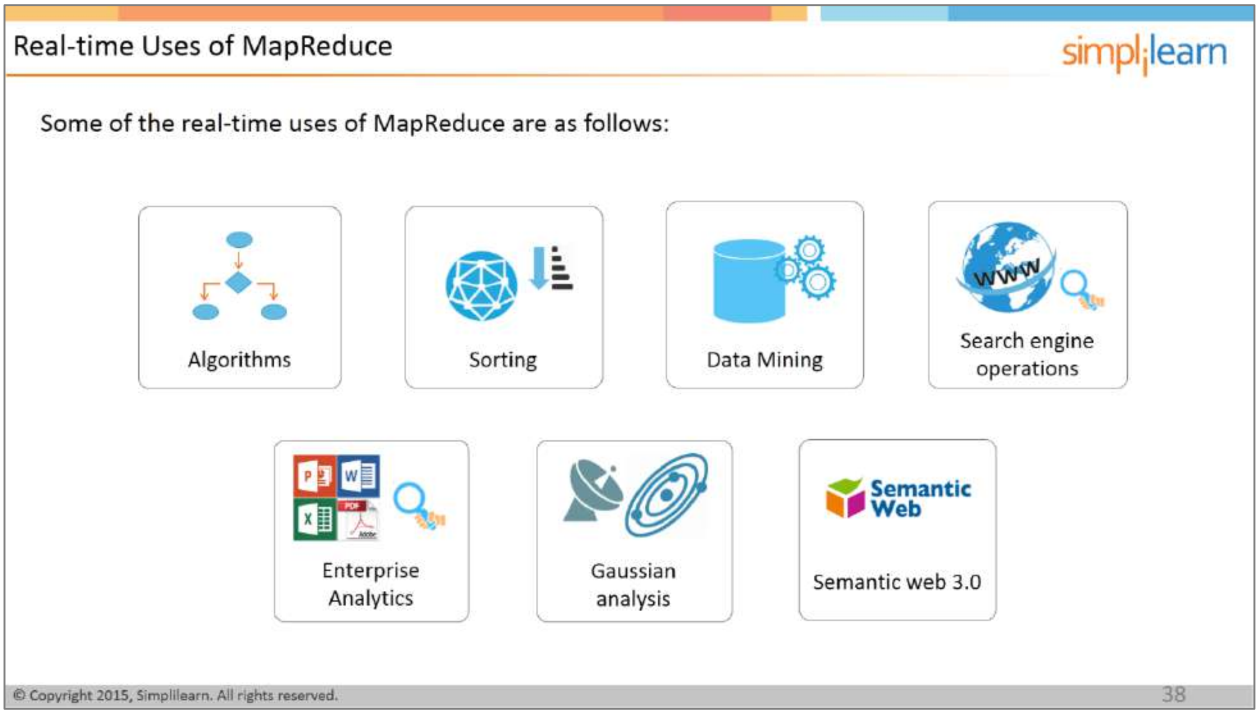
- Initially, a Hadoop MapReduce job is submitted by a client in the form of an input file or a number of input split of files containing data.
- MapReduce ApplicationMaster distributes the input split to separate NodeManagers.
- MapReduce ApplicationMaster coordinates with the NodeManagers.
- MapReduce ApplicationMaster resubmits the task(s) to an alternate Node Manager if the DataNode fails
- ResourceManager gathers the final output and informs the client of the success or failure status.



Characteristics of MapReduce

Some MapReduce characteristics are:

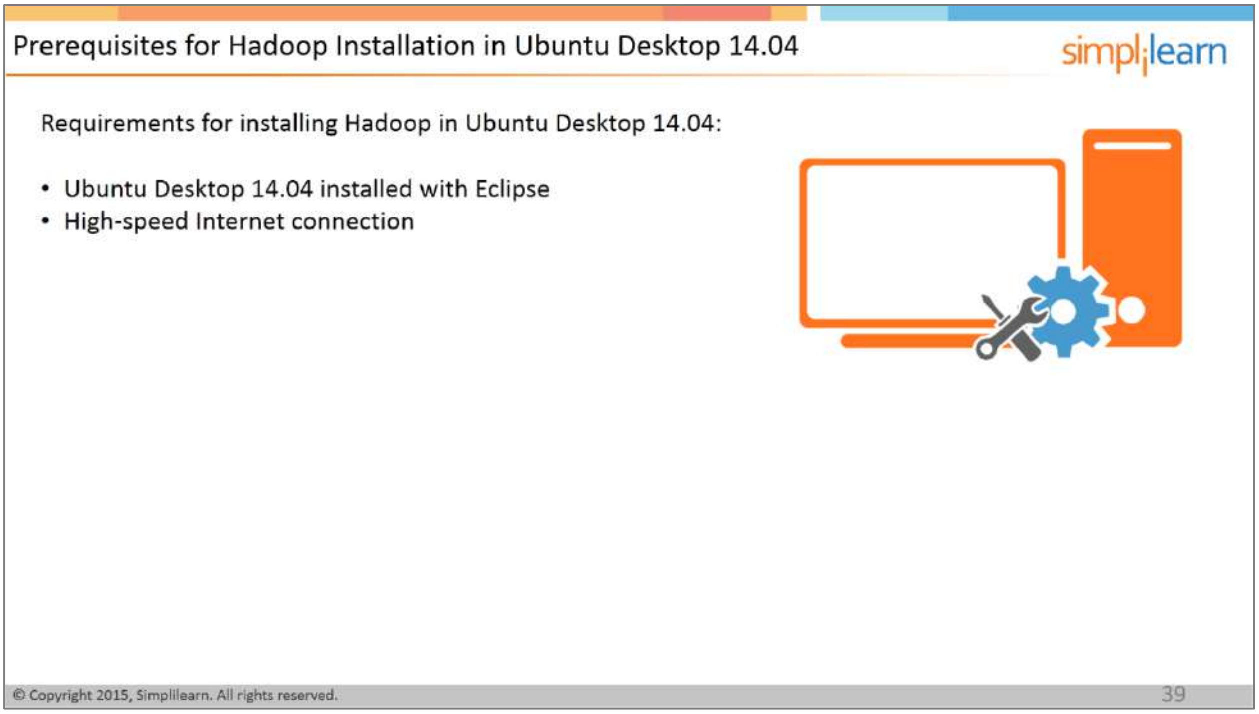
- MapReduce is designed to handle very large scale data in the range of petabytes and exabytes.
- It works well on write once and read many data, also known as WORM data.
- MapReduce allows parallelism without mutexes.
- The Map and Reduce operations are performed by the same processor.
- Operations are provisioned near the data as data locality is preferred.
- Commodity hardware and storage is leveraged in MapReduce.
- The runtime takes care of splitting and moving data for operations.



Real-time Uses of MapReduce

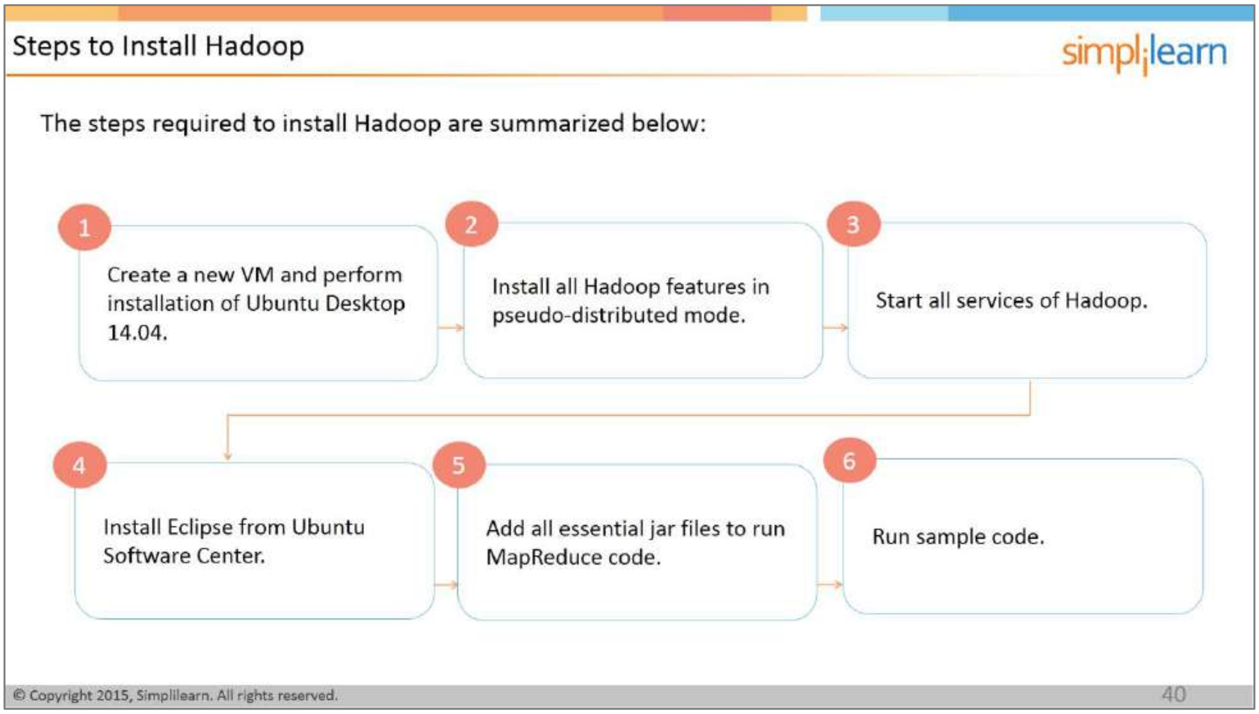
Some of the real-time uses of MapReduce are as follows:

- Simple algorithms such as grep, text-indexing, and reverse indexing
- Data-intensive computing such as sorting
- Data mining operations like Bayesian classification
- Search engine operations like keyword indexing, ad rendering, and page ranking
- Enterprise analytics
- Gaussian analysis for locating extra-terrestrial objects in astronomy
- Semantic web and web 3.0



Prerequisites for Hadoop Installation in Ubuntu Desktop 14.04

Ubuntu Desktop 14.04 VM installed with Eclipse and a high-speed Internet connection is required to install Hadoop in Ubuntu Desktop 14.04.



Steps to Install Hadoop

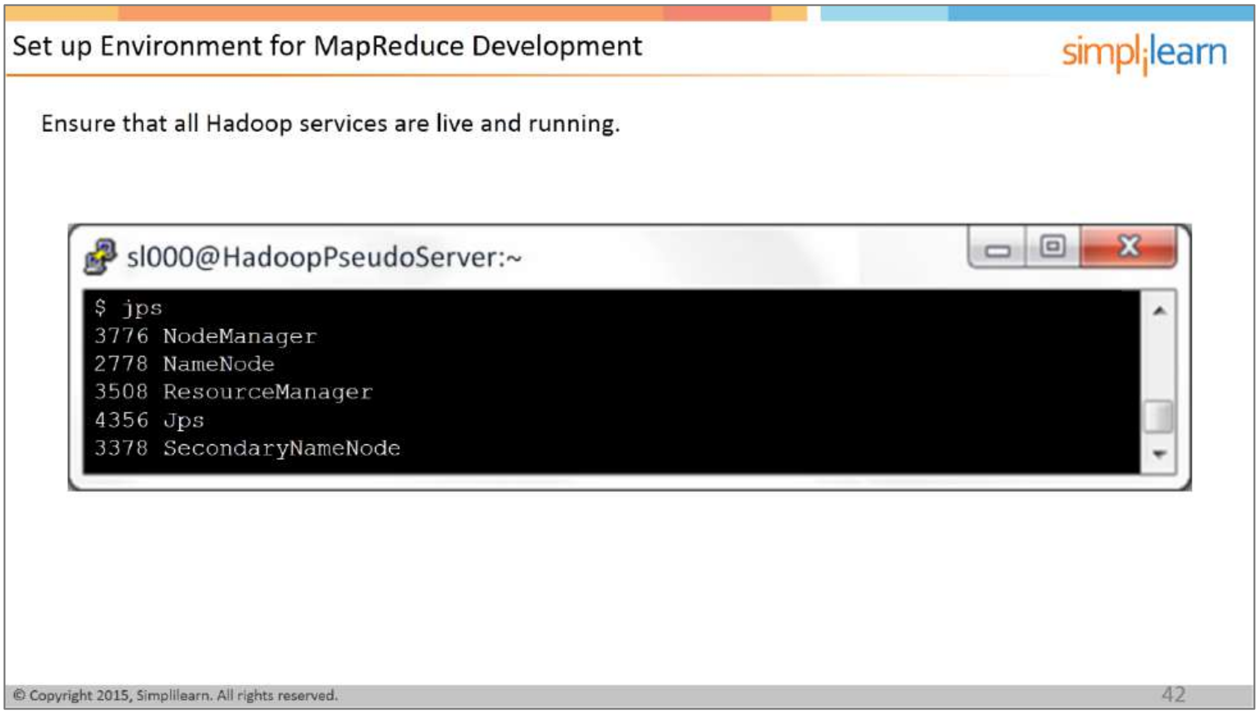
The steps required to install Hadoop in Ubuntu Desktop 14.04 LTS are listed:

1. Create a new VM and install Ubuntu Desktop 14.04 LTS Operating System.
2. Install all Hadoop features in pseudo-distributed mode.
3. Start all Hadoop services.
4. Install Eclipse from Ubuntu Software Centre.
5. Add all essential jar files to run MapReduce code.
6. Finally, run a sample code to ensure that the environment settings are correct.

Business Scenario

Olivia is the EVP of IT operations with Nutri Worldwide, Inc., and Tim Burnet, the AVP of IT-infra ops, is assigned to one of her projects. Olivia has asked Tim to analyze two large datasets and set up mock activities for anyone new to MapReduce. One of the dataset includes a novel, and the other includes a weather report for a city. Tim wants to install Eclipse on his system to create a new project to run default MapReduce programs. Some of the MapReduce programs that he wishes to run include counting the words from a paragraph and finding average temperatures.

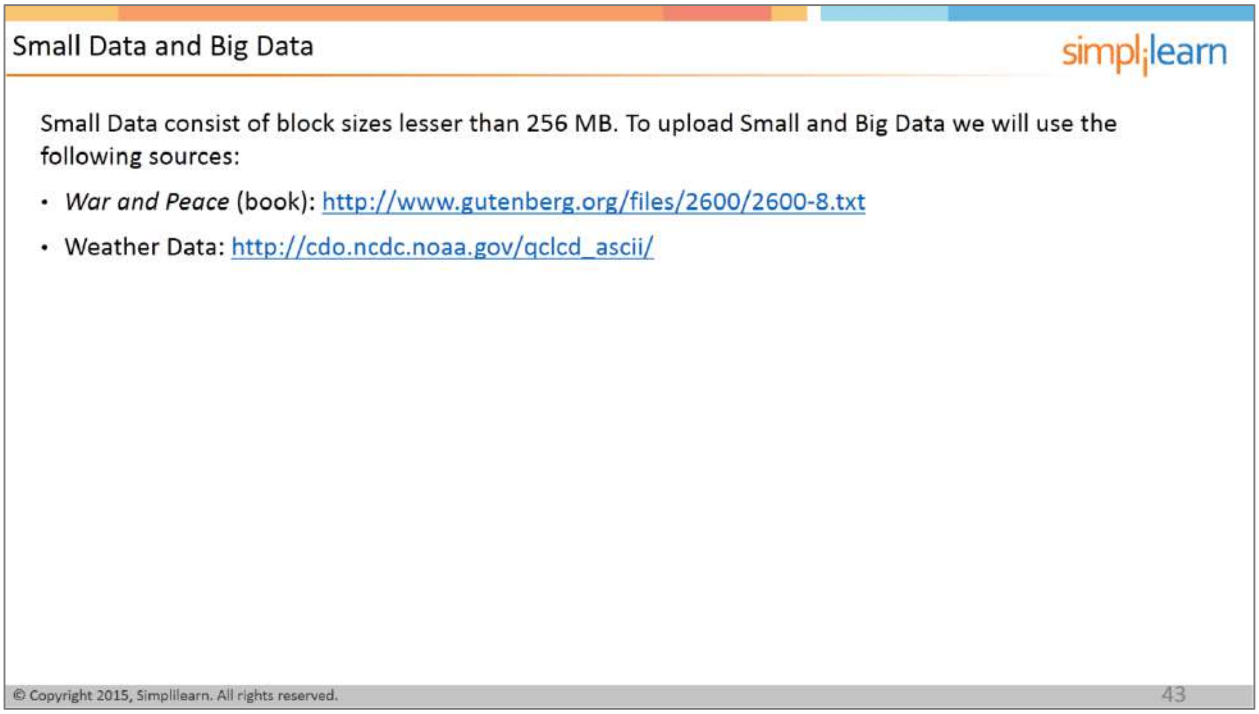
The demos in this lesson will illustrate how to install Ubuntu Desktop OS, install Eclipse, create projects, and write simple MapReduce programs.



Set Up Environment for MapReduce Development

Ensure that all Hadoop services are live and running. This can be verified by applying two steps:

- Use the command 'jps'
- Then look for all five services: NameNode, DataNode, NodeManager, ResourceManager, and SecondaryNameNode.

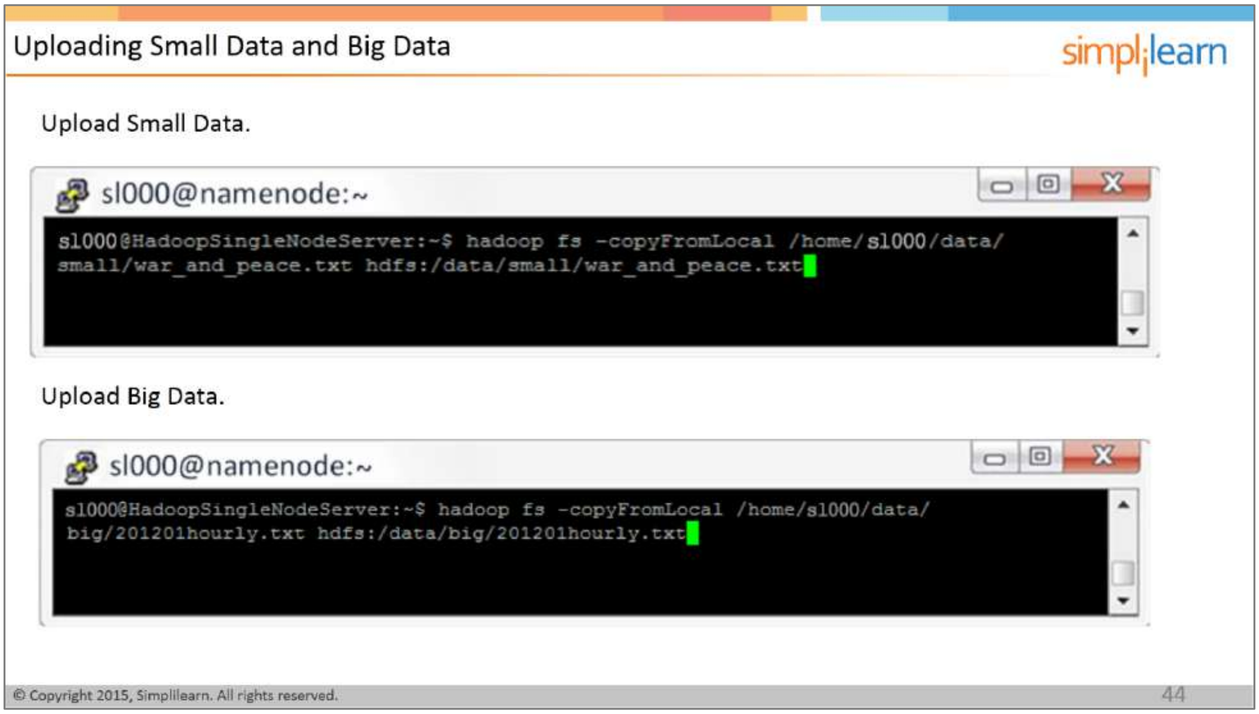


Small Data and Big Data

Small Data consist of block sizes lesser than 256 MB. To upload Small and Big Data we will use the following sources:

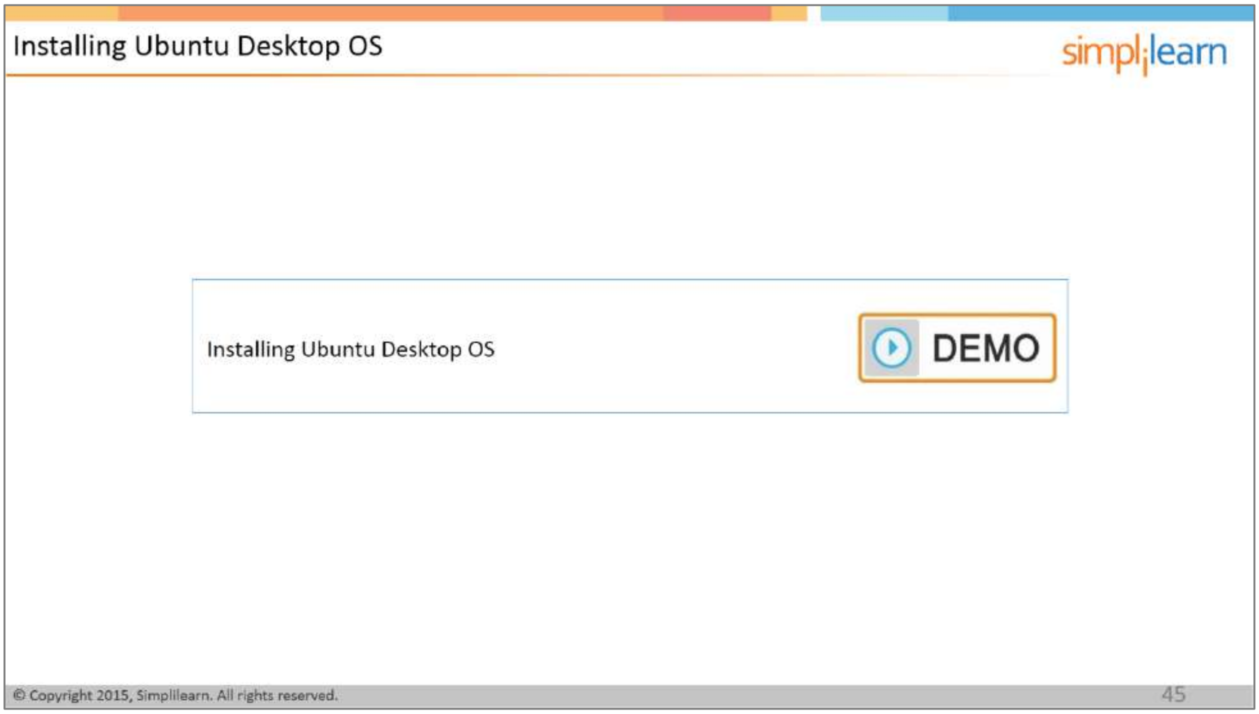
- *War and Peace* which is from an e-book website, and
- Weather data that is updated on an hourly basis.

The sample dataset from these sources will be provided and can be used to perform MapReduce operations. To download the data click the given URL.



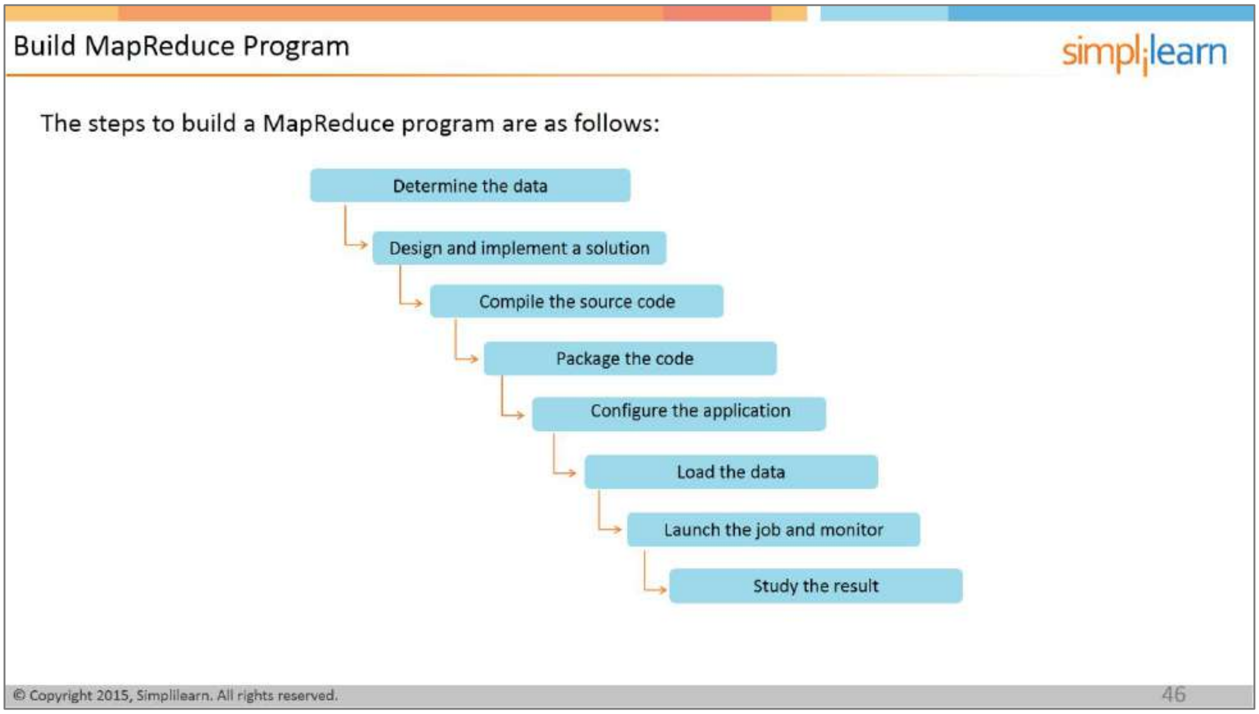
Uploading Small Data and Big Data

The command to upload any Data, Big or Small, from the local system to HDFS is hadoop (space) fs (space) -copyFromLocal (space) source file address (space) destination file address.



Installing Ubuntu Desktop OS

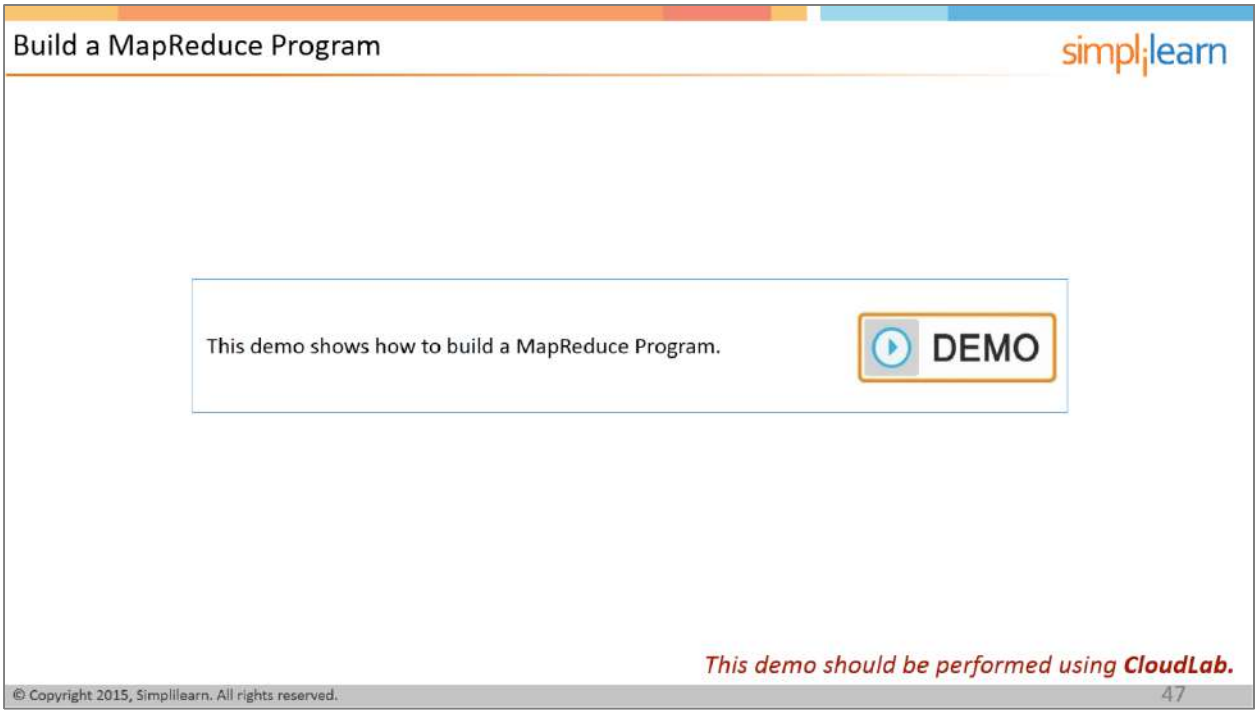
1. Select the language and start the installation by clicking the Install Ubuntu button in the Welcome wizard.
2. Click the Continue button and proceed with the installation.
3. In the Installation Type, ensure that the Erase disk and install Ubuntu option is selected. Then, click the Continue button.
4. By default, a drive is selected and the disk size that will be consumed is shown. Click the Install Now button.
5. You will see a progress bar of files being copied. Select your location from the dropdown and click Continue.
6. In the keyboard layout, choose English (US) as the language. Click the Continue button.
7. Ensure that all the textboxes are filled with your name and computer name. Enter the password. Retype the password in the password textbox.
8. Click the Continue button.
9. The installation process starts.
10. Click the Restart Now button once the installation is complete.
11. You will now be able to view the desktop screen of Ubuntu Desktop.



Build MapReduce Program

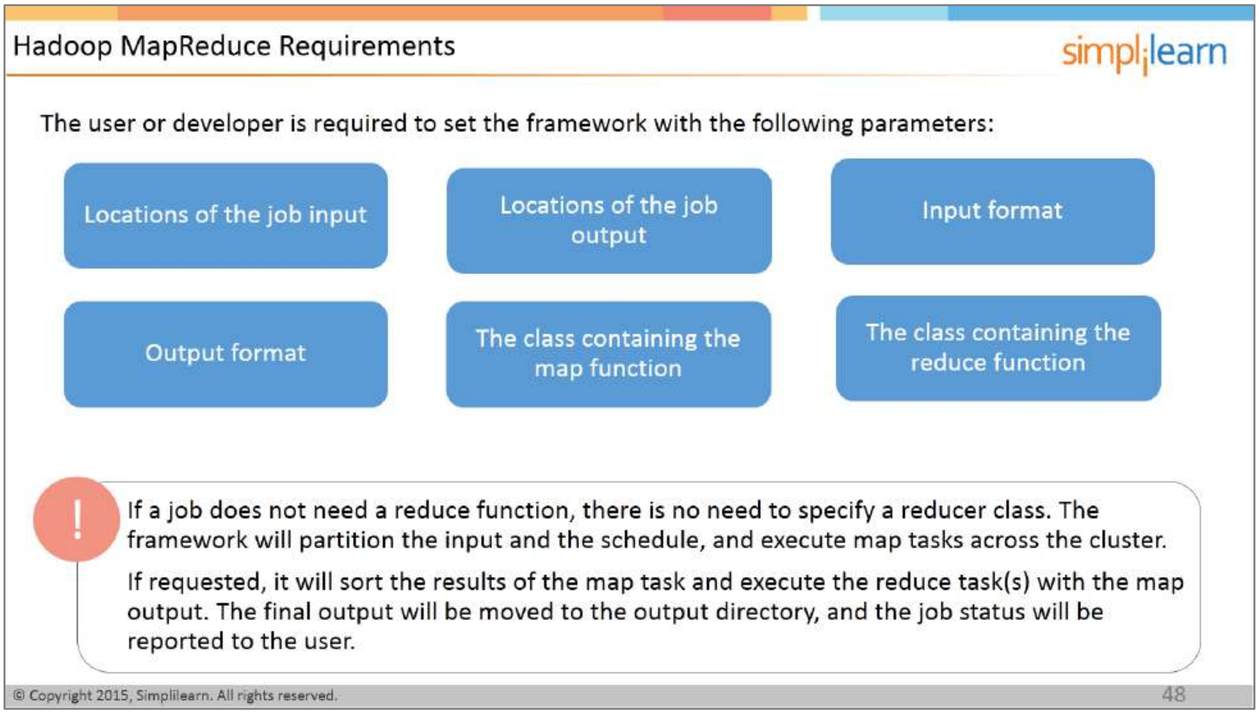
The steps to build a MapReduce program are:

- Determine if the data can be made parallel and solved using MapReduce. For example, we need to analyze whether the data is write once read many that is WORM, in nature.
- Design and implement a solution as mapper and reducer classes.
- Compile the source code with Hadoop core, and package the code as jar executable.
- Configure the application job as to the number of mapper and reducer tasks and to the number of input and output streams.
- Load the data, or use it on previously available data; then launch and monitor the job.
- Study the results.



Let us summarize the steps in this demo:

1. Let us run a MapReduce example shipped along with your Hadoop distribution.
2. In a newer version of Hadoop, the examples are placed in a jar file in the share/hadoop directory.
3. Further, under MapReduce directory, use the specified jar file.
4. We will be running an example to compute the value of 'pi', which is a computation intensive program. The first argument indicates how many maps to create. Here, we use 10 mappers. The second argument indicates how many samples are generated per map; here, we take 100 random samples. So this program uses 10 multiplied by 100, that is, 1000 random points to estimate pi. We could enhance 100 to 10 million and improve accuracy.
5. For 1000 samples, the value of pi is returned as nearly equal to 3.148.

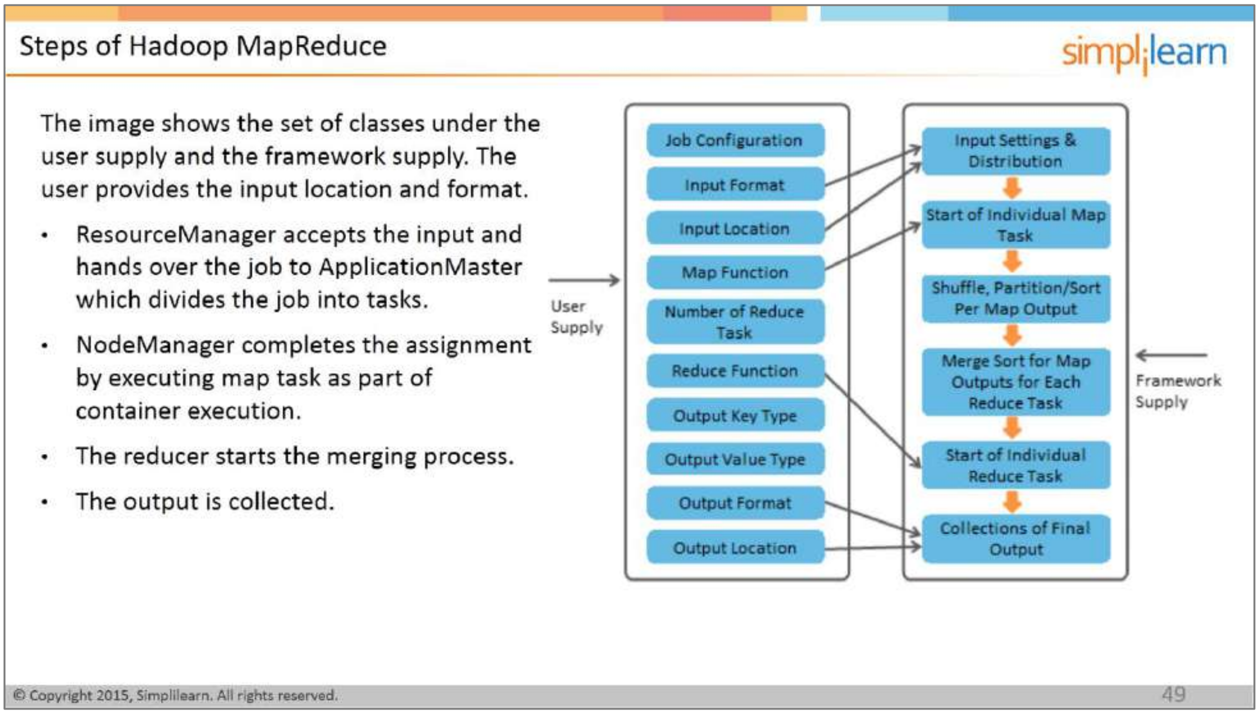


Hadoop MapReduce Requirements

The user or developer is required to set up the framework with the following parameters:

- The locations of the job input in the distributed file system
- The locations of the job output in the distributed file system
- The input format
- The output format
- The class containing the map function
- The class containing the reduce function, which is optional

If a job does not need a reduce function, there is no need to specify a reducer class. The framework will partition the input, schedule, and execute map tasks across the cluster. If requested, it will sort the results of the map task, and it will execute the reduce tasks with the map output. The final output will be moved to the output directory, and the job status will be reported to the user.



Steps of Hadoop MapReduce

The image shows the set of classes under the user supply and the framework supply.

User supply refers to the set of Java classes and methods provided to a Java developer for developing Hadoop MapReduce applications. Framework supply refers to defining the workflow of a job which is followed by all Hadoop services.

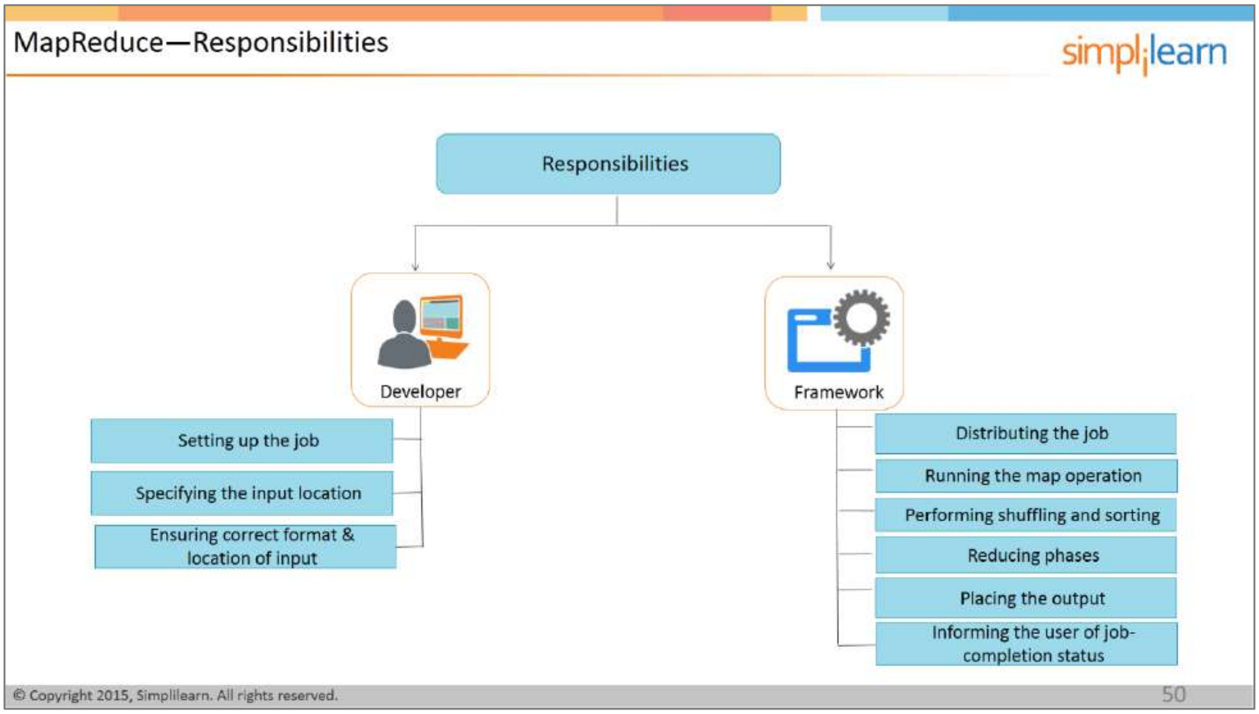
As shown in the image, the user provides the input location and the input format as required by the program logic. Once the ResourceManager accepts the input, the specific job is divided into tasks by ApplicationMaster. Each task is then assigned to an individual NodeManager.

Once the assignment is complete, the NodeManager will start the map task.

It performs shuffling, partitioning, and sorting for individual map outputs.

Once the sorting is complete, the Reducer starts the merging process. This is also called the reduce task.

The final step is collecting the output, which is performed once all the individual tasks are reduced. This reduction is based on programming logic.



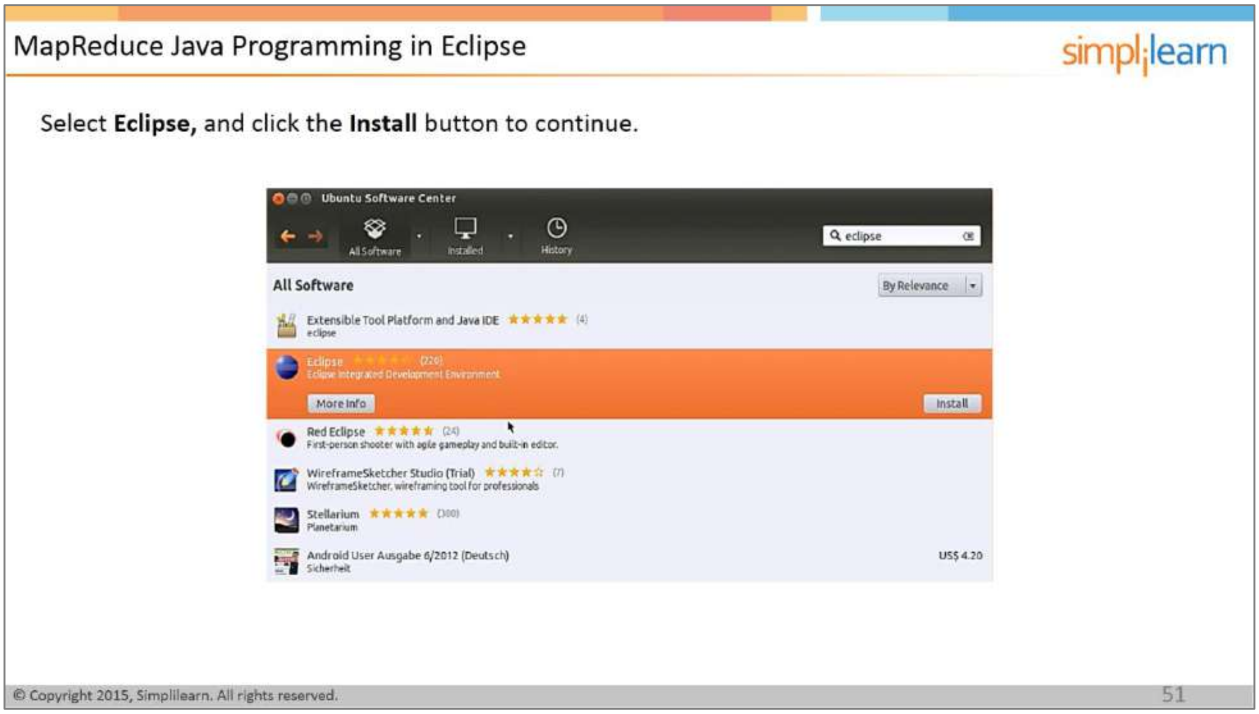
MapReduce—Responsibilities

The basic user or developer responsibilities of MapReduce are:

- Setting up the job,
- Specifying the input location, and
- Ensuring the input is in the expected format and location.

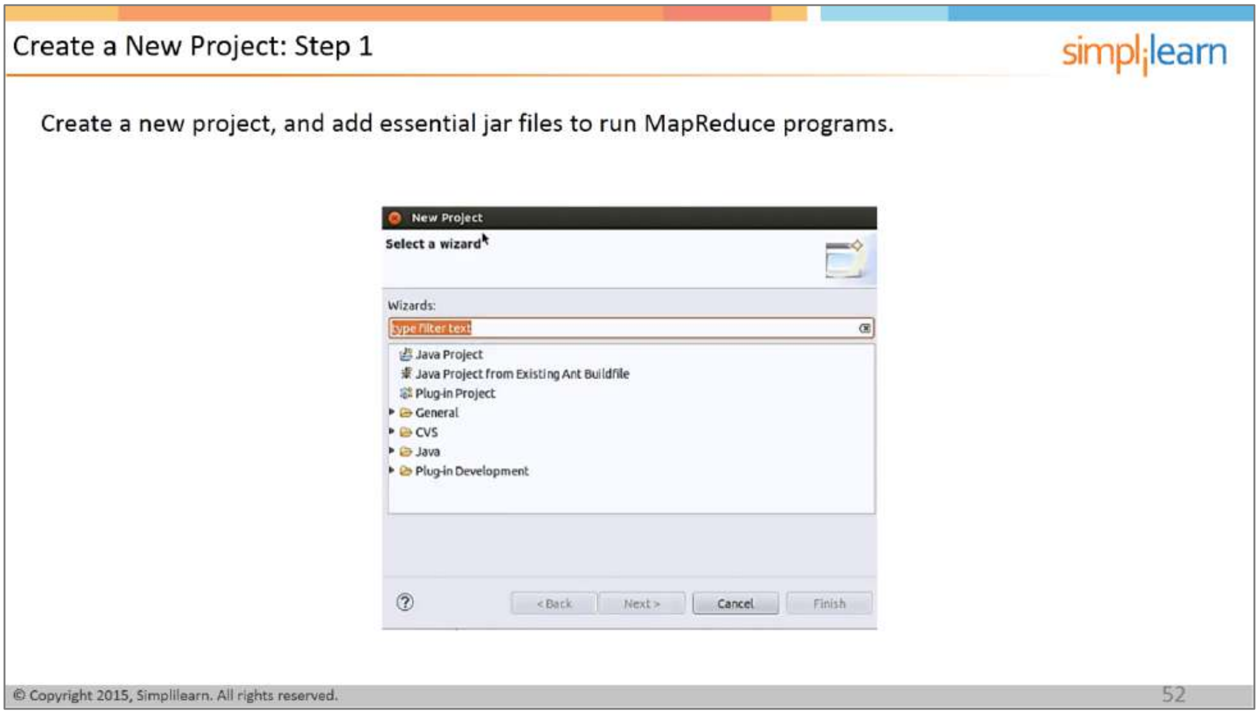
The framework responsibilities of MapReduce are as follows:

- Distributing jobs among the ApplicationMaster and NodeManager nodes of the cluster,
- Running the map operation,
- Performing the shuffling and sorting operations,
- Reducing phases,
- Placing the output in the output directory, and
- Informing the user of the job completion status.



MapReduce Java Programming in Eclipse

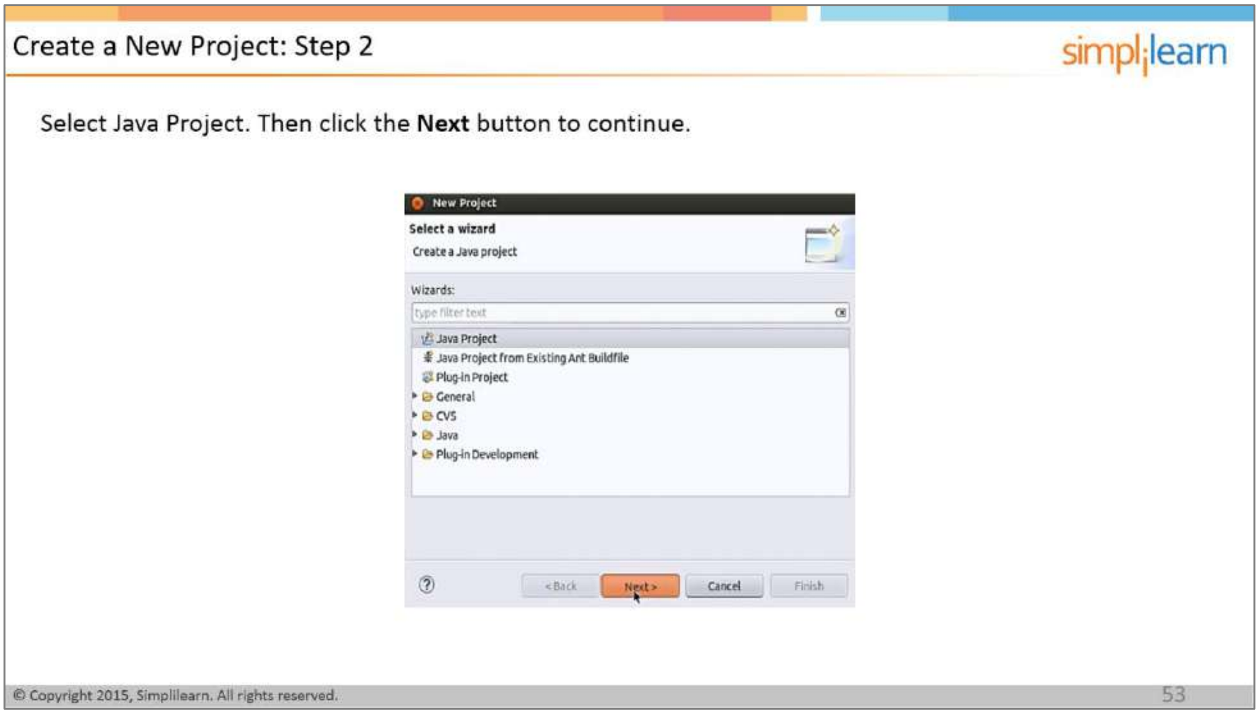
To install Eclipse in the Ubuntu Desktop 14.04, open Ubuntu Software Centre. Type “eclipse” in the search bar as shown. Select Eclipse from the list. Next, click the Install button to continue.



Create a New Project: Step 1

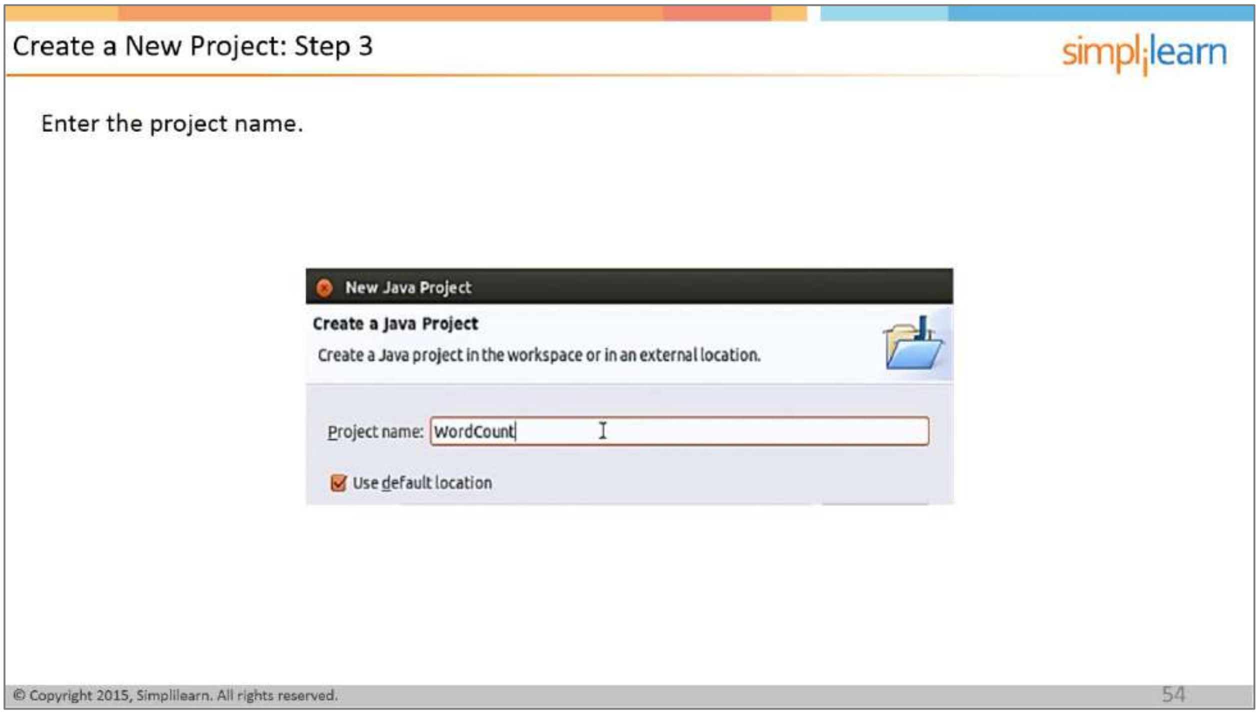
Create a new project and add the essential jar files to run MapReduce programs.

To create a new project, click the **File** menu. Select **New Project**. Alternatively, press **Ctrl+N** to start the wizard of the new project.



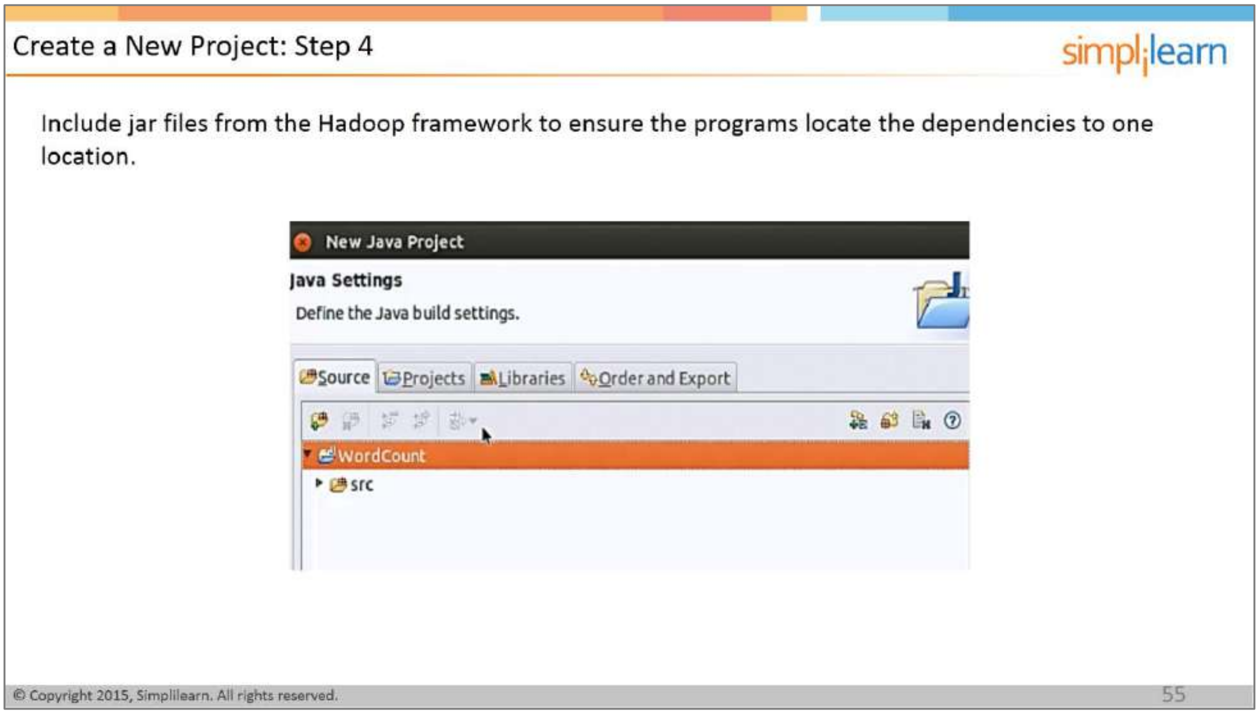
Create a New Project: Step 2

Select **Java Project** from the list. Then click the **Next** button to continue.



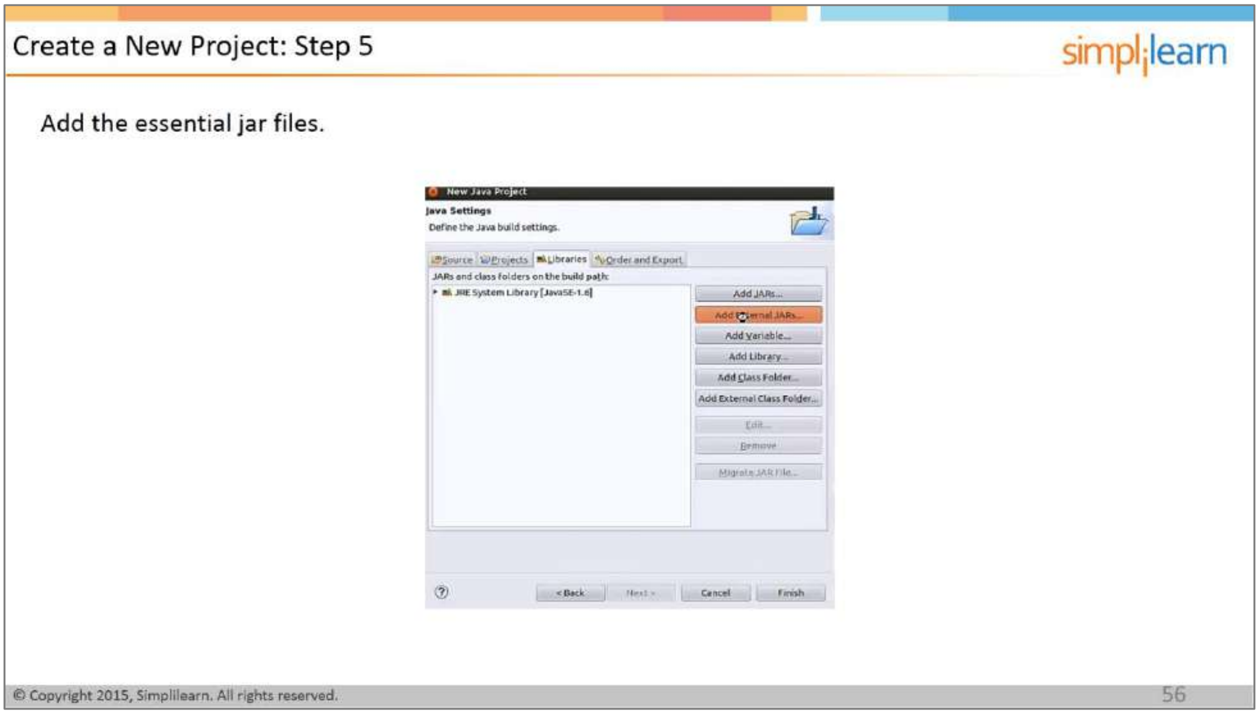
Create a New Project: Step 3

Type the project name as WordCount, and click the **Next** button to continue.



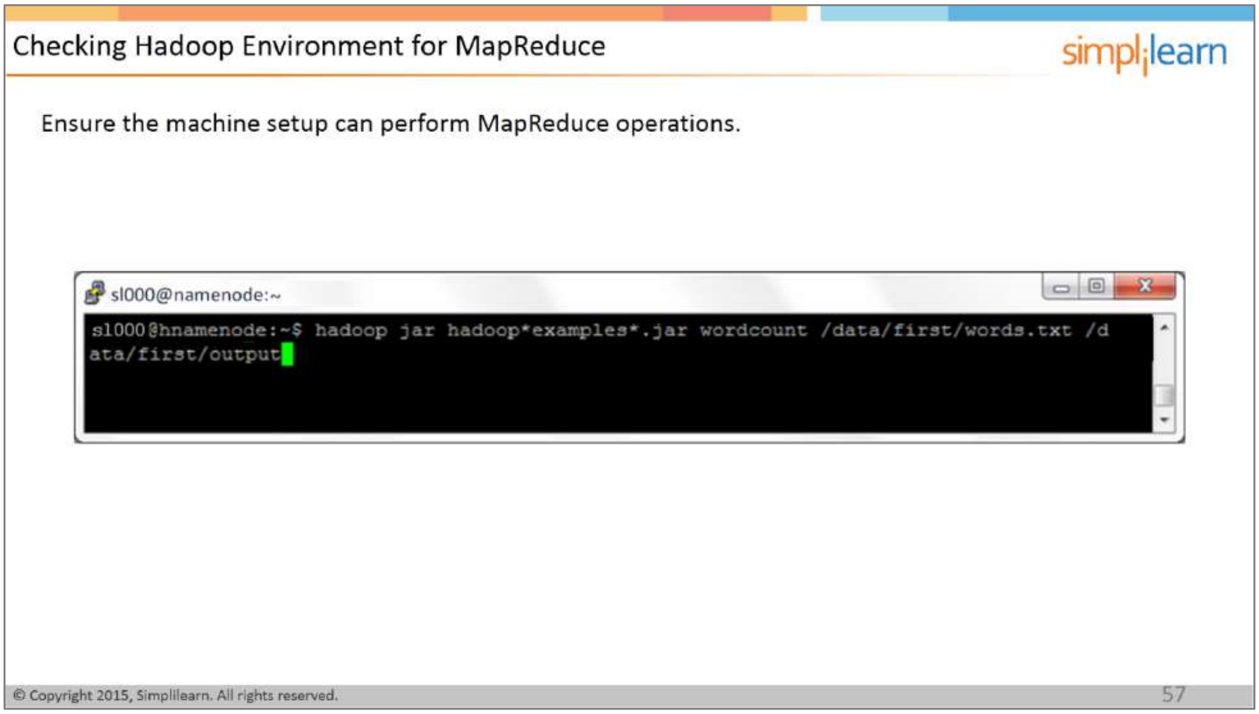
Create a New Project: Step 4

Include jar files from the Hadoop framework to ensure the programs locate the dependencies to one location.



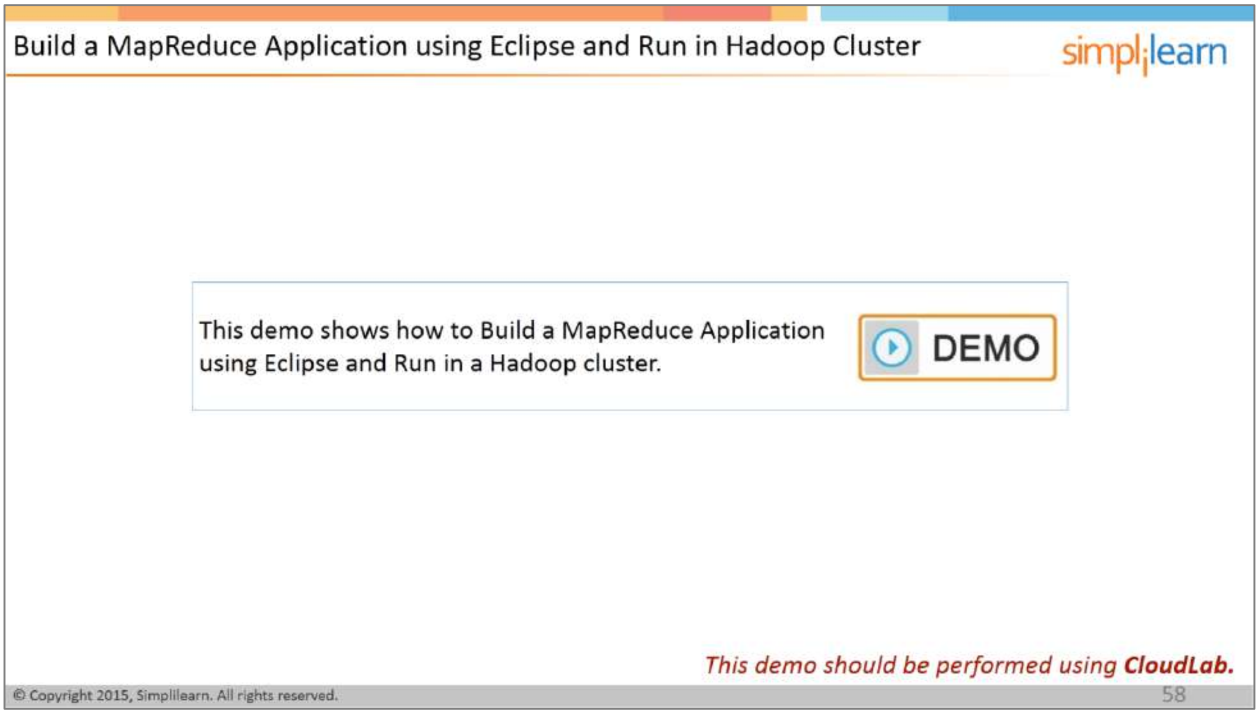
Create a New Project: Step 5

Under the **Libraries** tab, click the **Add External JARs...** button to add the essential jar files. After adding the jar files, click the **Finish** button to create the project successfully.



Checking Hadoop Environment for MapReduce

It is important to check whether the machine setup can perform MapReduce operations. To verify this, use the example jar files deployed by Hadoop. This can be done by running the command shown. Before executing this command, ensure that the words.txt file resides in the /data/first location.



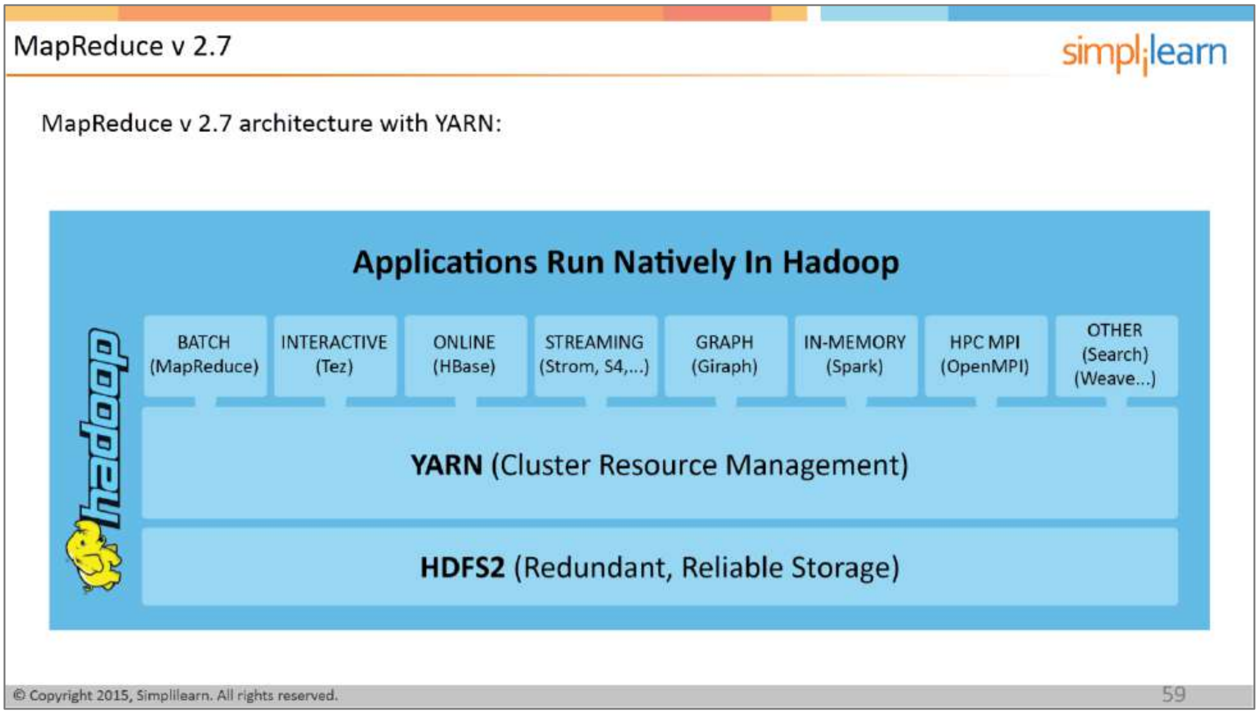
Build a MapReduce Program in Eclipse and Run in Hadoop Cluster

1. Let’s build a MapReduce Java program in Eclipse and then run in our Hadoop cluster. In this demo, we will run Eclipse in the Windows development machine and our Hadoop cluster will be in Ubuntu. First, let’s launch Eclipse.
2. Enter the workspace location.
3. Click OK.
4. The Eclipse window will open.
5. Close the welcome screen of Eclipse.
6. Select the New menu item.
7. Select Java Project.
8. The New Java Project window opens.
9. We will be build a WordCount program here to count the number of times each word occurs in a particular file.
10. Enter the name of the project as ‘WordCount’ and click Finish.
11. Right click the WordCount project in the panel on the left.
12. Select New and then Class.
13. The New Java Class window opens.
14. Enter the name of the class as ‘WordCount’.
15. Click Finish.

16. Now, let’s copy the WordCount program from the MapReduce tutorial on Hadoop’s website. You may go to Hadoop’s documentation or directly go to the link being shown.
17. Copy the source code for the Word Count program.
18. You would notice a lot of compilation errors. Let’s fix the build patch now. Select the project WordCount.
19. Select the Project menu item.
20. Click Properties.
21. In libraries, add external JARs. Browse to the unpacked Hadoop directory and go to share- Hadoop-MapReduce directory.
22. Select the Hadoop MapReduce client core and Hadoop MapReduce client common JAR files.
23.
24. Now, go to share-Hadoop-common directory.
25. Select the Hadoop common JAR file.
26. The compilation errors would have gone by now.

27. Let’s now see various portions of this program.
28. The usual Java imports are at the top of the program.
29. Further, there are Hadoop and MapReduce related import statements. Select the Description column header.
30. In the main method, we begin by setting configuration of the MapReduce job.
31. We set the name of the Mapper class.
32. We set the name of Combiner class.
33. Similarly, there is a Reducer class.
34. We can set the output key class.
35. We can also set the output value class.
36. Also, set the input data path for the source dataset.
37. Set the output path to a location where the results are desired.
38. Our Mapper class extends Mapper.
39. It has a map method which takes key and value as arguments and uses context.
40. In the WordCount logic, we just tokenize each line by space character and extract individual words.
41. Our Reducer class similarly extends Reducer.
42. The Reduce method takes a key and an iterable list of values as arguments.
43. The final output is again written as key value pairs.
44. Select the New menu item.
45. Let’s now build and export a JAR file to run this program on a Hadoop cluster. Click File menu and then Export.
46. The Export window opens.
47. Expand Java.
48. Select JAR file.
49. Click the Next button.
50. Enter the path and name of JAR. In this case, let’s name it ‘WordCount.jar’.

51. Make sure that you to select the project.
52. Now, let's transfer this JAR to the Hadoop cluster. If you are using Windows, you can use any SCP or FTP client such as WinSCP. Login to WinSCP using the IP address of the Hadoop Ubuntu cluster.
53. Enter the username of the Hadoop machine.
54. Enter the password.
55. Select the WordCount.jar file from the local Windows machine.
56. Using WinSCP, you can drag and drop to the Ubuntu machine in the panel on the right.
57. The Copy window opens.
58. Click the Copy button.
59. Now, run the WordCount program in the Hadoop cluster using the hadoop jar command. Specify the input file name on which WordCount is to be applied and also the output result path.
60. View the results in the output directory.
61. You will notice a file named similar to the part r1000.
62. View the contents of this output file using the hadoop fs -cat command.
63. The output will have a count of each word's occurrence in the input dataset.



MapReduce v 2.0

This image shows the MapReduce v2.0 architecture comprising YARN.

Spot the Error

The Hadoop services of LUZ Inc. suddenly stop due to which there are operation failures. Debby is assigned the task of checking which Hadoop services are running but she is unable to get the output. Help her identify the error.

Spot the incorrect code.

```
$ Jps
3776 NodeManager
2778 NameNode
3508 ResourceManager
4356 Jps
3378 SecondaryNameNode
$ Jps ResourceManager
```


Spot the Error

The incorrect code is: `$ Jps`

The correct code is: `$ jps`

QUIZ 1

Which one is an example of MapReduce?

a. A mother feeding her kids one by one

b. People standing in a queue outside ATM

c. People going out to collect donations

d. People running in a relay race

QUIZ 2

Who executes the map task?

a. ResourceManager

b. ApplicationManager

c. ApplicationMaster

d. NodeManager

QUIZ 3

What is the sequence of a MapReduce Job?

a. Map, input split, reduce

b. Input split, map, reduce

c. Map and then reduce

d. Map, split, map

simplilearn

QUIZ 4

Which one is the responsibility of the MapReduce developer?

a. Map, reduce

b. Reduce, input split

c. Input split, map

d. Sort, shuffle

QUIZ 5

Which command is used to upload files in the Hadoop environment?

a. Hadoop fs

b. Hadoop fs -ls

c. Hadoop fs -copyFromLocal source destination

d. Hadoop fs -copyToLocal source destination

QUIZ 6

Which command is used to ensure all of the Hadoop daemons are running in the Hadoop environment?

a. Hadoop fs

b. Hadoop fs -ls

c. jps

d. Hadoop fs -copyToLocal source destination

QUIZ 7

Your client application submits a MapReduce job to your Hadoop cluster. Identify the Hadoop daemon that works with the NodeManager to execute and monitor the granted resources (bundled as containers) for a given application.

a. ResourceManager

b. Container

c. ApplicationMaster

d. ApplicationManager

QUIZ 8

What data does a reduce method process?

a. All the data in a single input file

b. All data produced by a single mapper

c. All data for a given key, regardless of which mapper(s) produced it

d. All data for a given value, regardless of which mapper(s) produced it

QUIZ 9

Assuming default settings, which best describes the order of data provided to a reducer's reduce method?

a. The keys given to a reducer aren't in a predictable order, but the values associated with those keys always are.

b. Both the keys and the values passed to a reducer always appear in sorted order.

c. Neither keys nor values are in any predictable order.

d. The keys given to a reducer are in sorted order, but the values associated with each key are in no predictable order.

 80

QUIZ 10

If you run the word count MapReduce program with m mappers and r reducers, how many output files will you get at the end of the job? How many key-value pairs will there be in each file? Assume k is the number of unique words in the input files.

a. There will be r files, each with exactly k/r key-value pairs

b. There will be r files, each with approximately k/m key-value pairs

c. There will be r files, each with approximately k/r key-value pairs

d. There will be m files, each with exactly k/m key-value pairs

 82

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which one is an example of MapReduce?	c. People going out to collect donations can do jobs in parallel, as in the map phase, and they can put the collection in one place, as in the reduce phase.
2	Who executes the map task?	d. The NodeManager executes map and reduce task as part of container execution.
3	What is the sequence of MapReduce Job?	b. For MapReducing, the load on multiple nodes for parallel processing needs to be distributed, so the first process must be input split, followed by the map and reduce processes.
4	Which one is the responsibility of the MapReduce developer?	a. The map and reduce processes are ones that can be done in parallel. The developer has the knowledge regarding these processes. Thus, **a** is the correct answer.
5	Which command is used to upload files in the Hadoop environment?	c. The Hadoop fs -copyFromLocal source destination command is used to perform uploading of files in the Hadoop environment.
6	Which command is used to ensure all of the Hadoop daemons are running in the Hadoop environment?	c. Jps is used to identify whether the Hadoop daemons are running.
7	Your client application submits a MapReduce job to your Hadoop cluster. Identify the Hadoop daemon that works with the NodeManager to execute and monitor the granted resources (bundled as containers) for a given application.	c. **:** Application Master works with the NodeManager to execute and monitor the granted resources (bundled as containers) for a given application.
8	What data does a reduce method process?	c. The reducer reduces all data for a given key irrespective of the type of mapper used.
9	Assuming default settings, which best describes the order of data provided to a reducer's reduce method?	d. The keys given to a reducer are in sorted order, but the values associated with each key are in no predictable order.
10	If you run the word count MapReduce program with m mappers and r reducers, how many output files will you get at the end of the job? And how many key-value	a. There will be r files, each with exactly k/r key-value pairs if default settings are applied.

	pairs will there be in each file? Assume k is the number of unique words in the input files.	

Case Study

Scenario | Analysis | Solution

XY Invest provides investment advice to high net worth individuals and maintains stock market data of various exchanges. It handles stock market data critical for analysis and monitoring. This entails time-intensive processing of huge amounts of data, and vertical scaling is proving to be expensive. In its effort to find an alternate solution for processing, XY identifies that using Hadoop would reduce the company's cost and effort significantly. It wants to use MapReduce, the processing component of Hadoop.

Case Study

Scenario | Analysis | Solution

The company does research on Hadoop and finds that it is a popular solution for Big Data storage and processing. Hadoop has two components, HDFS for storage and MapReduce for processing.

Advantages of using MapReduce:

1. It distributes processing so jobs can be completed really quickly.
2. It is highly fault tolerant, so even if some tasks fail, they will be automatically retried.
3. It provides a Java interface for writing MapReduce jobs.
4. It provides a powerful map and reduce paradigm where map does the filtering of data and reduce does the aggregation of data.

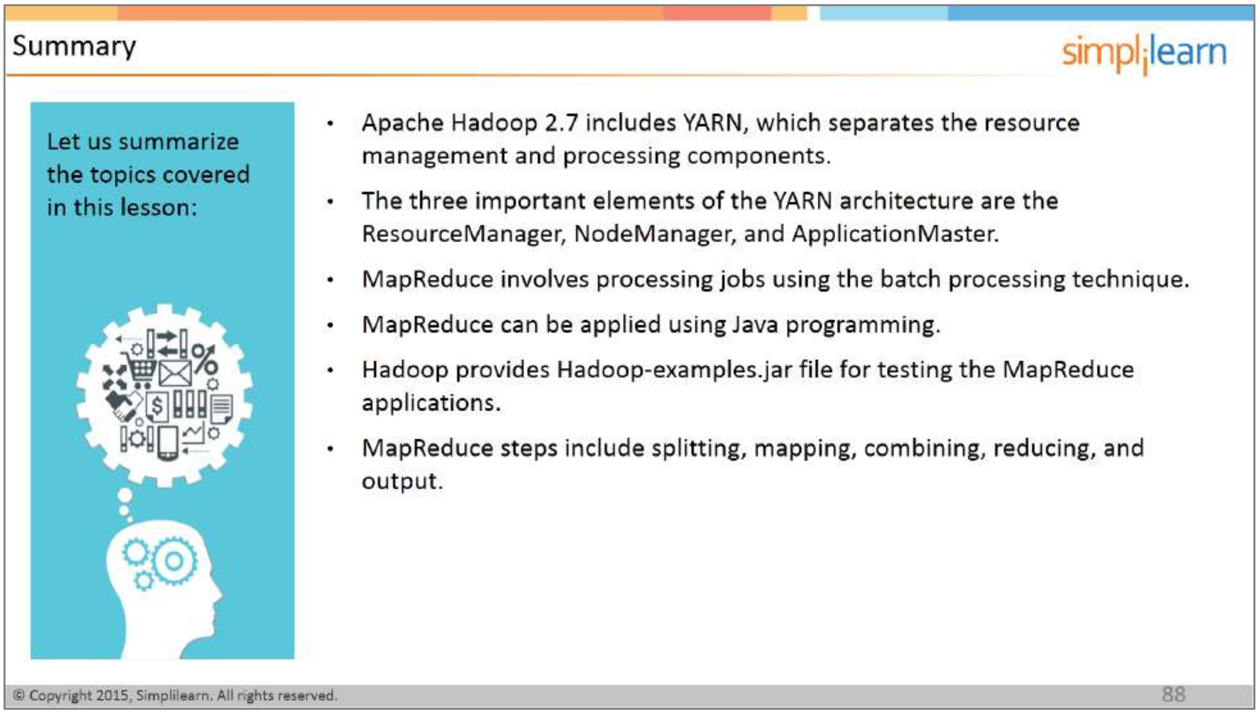
Case Study

Scenario | Analysis | Solution

Perform the following steps to explore the features of MapReduce paradigm:

1. Store log data from Hadoop in HDFS.
2. Write a Java program per MapReduce specifications to process the log files. Map does the horizontal and vertical filtering of data; Reduce does the aggregation of data.
3. Create the *mapreduce.jar* file by compiling the Java program.
4. Run the program using YARN by providing input and output parameters.
5. Verify the program and output.

 87



Summary

Let us summarize the topics covered in this lesson:

Apache Hadoop 2.0 includes YARN, which separates the resource management and processing components.

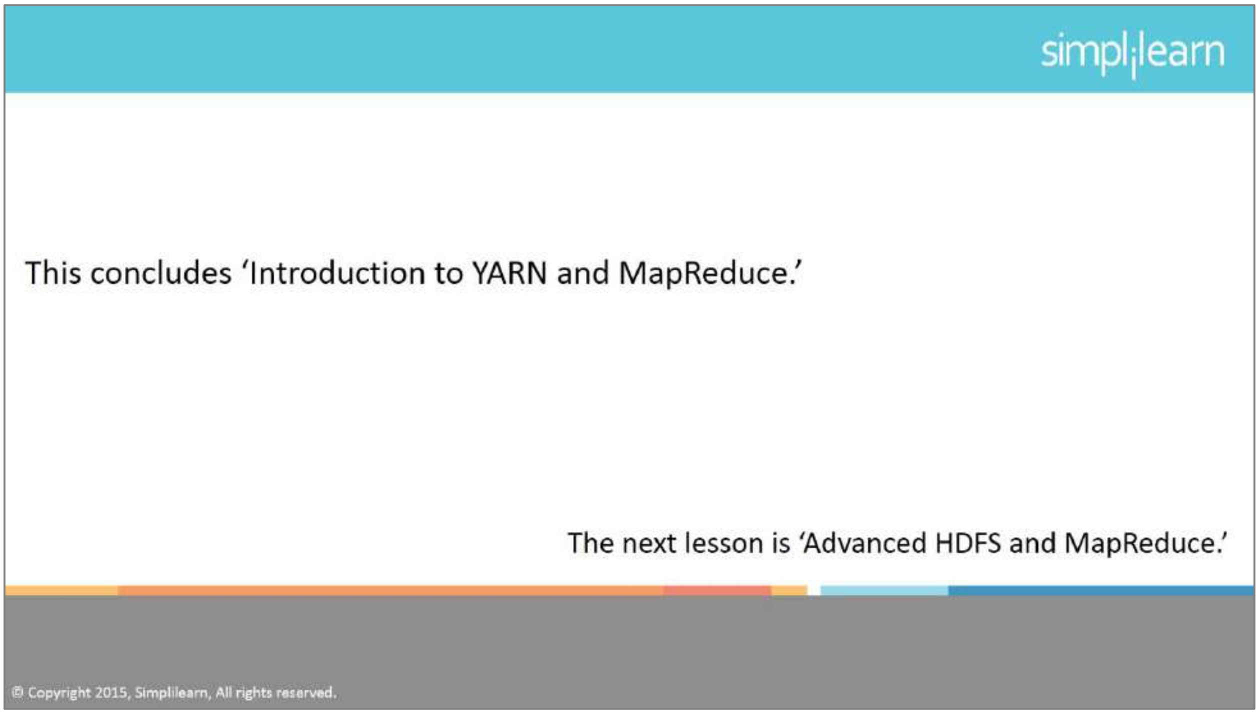
The three important elements of the YARN architecture are the ResourceManager, NodeManager, and ApplicationMaster.

MapReduce involves processing jobs using the batch processing technique.

MapReduce can be done using Java programming.

Hadoop provides with hadoop-examples.jar file which is normally used by administrators and programmers to perform testing of the MapReduce applications.

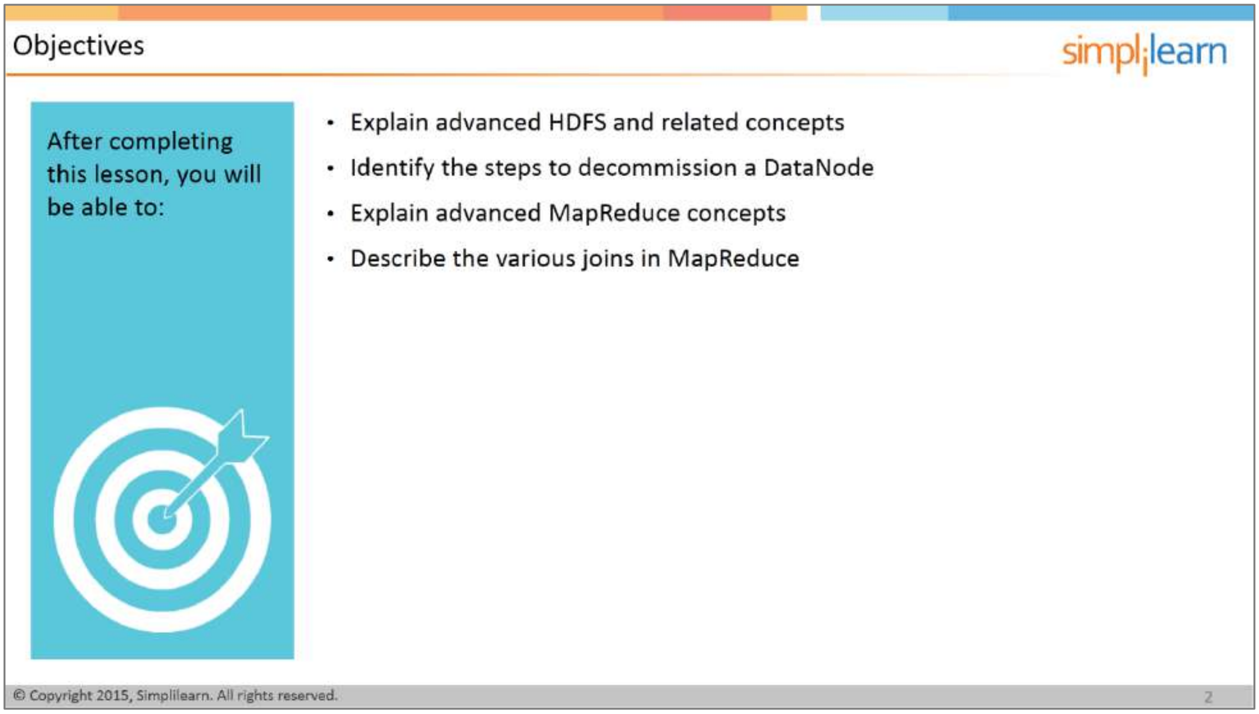
MapReduce contains steps such as splitting, mapping, combining, reducing, and output.



Conclusion

This concludes 'Introduction to YARN and MapReduce.' In the next lesson, we will focus on 'Advanced HDFS and MapReduce.'

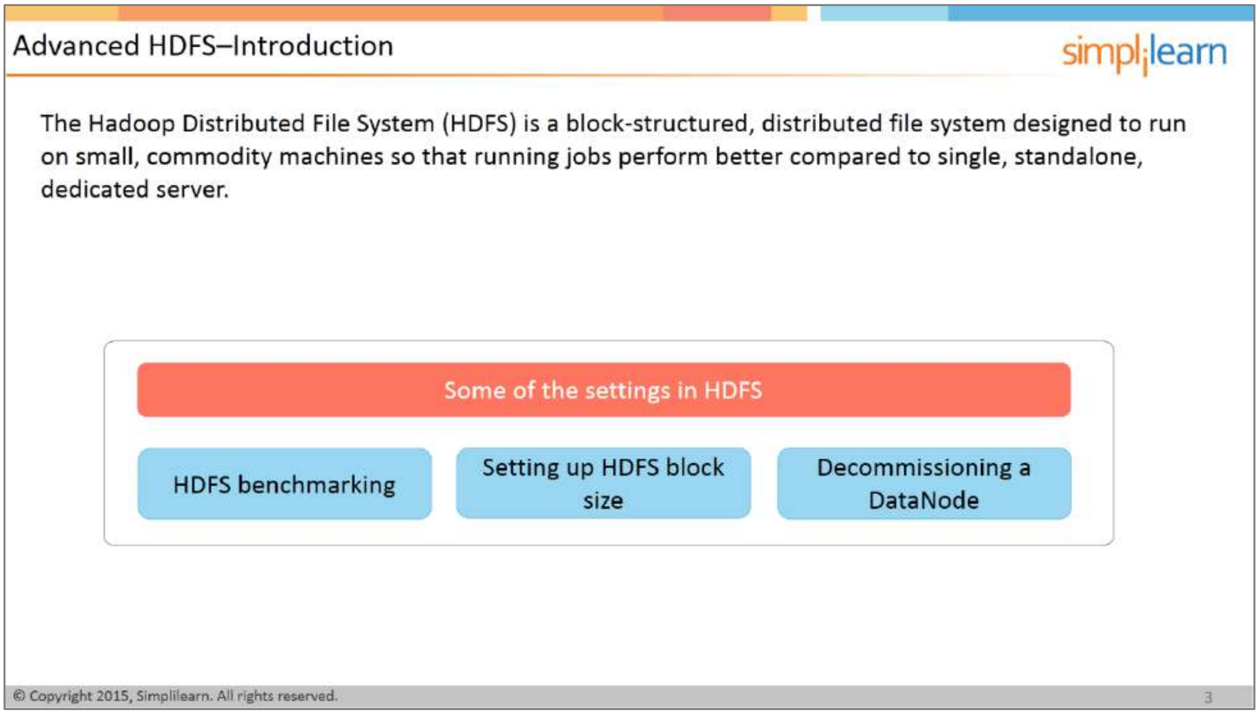
Lesson 5—Advanced HDFS and MapReduce



Objectives

After completing this lesson, you will be able to:

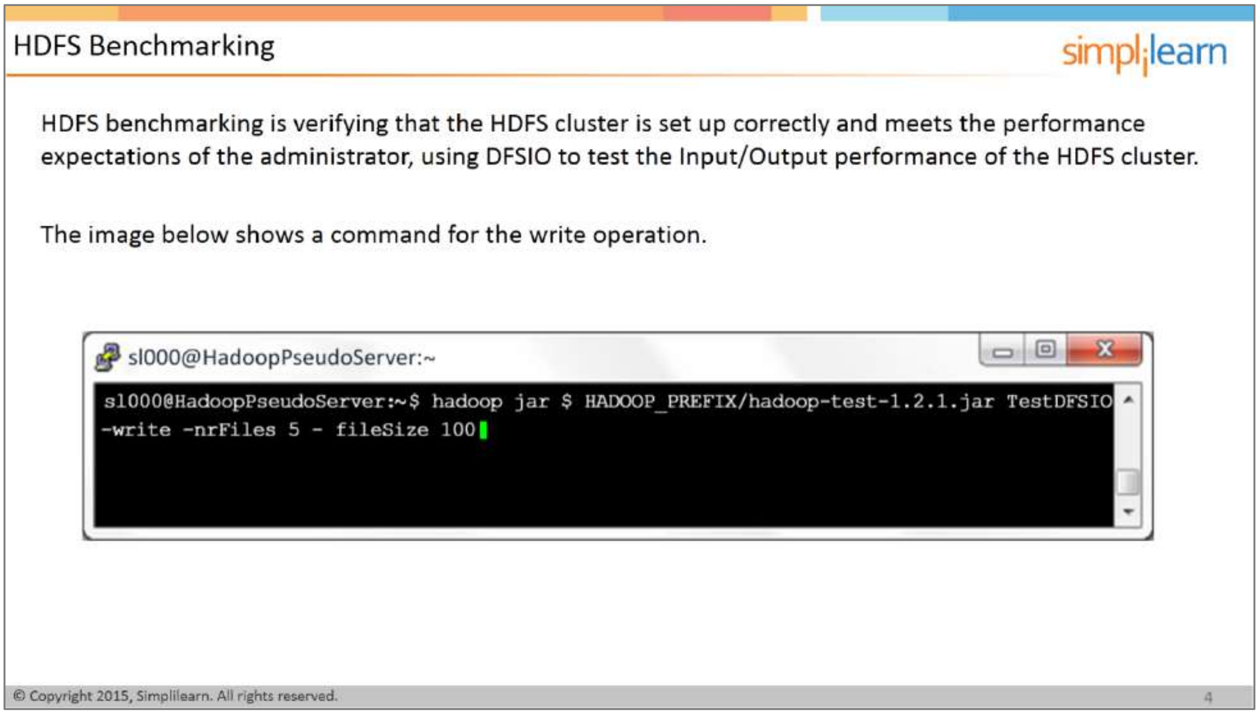
- Explain advanced HDFS and related concepts;
- Identify the steps to decommission a DataNode;
- Explain advanced MapReduce concepts; and
- Describe the various joins in MapReduce.



Advanced HDFS—Introduction

The Hadoop Distributed File System or HDFS is a block-structured, distributed file system designed to run on small, commodity machines so that running jobs perform better compared to single, standalone, dedicated server. HDFS answers the storage needs of Big Data, and makes data accessible to Hadoop services.

Some of the settings in advanced HDFS, discussed in this lesson, are HDFS benchmarking, setting up HDFS block size, and decommissioning or removing a DataNode.



HDFS Benchmarking

HDFS benchmarking is verifying that the HDFS cluster is set up correctly and meets the performance expectations of the administrator. You will use DFSIO to test the Input/Output performance of the HDFS cluster.

The image shown displays the command to be used for the write operation.

The image shown displays the command for the read operation. This benchmark uses the file written by the write command executed earlier.

HDFS Benchmarking (contd.)

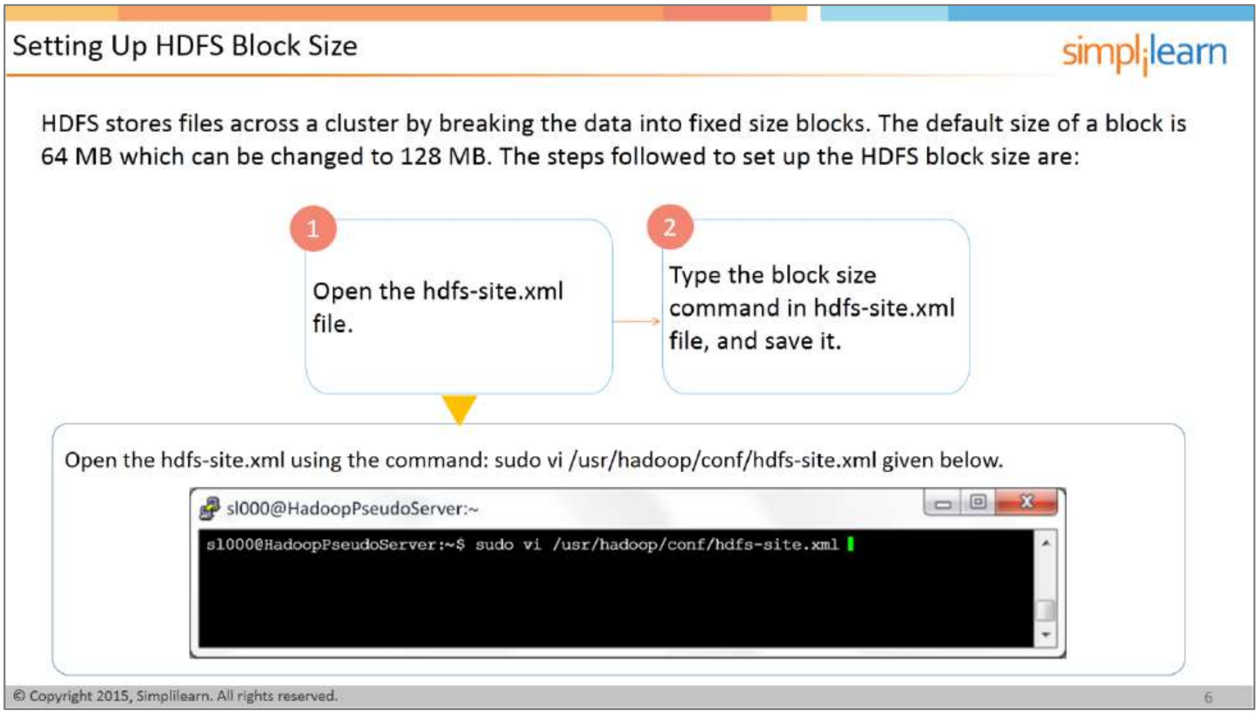
The image below shows a command for the read operation. This benchmark uses the file written by the write command executed earlier.

sl000@HadoopPseudoServer:~

```
sl000@HadoopPseudoServer:~$ hadoop jar $ HADOOP_PREFIX/hadoop-test-1.2.1.jar TestDFSIO
-read -nrFiles 5 - fileSize 100
```

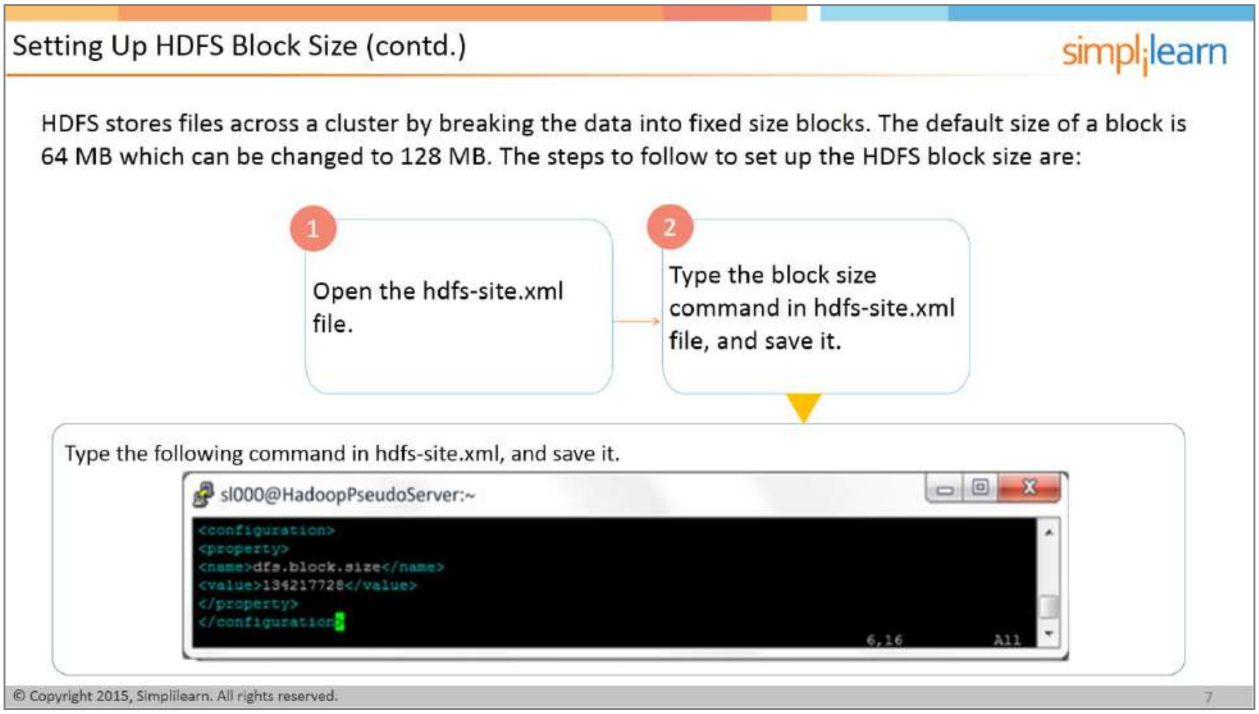

HDFS Benchmarking

HDFS stores files across a cluster by breaking the data into fixed size blocks. The default size of a block is 64 MB. However, a Hadoop administrator can change the size of the block, in this case, to 128 MB.



Setting Up HDFS Block Size

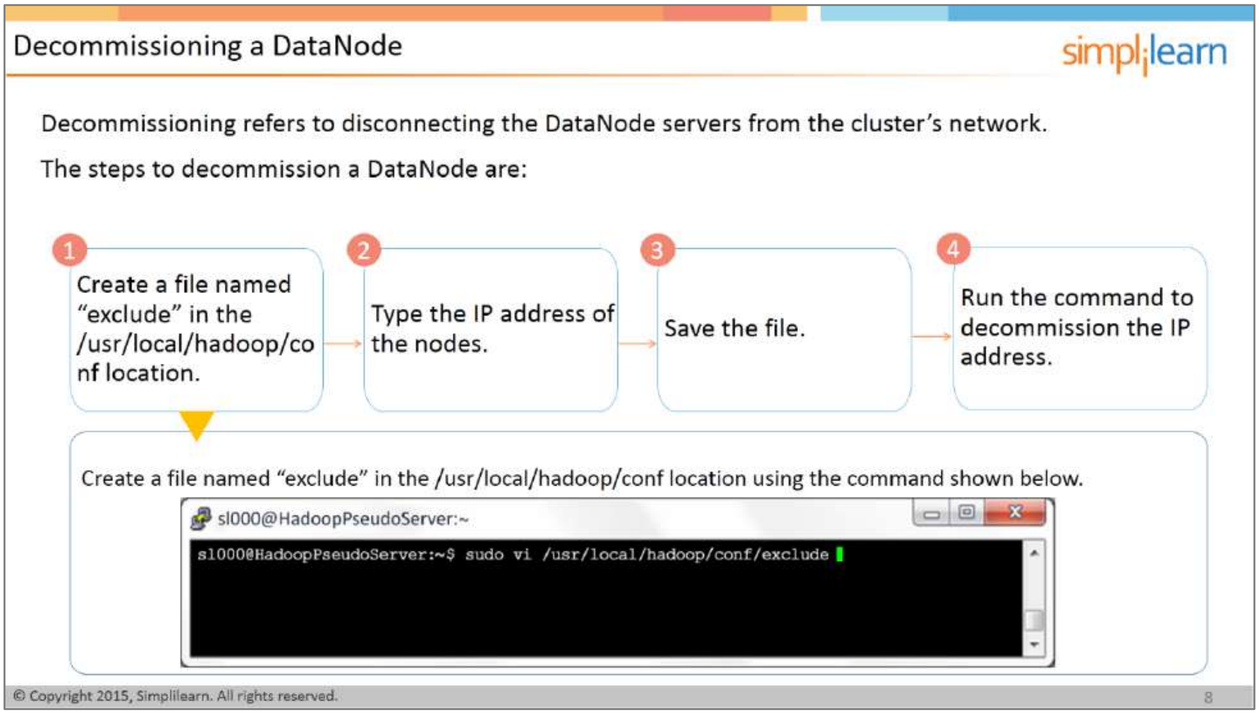
To set up the HDFS block size, open the xml file as shown. The command used to open the xml file is shown here.



Setting Up HDFS Block Size

Next, type the block size command and save the file.

The command specified in the image ensures that the block size of any file uploaded in HDFS from now on will be 128 MB. The parameter dfs.block.size is responsible for setting the size of the block, which the NameNode server has to create. The block size is specified in bytes, 134217728 in this case.

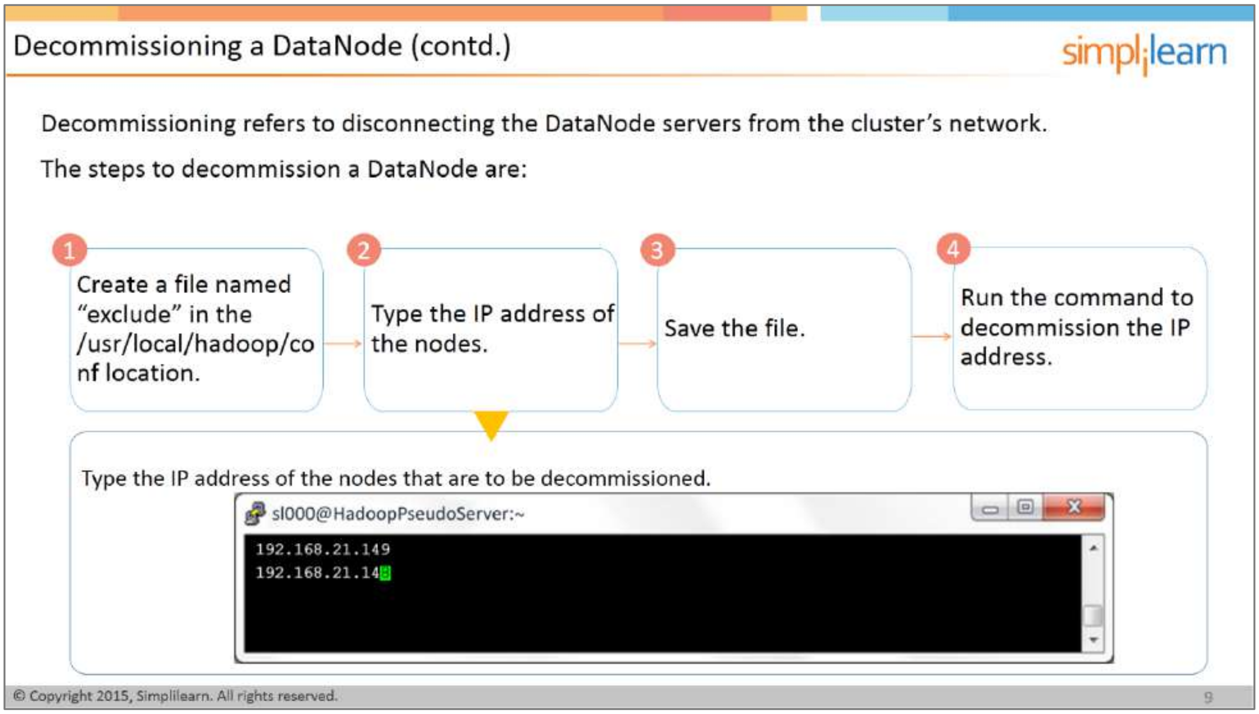


Decommissioning a DataNode

Decommissioning refers to disconnecting DataNode servers from the cluster's network. There may be multiple situations, such as a hardware upgrade or failure, when you want to remove one or more nodes from an HDFS cluster. This can be done in four steps.

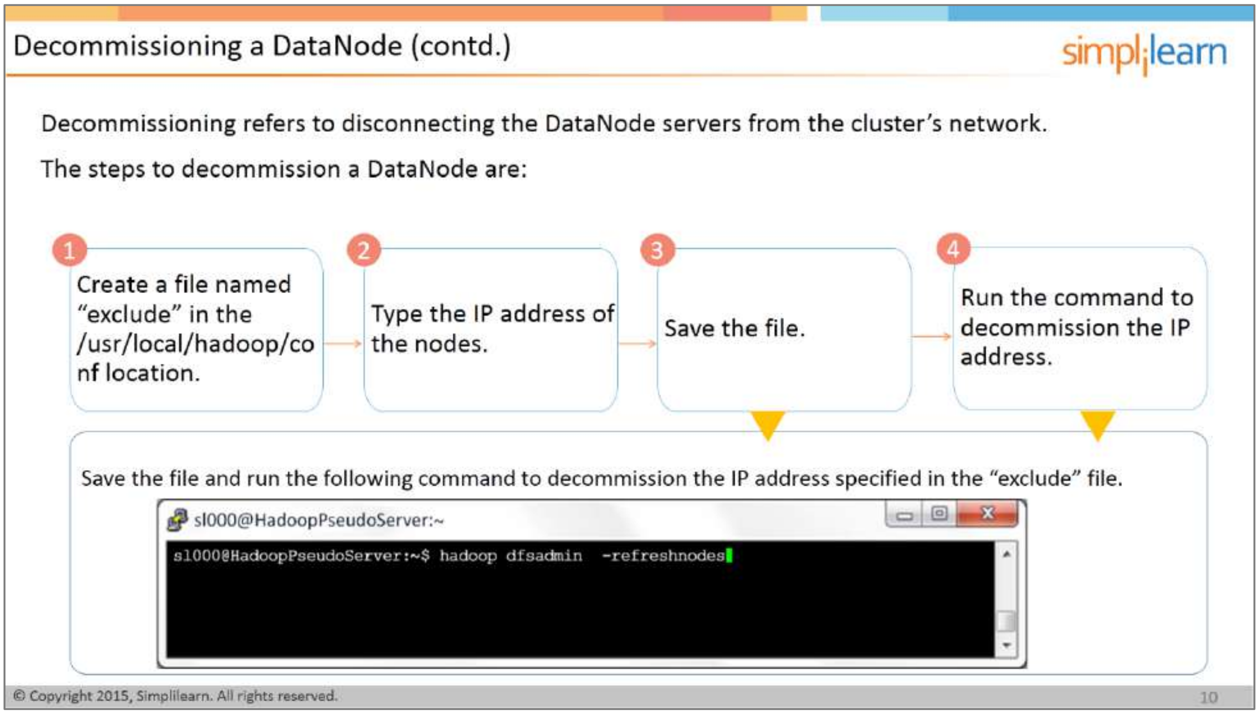
First, create a file named "exclude" in the location mentioned here. Second, type the IP address of the node. Next, save the file. Finally, run the command to decommission the IP address of the node.

Use the command shown to create a file named "exclude" in the specified location.



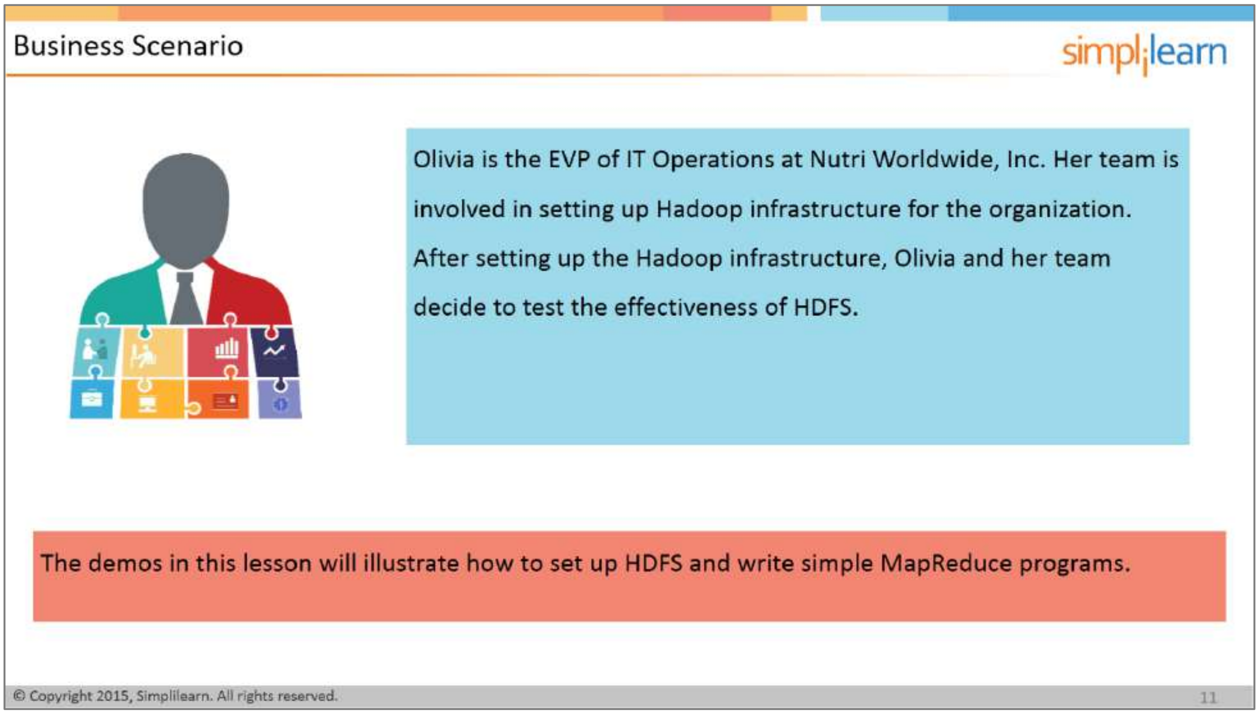
Decommissioning a DataNode

Type the IP address of the node or nodes to be decommissioned as shown.



Decommissioning a DataNode

Save the file, and run the command displayed here to decommission the IP address specified in the file "exclude."



Business Scenario

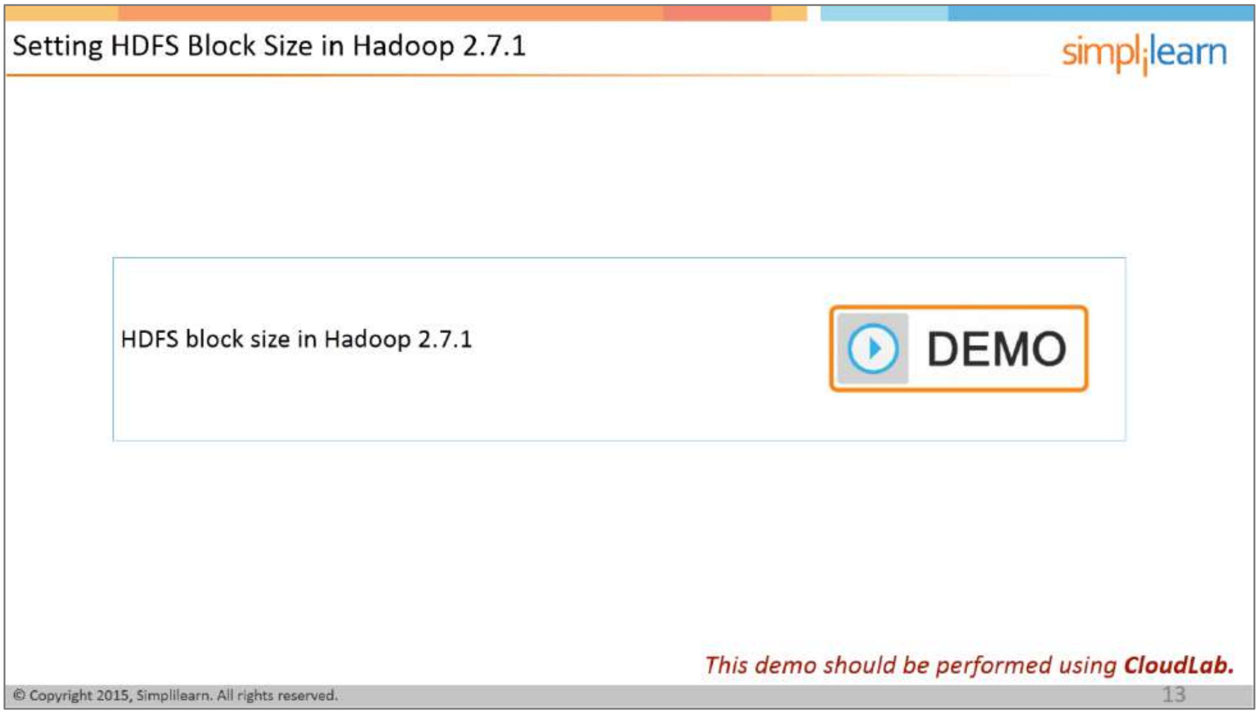
Olivia is the EVP of IT Operations at Nutri Worldwide, Inc. Her team is involved in setting up Hadoop infrastructure for the organization. After setting up the Hadoop infrastructure, Olivia and her team decide to test the effectiveness of HDFS.

HDFS

You will see a demo on HDFS.

1. Start Hadoop services using the command start-all.sh.
2. Ensure hadoop services are live and running.
3. Let us perform benchmarking of your HDFS. We will initially perform the write operation.
4. The result shows the execution time to write files and IO rate and throughput.
5. Let us perform the read operation of the file..
6. The command will be executed. The result will show the output time. This is how benchmarking is done. In the next step we will see how to manually set the block size.
7. We can set the block size manually by editing hdfs-site.xml. We need to set the parameter dfs.block.size as 134217728, that is, 128MB.
8. We need to refresh HDFS. To refresh HDFS, we need to stop and then start the hadoop services. To stop the services, the command is stop-all.sh.
9. Let us upload a dataset for testing the same.
10. Let us see the block size. Access the GUI. The link to access the GUI is http://192.168.21.184:50070.
11. Click datanew.
12. Next, click big.
13. Observe the block size. In the next step, we will perform the decommissioning of DataNodes.
14. We need to create an exclude file. Enter the IP address of the DataNode and save the file.

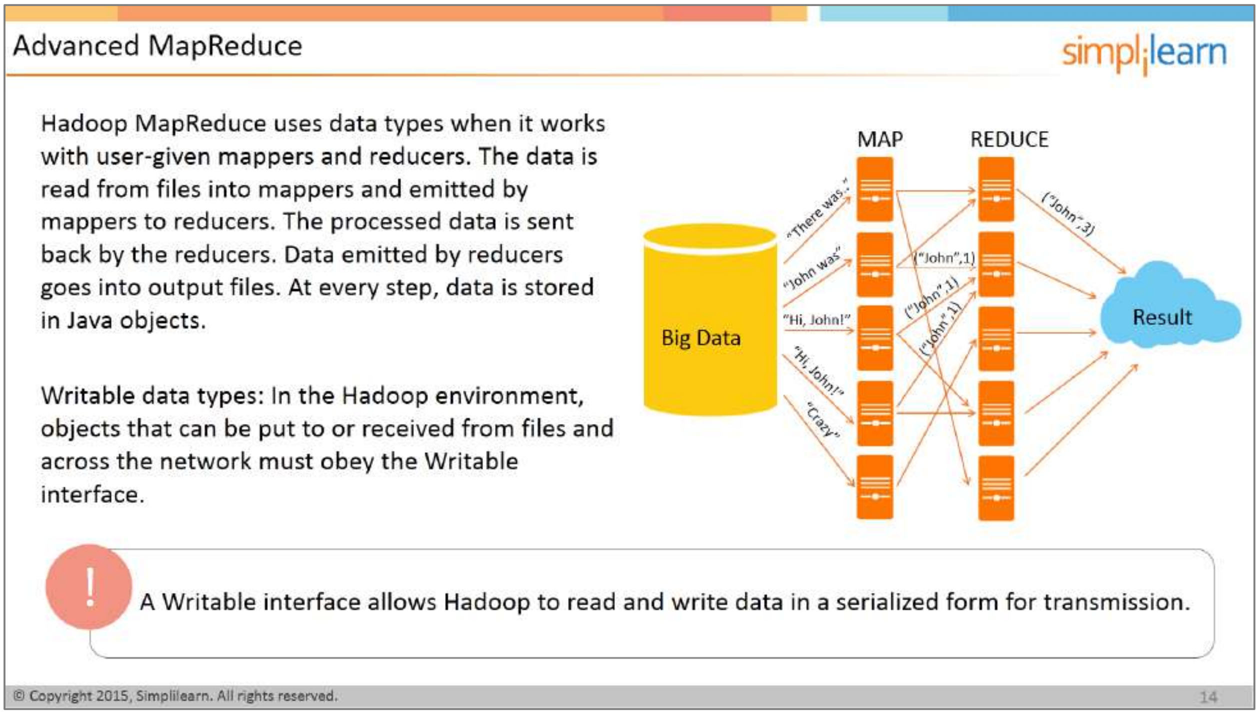
15. Let us refresh the cluster. Thus we have successfully executed the command.



Setting HDFS Block Size in Hadoop 2.7.1

You will see a demo on setting HDFS block size in Hadoop 2.7.1.

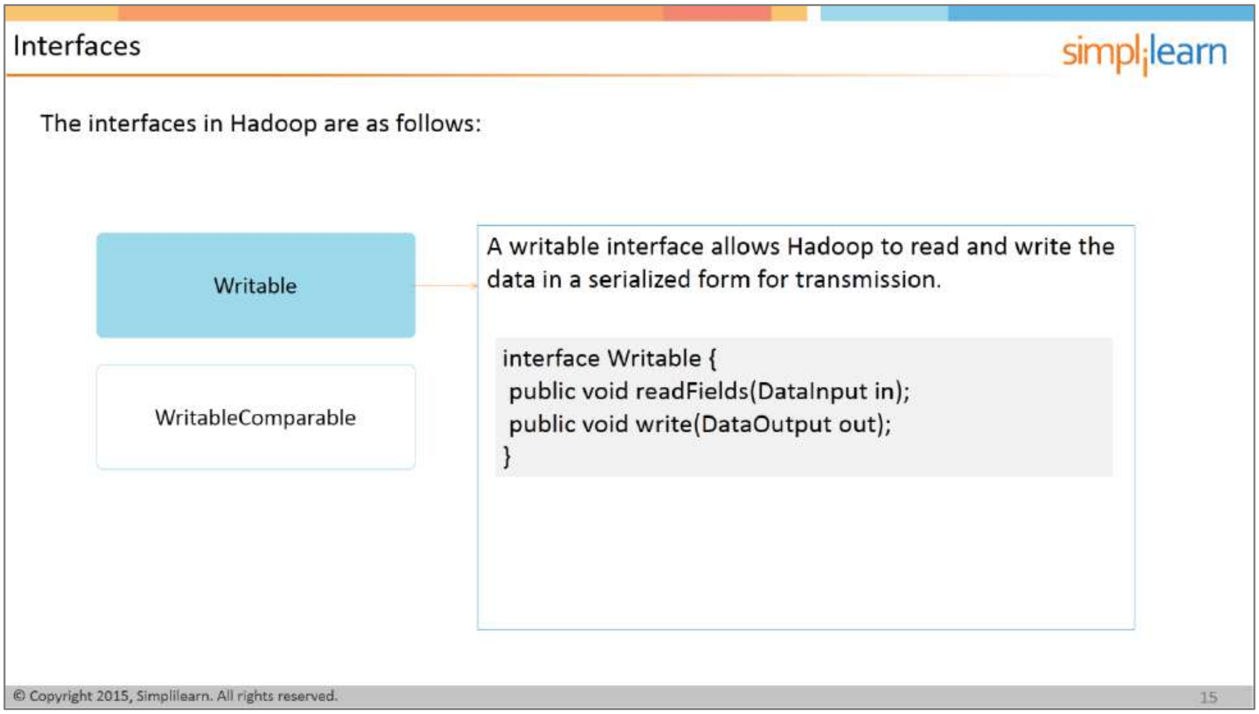
1. Download an e-book into your machine from the URL shown.
2. Let's copy into HDFS using the hadoop fs -copyFromLocal command.
3. Set the dfs block size to 30 bytes while copying the file.
4. You will get an error. Now let's try to copy into HDFS by setting the block size to 64 bytes while copying the file.
5. You will again get an error. Now let's copy into HDFS using a block size of 1048576 bytes while copying the file.
6. This time the copying is successful, as 1048576 is a multiple of 64, which is the requirement for HDFS files block size. Verify the file using the command given.
7. Check further file statistics using hadoop fsck filename command.
8. We notice that the file is around 1573151 bytes and is copied into two blocks, as the size of each block is 1048576 bytes.
9. Verify the block size by running the command displayed.
10. Notice the block size matches what was set, that is 1048576.



Advanced MapReduce

Hadoop MapReduce uses data types to work with user-given mappers and reducers. The data is read from files into mappers, and emitted by mappers to reducers. Processed data is sent back by the reducers. Data emitted by reducers goes into output files. At every step, data is stored in Java objects.

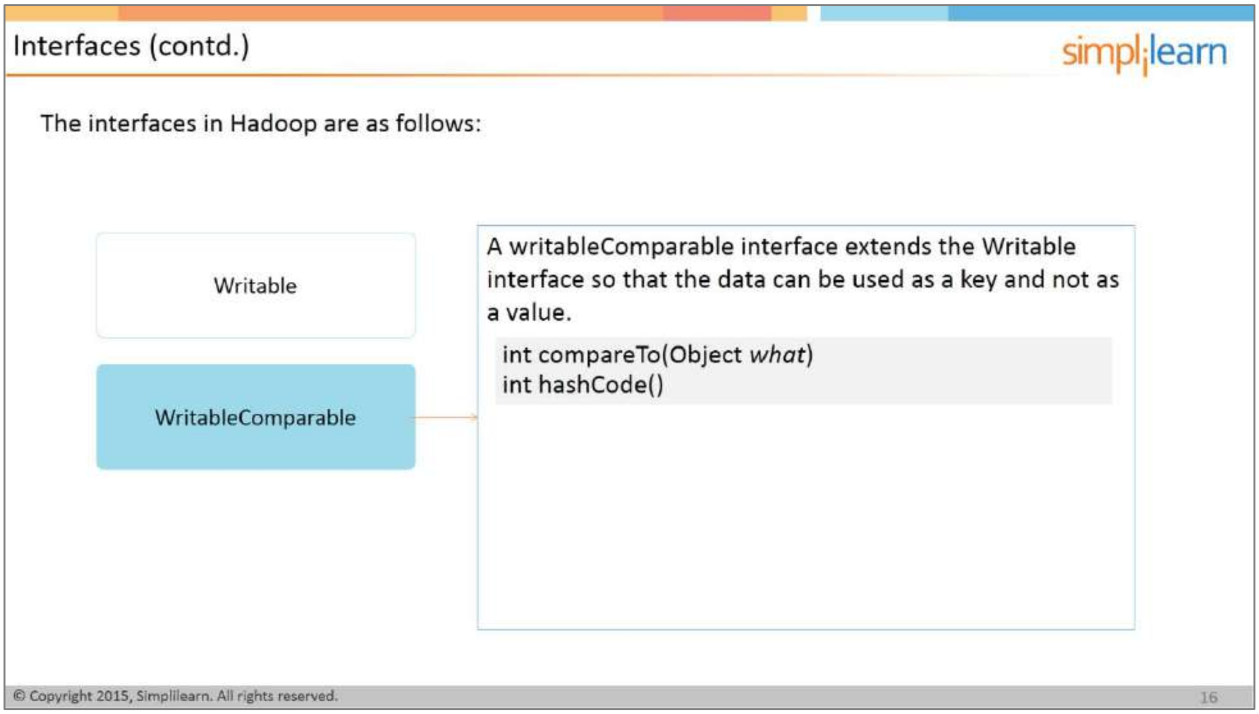
In the Hadoop environment, objects that can be put to or received from files and across the network must obey the Writable interface, which allows Hadoop to read and write data in a serialized form for transmission. Let's look at Hadoop interfaces in some more detail.



Interfaces

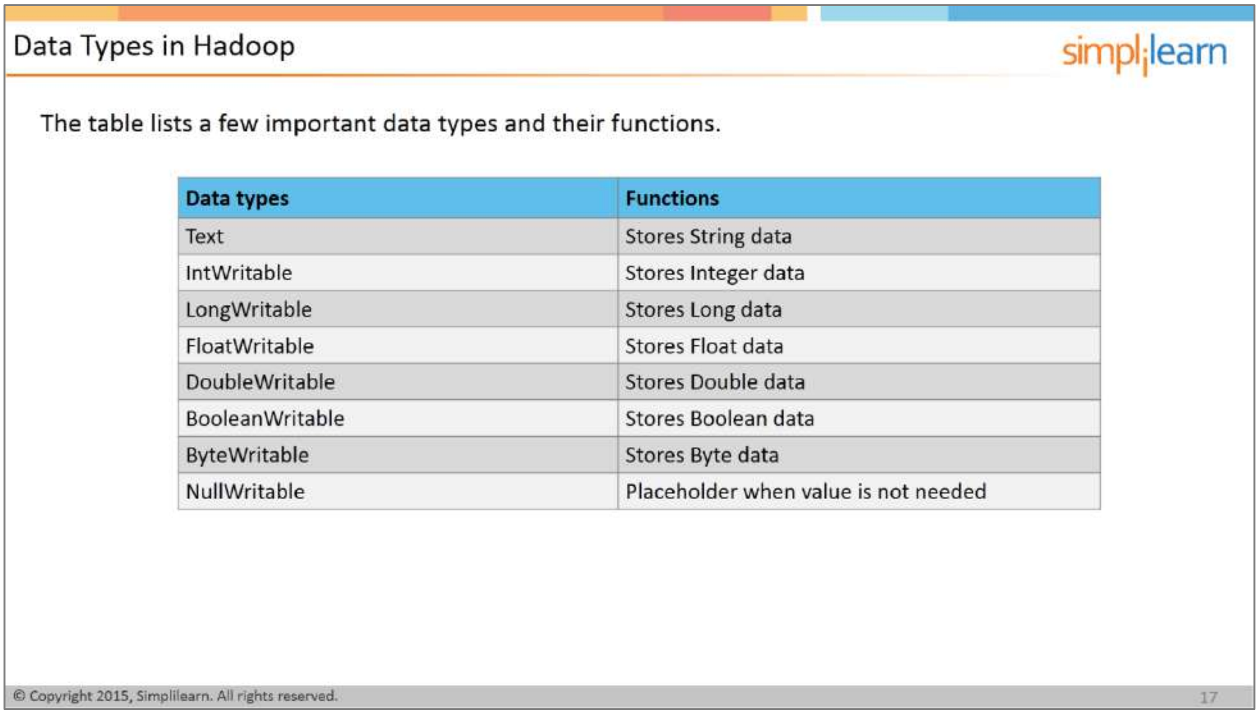
The interfaces in Hadoop are Writable and WritableComparable.

As you've already seen, a Writable interface allows Hadoop to read and write data in a serialized form for transmission. A Writable interface consists of two methods: read fields and write as shown here.



Interfaces (contd.)

A WritableComparable interface extends the Writable interface, so that the data can be used as a key and not as a value. As shown here, the WritableComparable implements two methods: compareTo and hashCode.

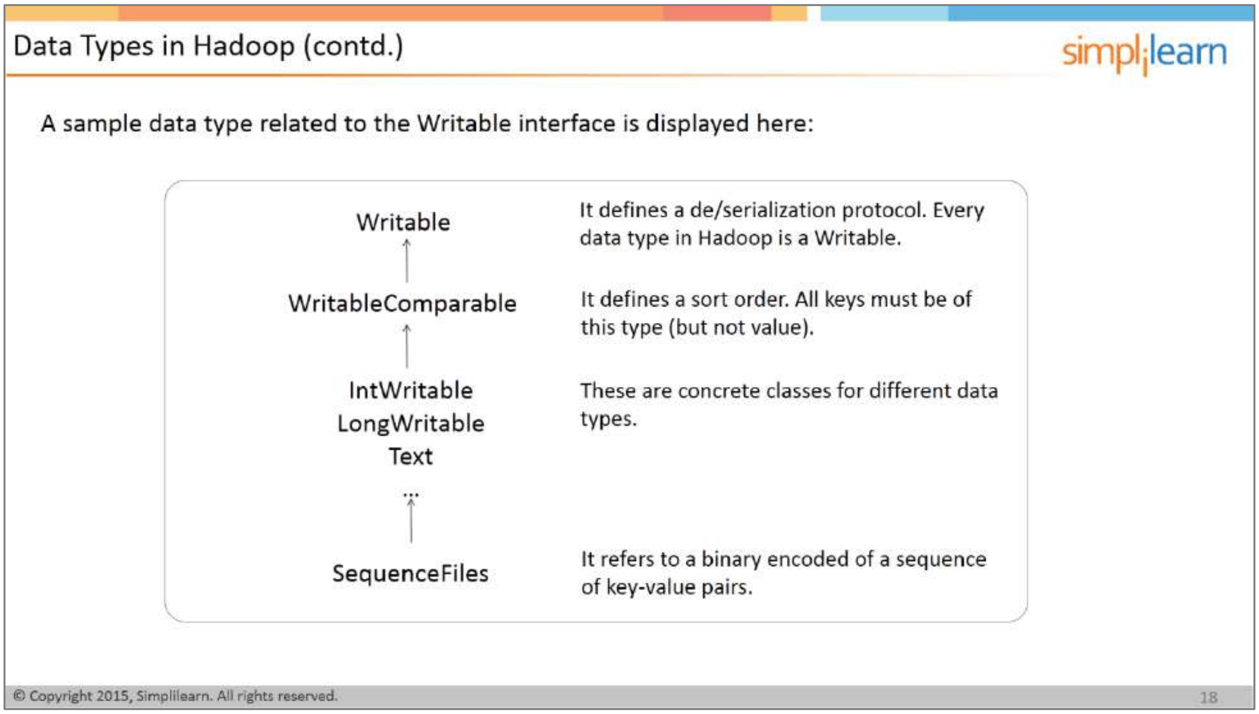


Data types	Functions
Text	Stores String data
IntWritable	Stores Integer data
LongWritable	Stores Long data
FloatWritable	Stores Float data
DoubleWritable	Stores Double data
BooleanWritable	Stores Boolean data
ByteWritable	Stores Byte data
NullWritable	Placeholder when value is not needed

Data Types in Hadoop

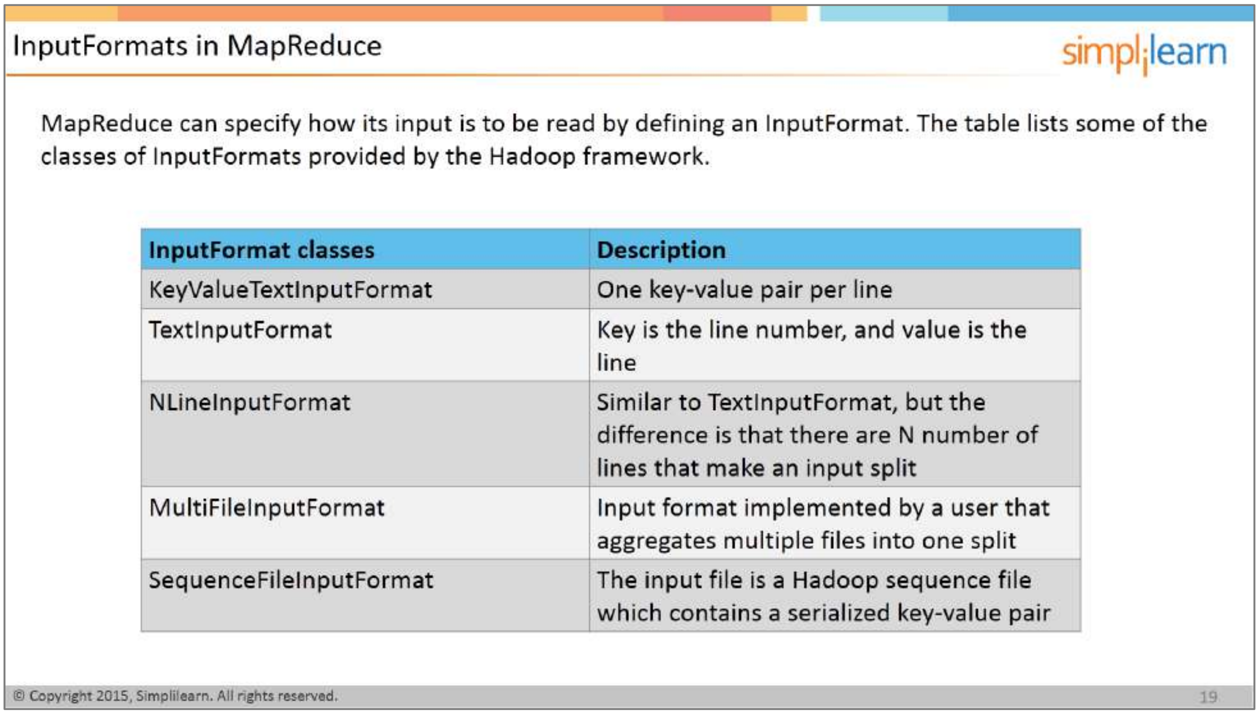
Let's now examine various data types in Hadoop and their functions.

The first data type is Text. The function of this data type is to store String data. The IntWritable data type stores integer data. LongWritable, as the name suggests, stores Long data. Similarly, other data types are FloatWritable for storing Float data, DoubleWritable for storing Double data. There is also BooleanWritable and ByteWritable data types. NullWritable is a placeholder when a value in not needed.



Data Types in Hadoop (contd.)

The illustration here shows a sample data type you can create on your own. This data type will need you to implement a Writable interface.


InputFormats in MapReduce

simplilearn

MapReduce can specify how its input is to be read by defining an InputFormat. The table lists some of the classes of InputFormats provided by the Hadoop framework.

InputFormat classes	Description
KeyValueTextInputFormat	One key-value pair per line
TextInputFormat	Key is the line number, and value is the line
NLineInputFormat	Similar to TextInputFormat, but the difference is that there are N number of lines that make an input split
MultiFileInputFormat	Input format implemented by a user that aggregates multiple files into one split
SequenceFileInputFormat	The input file is a Hadoop sequence file which contains a serialized key-value pair

19

InputFormats in MapReduce

MapReduce can specify how its input is to be read by defining an InputFormat. The table lists some of the classes of InputFormats provided by the Hadoop framework. Let's look at each of them.

- The first class is KeyValueTextInputFormat, which is used to create a single key-value pair per line.
- TextInputFormat is used to create a program considering a key as the line number and a value as the line itself.
- NLineInputFormat is similar to TextInputFormat except that there are N number of lines that make an input split.
- MultiFileInput Format is used to implement an input format that aggregates multiple files into one split.
- For the class SequenceFileInputFormat to be implemented, the input file must be a Hadoop sequence file which contains serialized key-value pairs.

OutputFormats in MapReduce

The table lists some of the key classes of OutputFormats provided by the Hadoop framework.

OutputFormat classes	Description
TextOutputFormat	It is the default OutputFormat, and writes records as lines of text. Each key-value pair is separated by a TAB character. This can be customized by using the mapred.textoutputformat.separator property. The corresponding InputFormat is KeyValueTextInputFormat.
SequenceFileOutputFormat	It writes sequence files to save the output. It is compact and compressed.
SequenceFileAsBinaryOutputFormat	It writes key and value in raw binary format into a sequential file container.
MapFileOutputFormat	It writes MapFiles as the output. The keys in a MapFile must be added in an order, and the reducer will emit keys in the sorted order.
MultipleTextOutputFormat	It writes data to multiple files whose names are derived from output keys and values.
MultipleSequenceFileOutputFormat	It creates output in multiple files in a compressed form.

OutputFormats in MapReduce

The classes for MapReduce OutputFormat is discussed here.

The first class is the default OutputFormat, TextOutputFormat. It writes records as lines of text. Each key-value pair is separated by a TAB character. This can be customized by using the mapred.textoutputformat.separator property. The corresponding input format is KeyValueTextInputFormat.

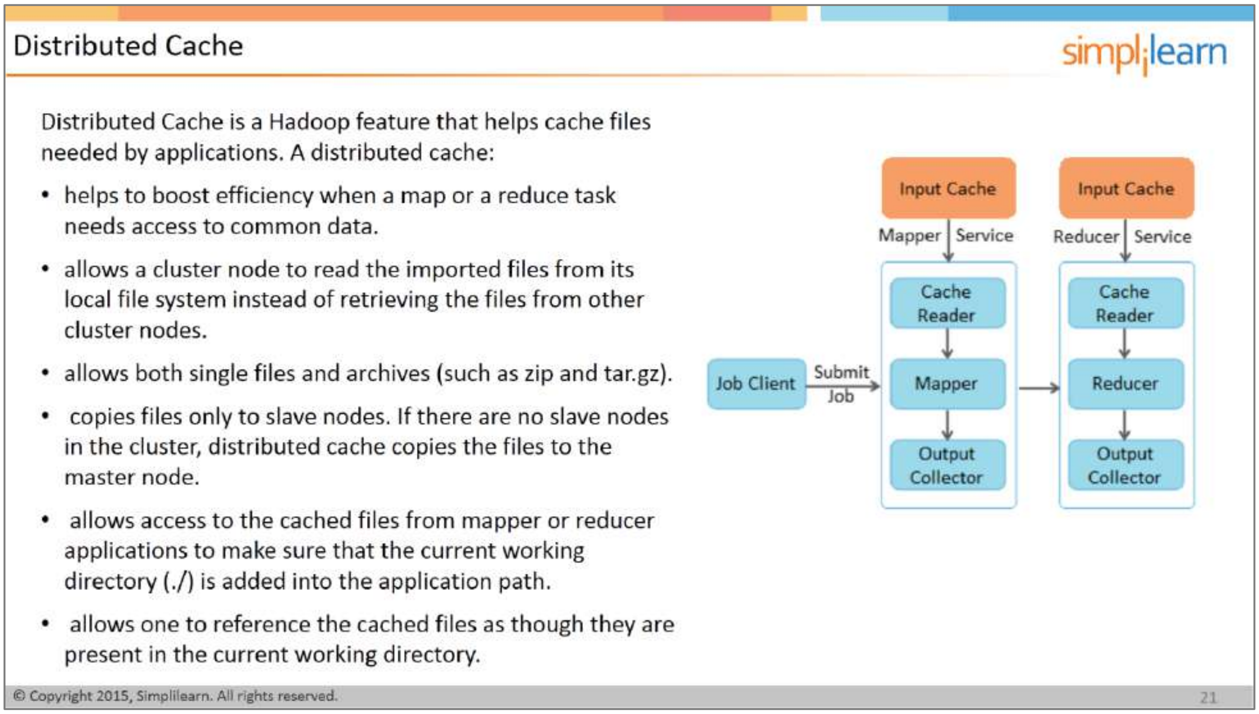
SequenceFileOutputFormat writes sequence files to save output. This represents a compact and compressed version of normal data blocks.

SequenceFileAsBinaryOutputFormat is responsible for writing key-value pairs that are in a raw binary format into a sequential file container.

MapFileOutputFormat writes MapFiles as the output. The keys in a MapFile are added in a specific order. The reducer then emits keys in the sorted order.

MultipleTextOutputFormat writes data to multiple files whose names are derived from output keys and values.

MultipleSequenceFileOutputFormat creates output in multiple files in a compressed form.

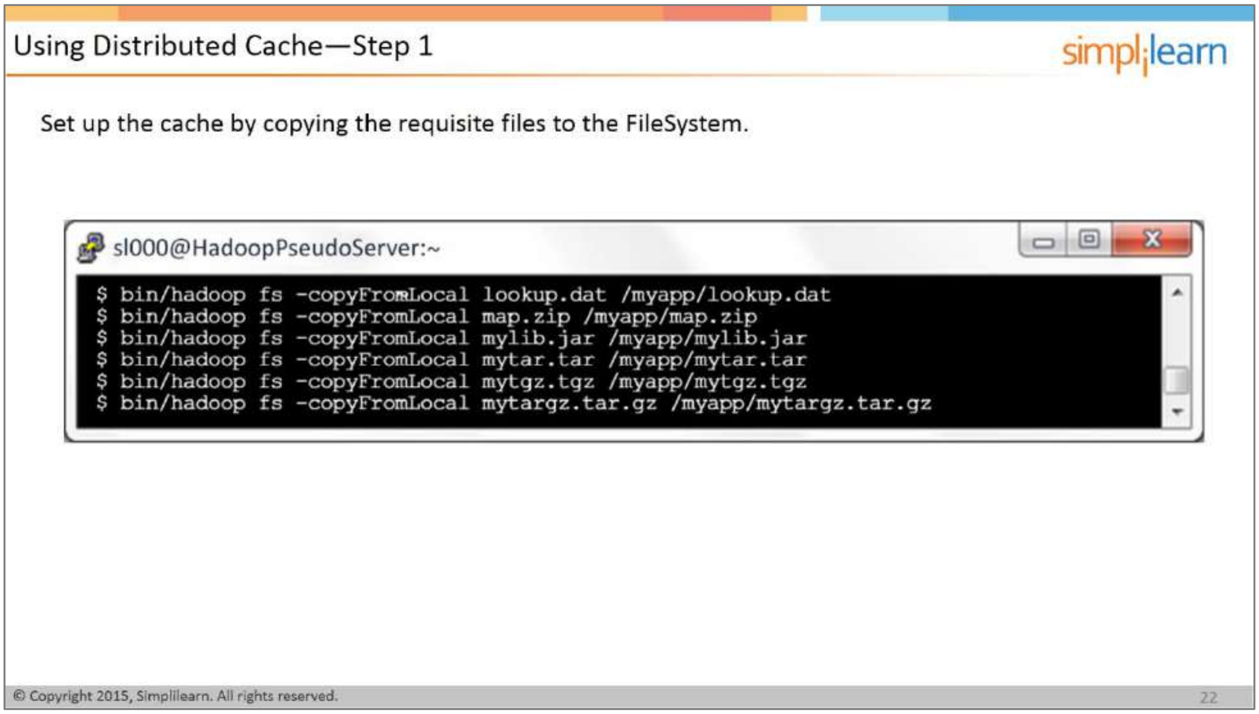


Distributed Cache

Distributed Cache is a Hadoop feature to cache files needed by applications.

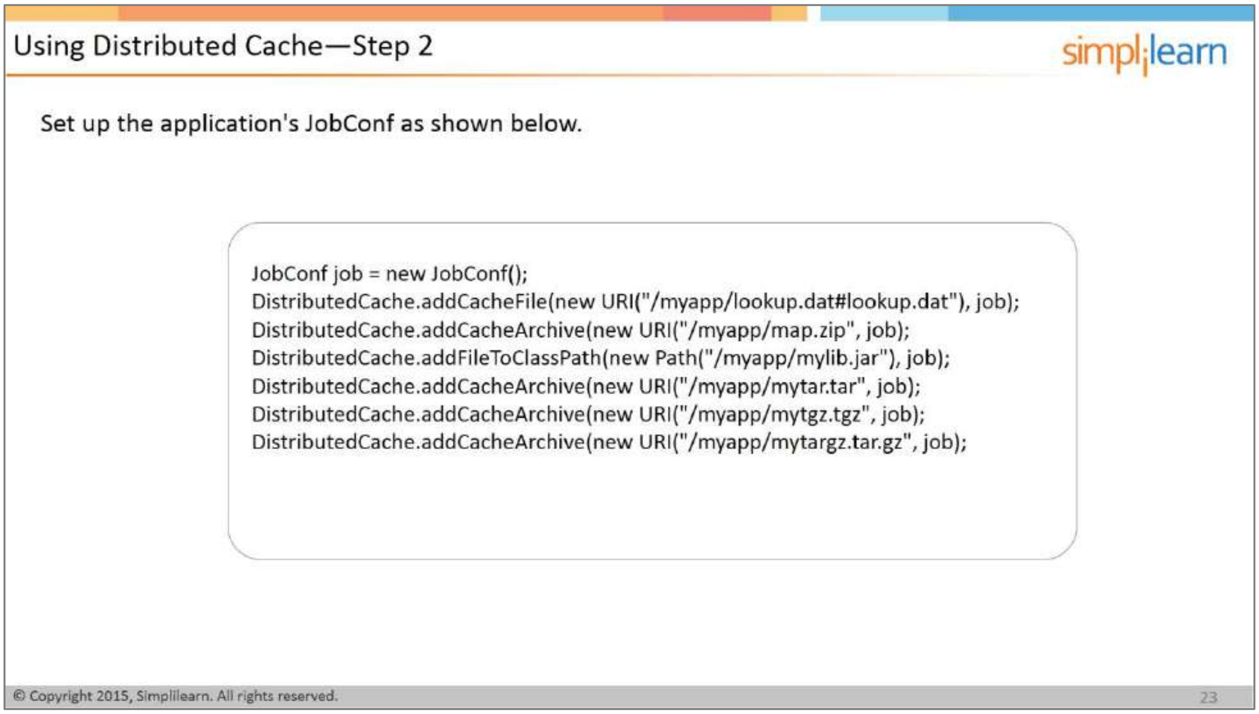
A distributed cache:

- helps boost efficiency when a map or a reduce task needs access to common data.
- lets a cluster node read the imported files from its local file system instead of retrieving the files from other cluster nodes.
- allows both single files and archives like zip and tar.gz.
- copies files only to slave nodes. If there are no slave nodes in the cluster, the distributed cache copies the files to the master node.
- allows access to the cached files from mapper or reducer applications to make sure that the current working directory is added into the application path.
- allows referencing of the cached files as though they are present in the current working directory.



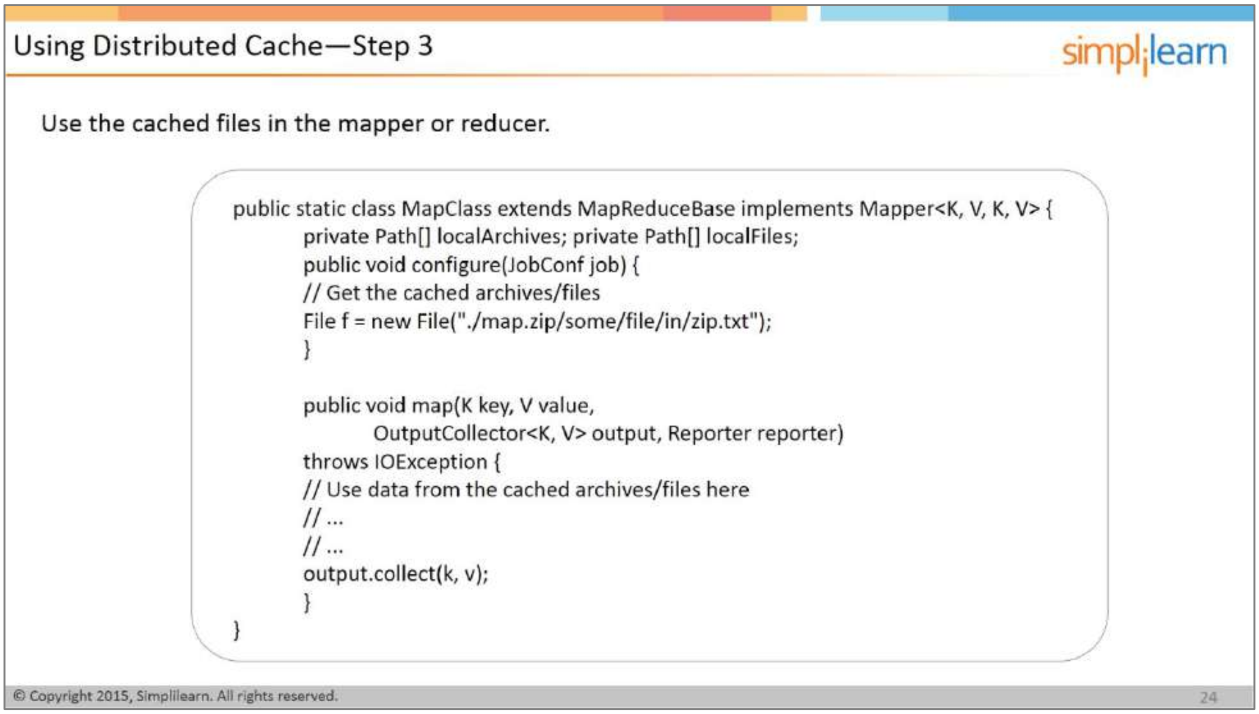
Using Distributed Cache—Step 1

Set up the cache by copying the requisite files to the FileSystem.



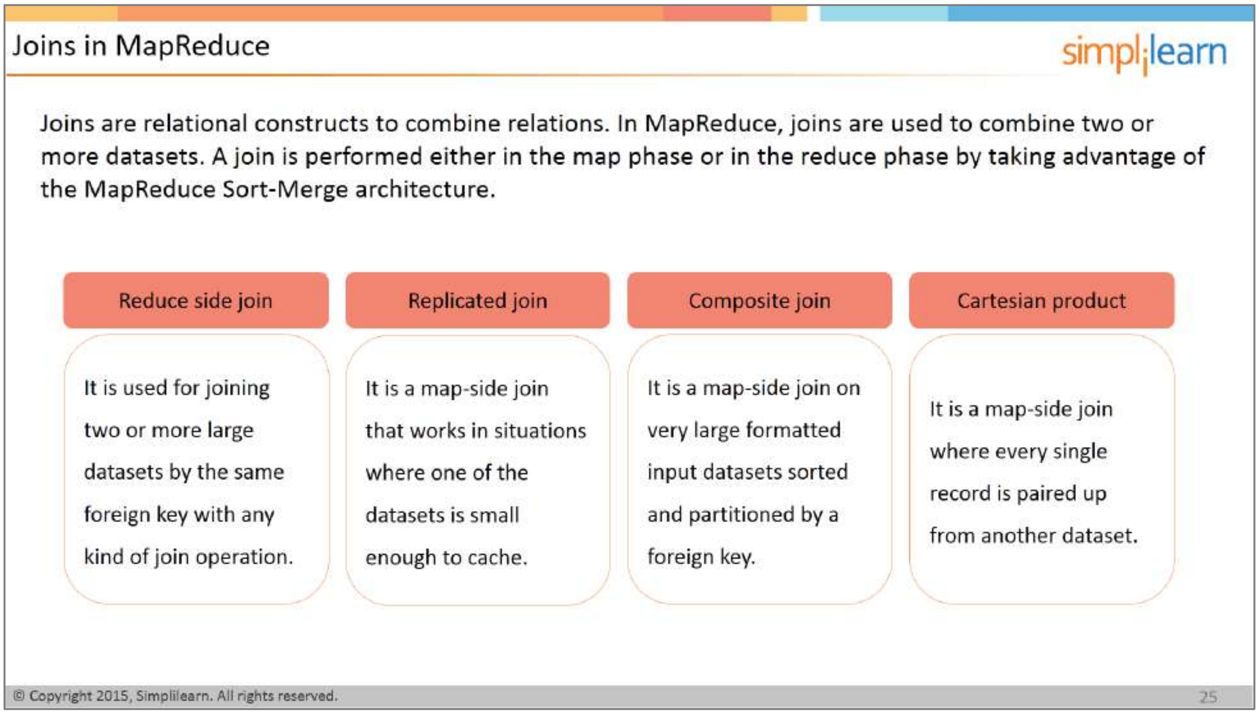
Using Distributed Cache—Step 2

Set up the application's JobConf as shown.



Using Distributed Cache—Step 3

Use the cached files in the mapper or reducer.

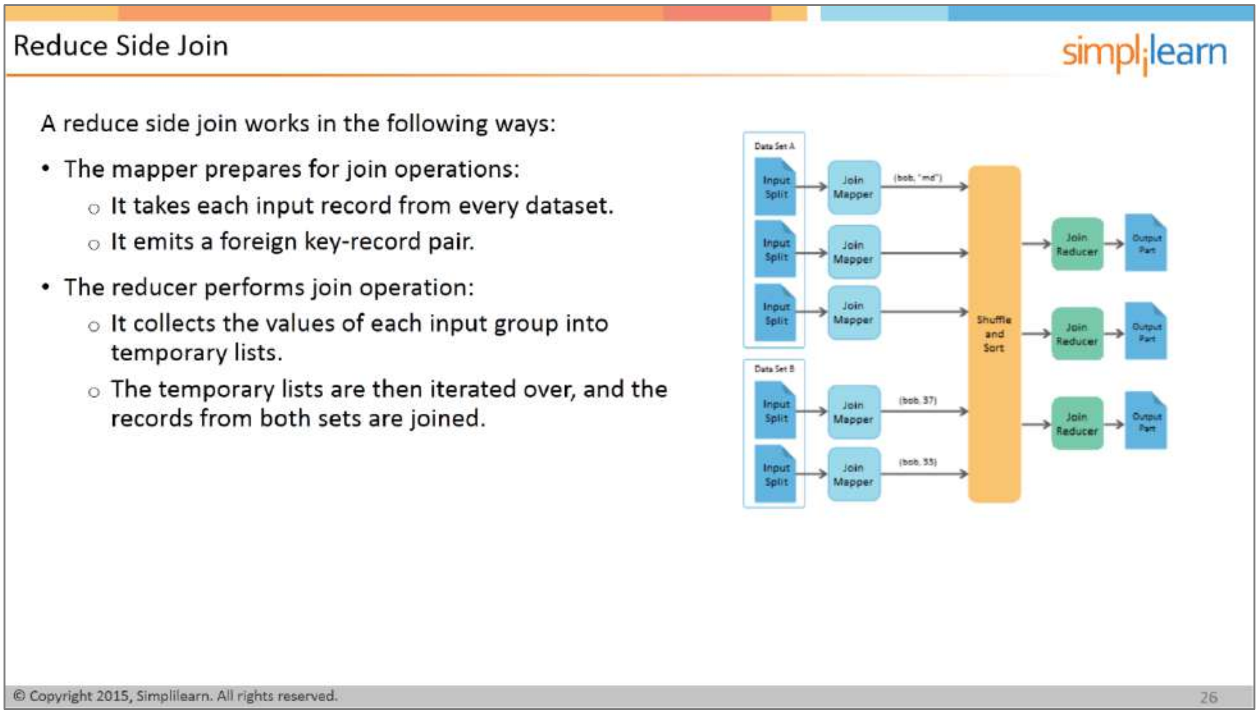


Joins in MapReduce

Joins are relational constructs that can be used to combine relations. In MapReduce, joins are applicable in situations where you have two or more datasets you want to combine. A join is performed either in the map phase or in the reduce phase by taking advantage of the MapReduce Sort-Merge architecture.

The various join patterns available in MapReduce are reduce side join, replicated join, composite join, and Cartesian product.

- A reduce side join is used for joining two or more large datasets by the same foreign key with any kind of join operation.
- A replicated join is a map-side join that works in situations where one of the datasets is small enough to cache.
- A composite join is a map-side join used on very large formatted input data sets sorted and partitioned by a foreign key.
- A Cartesian product is a map-side join where every single record is paired up with another dataset.

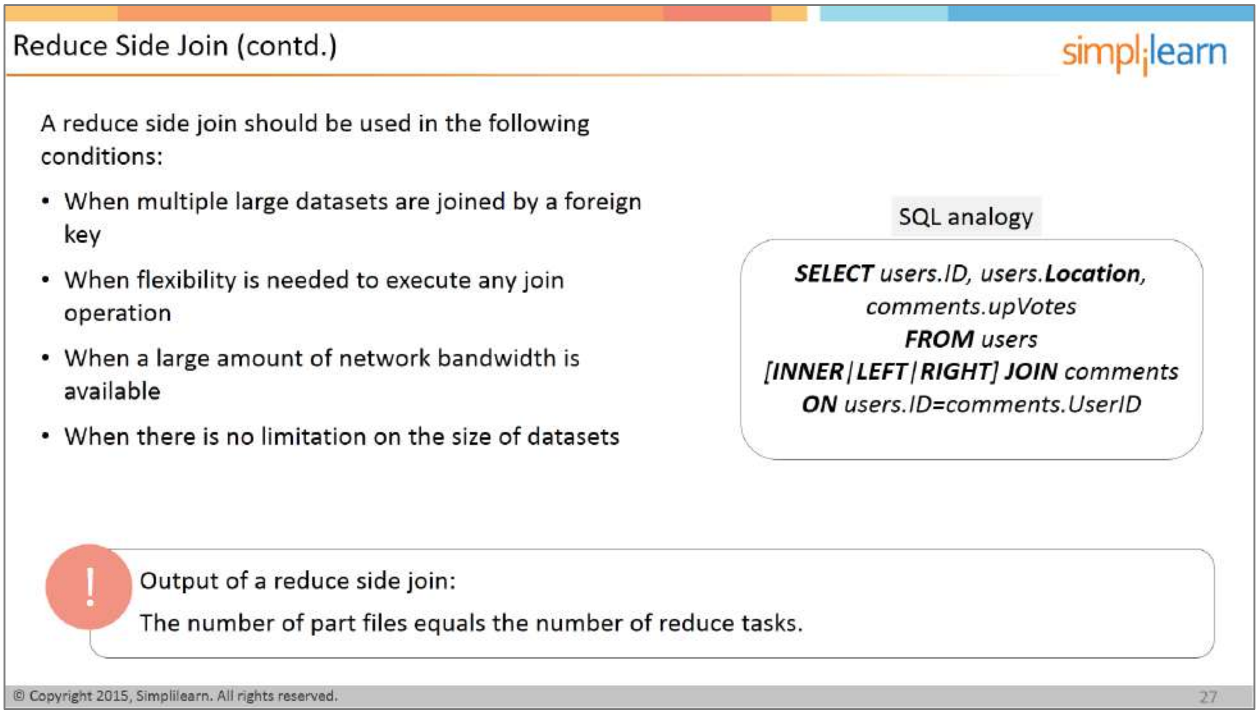


Reduce Side Join

A reduce side join works in the following ways:

The mapper prepares for join operations. It takes each input record from every dataset and emits a foreign key-record pair.

The reducer performs a join operation where it collects the values of each input group into temporary lists. The temporary lists are then iterated over, and the records from both sets are joined.



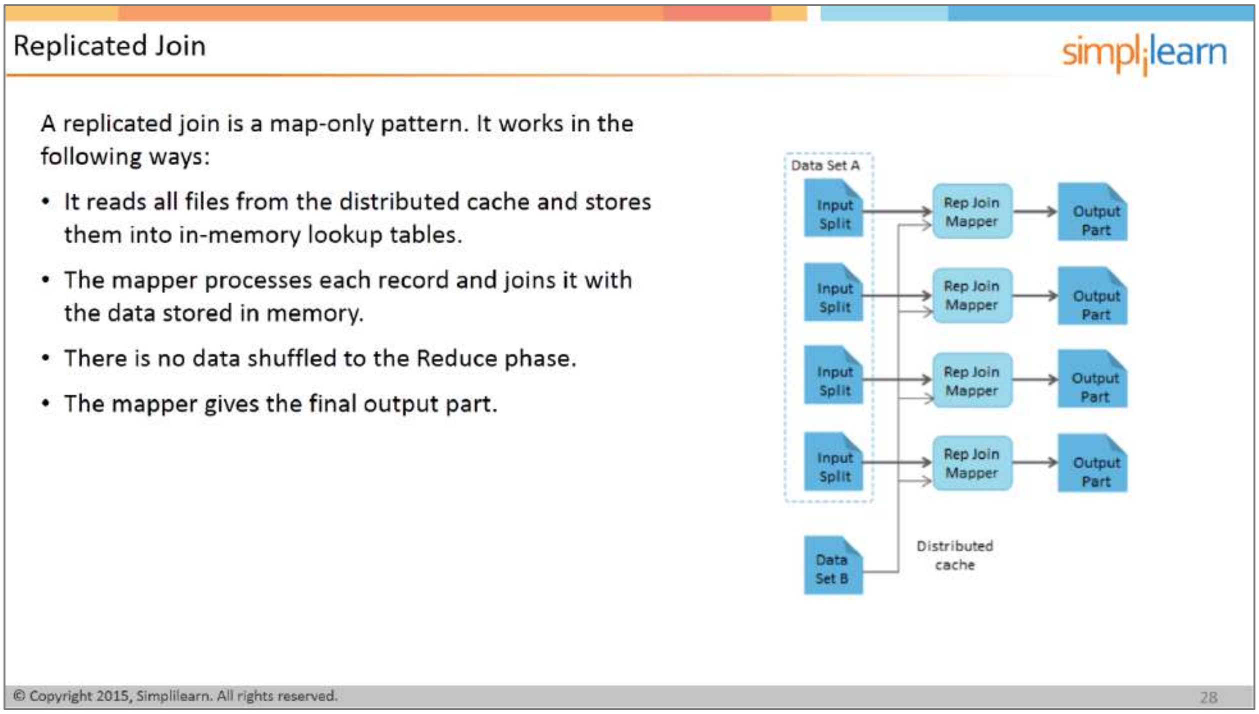
Reduce Side Join (contd.)

A reduce side join can be used:

- when multiple large datasets are being joined by a foreign key,
- when flexibility is needed to execute any join operation,
- when a large amount of network bandwidth is available, and
- when there is no limitation on the size of datasets.

An SQL analogy of a reduce side join is given.

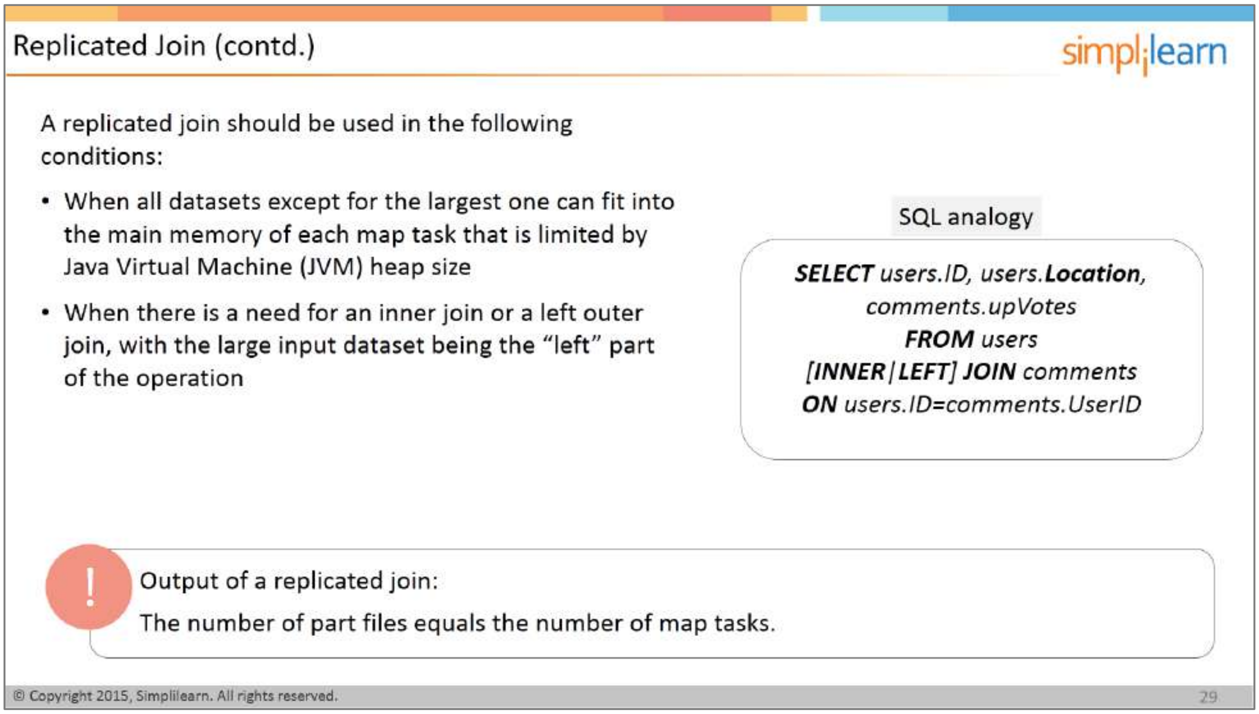
In the output of a reduce side join, the number of part files equals the number of reduce tasks.



Replicated Join

A replicated join is a map-only pattern that works as follows:

- It reads all files from the distributed cache and stores them into in-memory lookup tables.
- The mapper processes each record and joins it with the data stored in memory.
- There is no data shuffled to the reduce phase.
- The mapper provides the final output part.

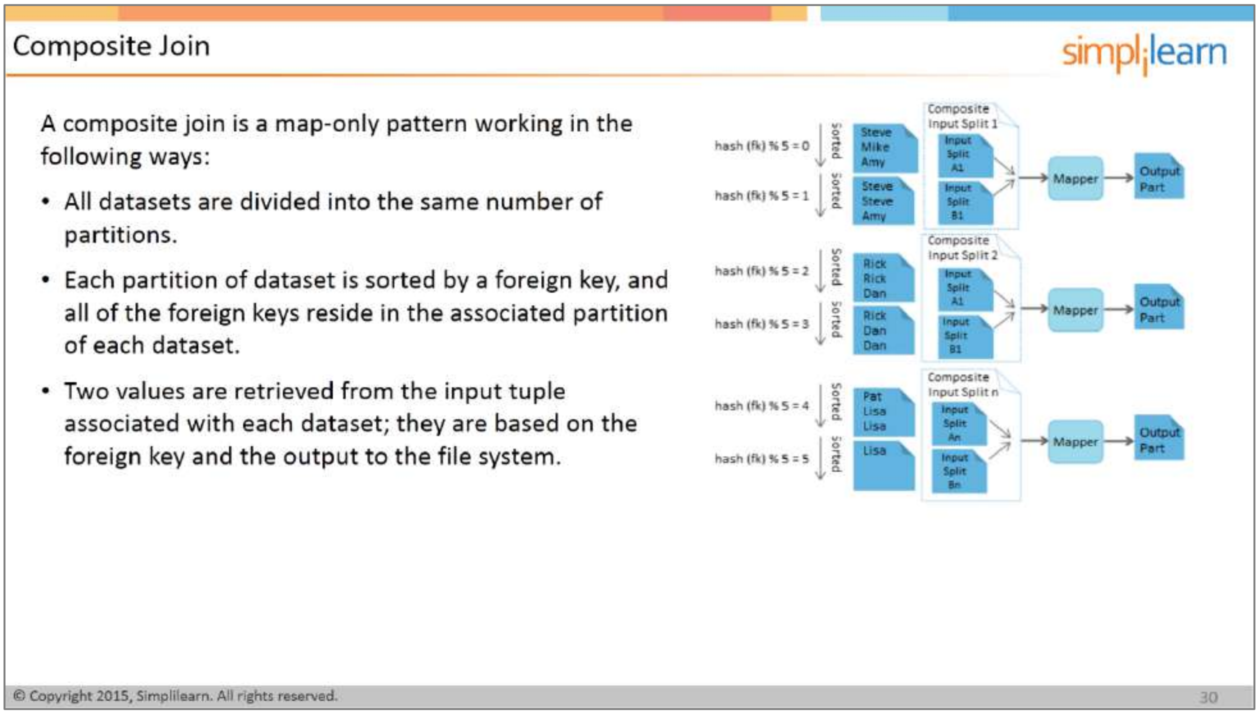


Replicated Join (contd.)

Replicated joins should be used:

- when all datasets, except for the largest one, can fit into the main memory of each map task that is limited by Java Virtual Machine or JVM heap size, and
- when there is a need for an inner join or a left outer join, with the large input dataset being the "left" part of the operation.

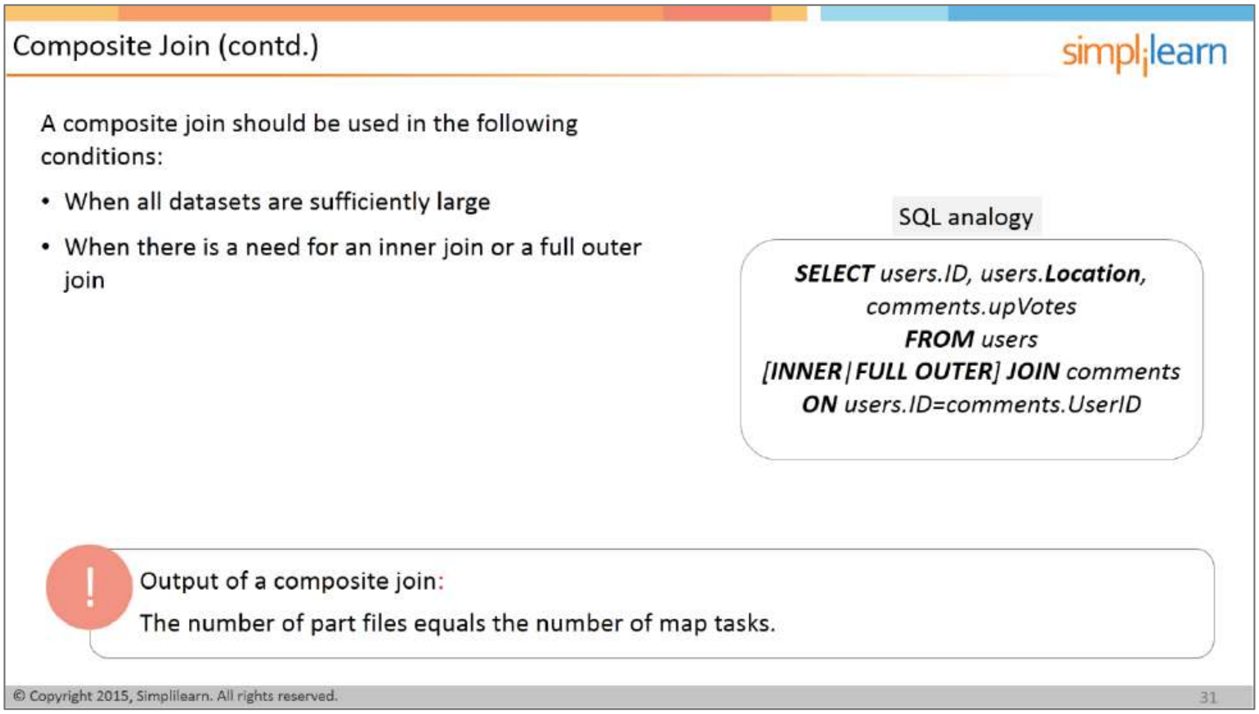
An SQL analogy of a replicated join is given.
In the output of a replicated join, the number of part files equals the number of map tasks.



Composite Join

A composite join is a map-only pattern working in the following ways:

- All datasets are divided into the same number of partitions.
- Each partition of the data set is sorted by a foreign key, and all the foreign keys reside in the associated partition of each dataset.
- Two values are retrieved from the input tuple associated with each data set based on the foreign key and the output to the file system.

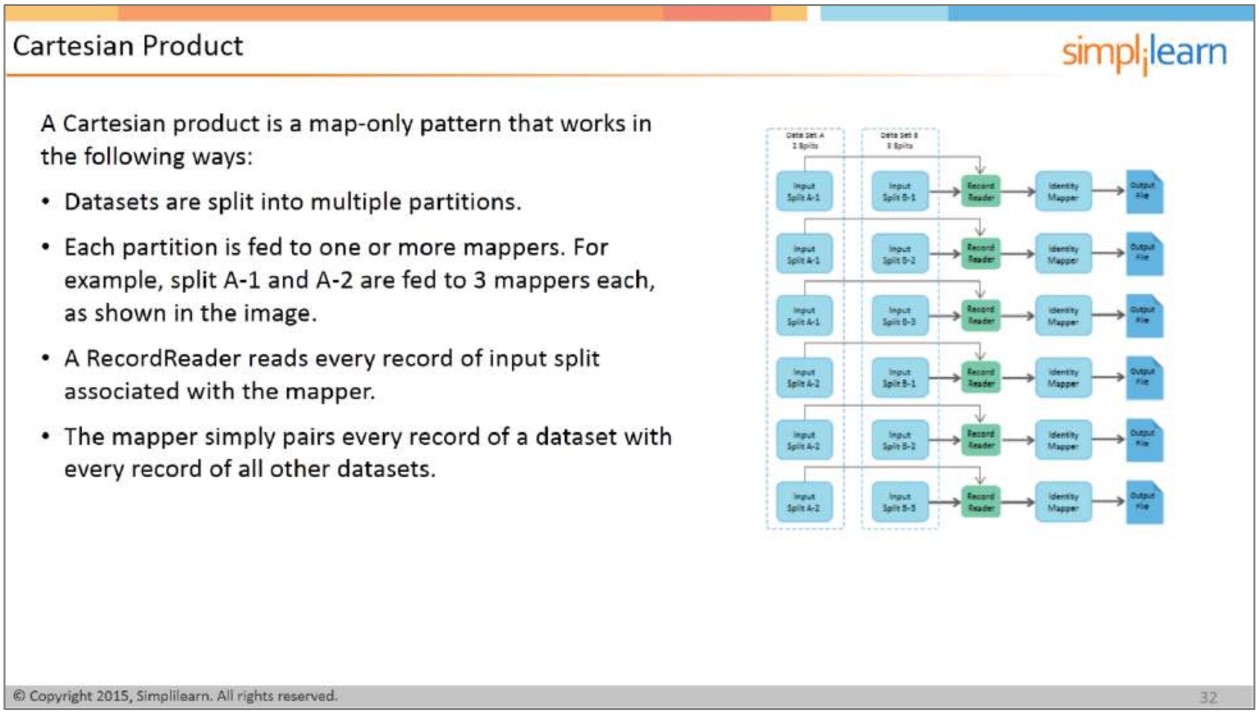


Composite Join (contd.)

Composite joins should be used:

- when all datasets are sufficiently large, and
- when there is a need for an inner join or a full outer join.

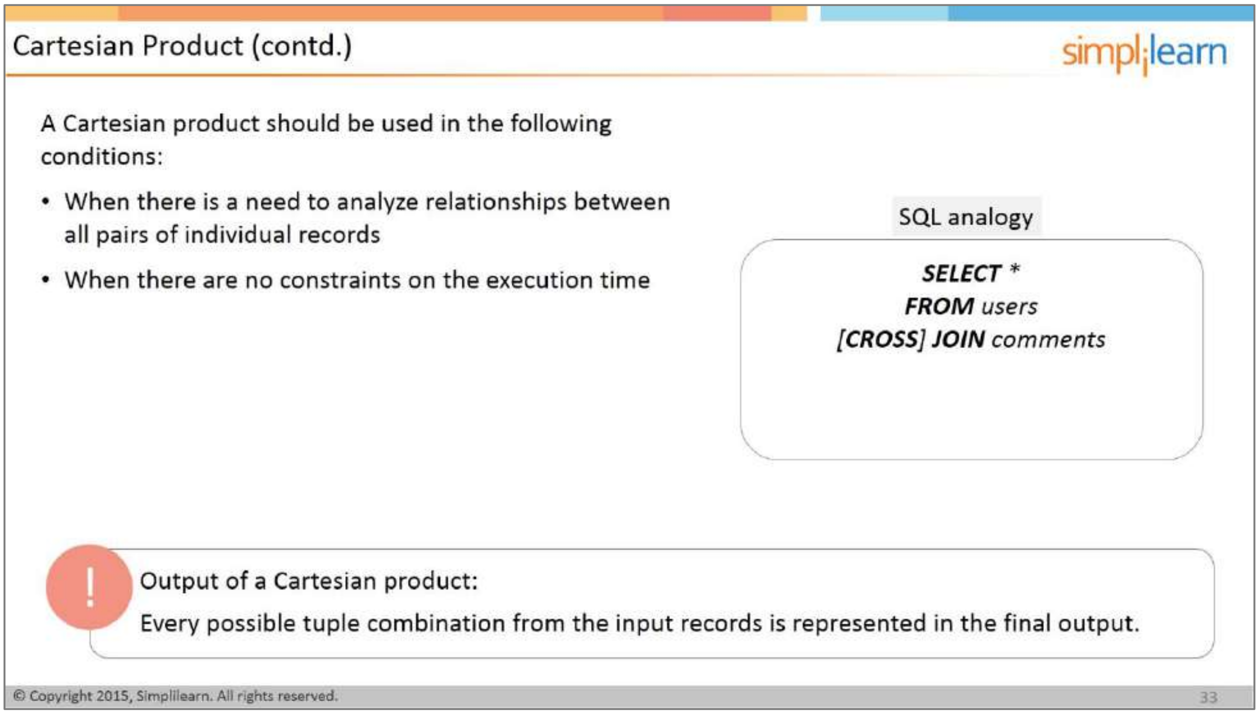
An SQL analogy of a composite join is displayed.
In the output of a composite join, the number of part files equals the number of map tasks.



Cartesian Product

A Cartesian product is a map-only pattern that works in the following ways:

- Datasets are split into multiple partitions.
- Each partition is fed to one or more mappers. For example, in the image shown here, split A-1 and A-2 are fed to three mappers each.
- A RecordReader reads every record of input split associated with the mapper.
- The mapper simply pairs every record of a dataset with every record of all other datasets.

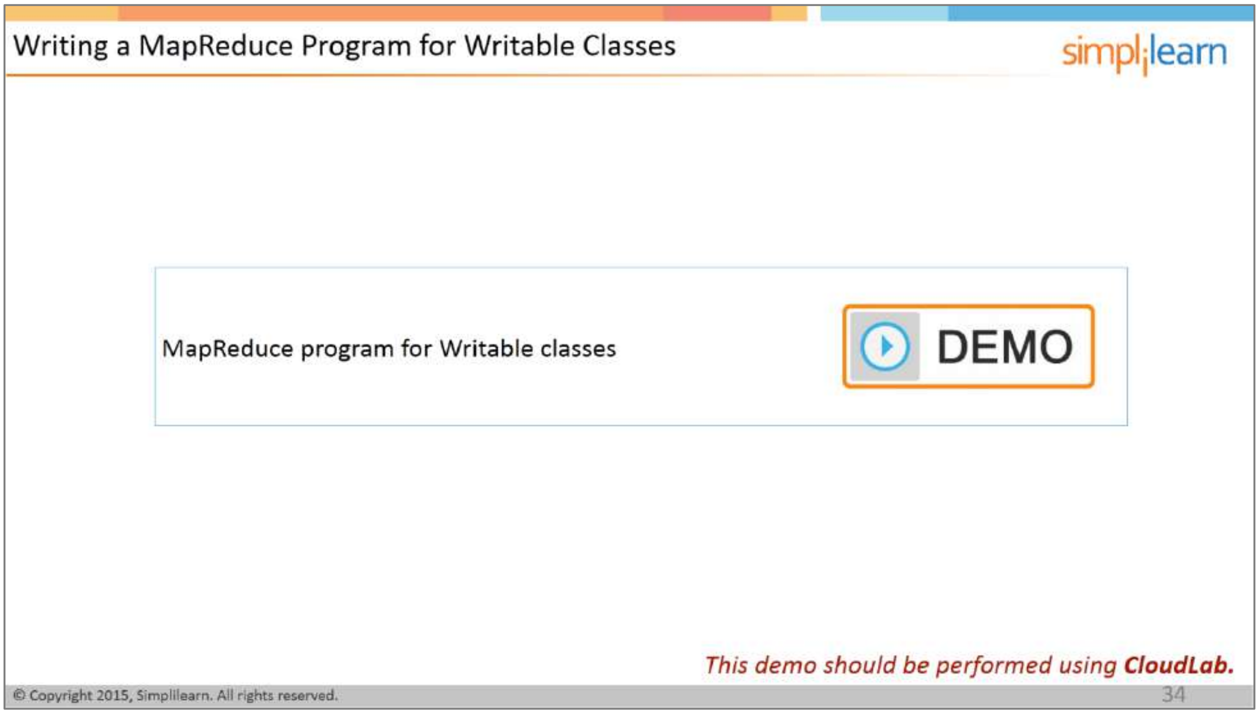


Cartesian Product (contd.)

A Cartesian product should be used:

- when there is a need to analyze relationships between all pairs of individual records, and
- when there are no constraints on the execution time.

In the output of a Cartesian product, every possible tuple combination from the input records is represented.



Writing a MapReduce Program for Writable Classes

1. Create a new project by clicking File Menu, New, and Project.
2. Select Java Project and click the Next button.
3. Type the name for the project and click the Next button to continue.
4. Select Libraries.
5. Click the Add External JARs button to add external JARs.
6. Select all jar files.
7. Click the OK button to add jar files.
8. Click the Add External JAR files button again.
9. Double-click the lib folder.
10. Select all JAR files.
11. Click the OK button.
12. Click the Finish button to continue.
13. Click the No button to continue.
14. Click the Play button to execute the program.
15. You will receive an error stating input is not received. Right-click the App.java file.
16. Click the Properties button.
17. Select the class file and click the Edit button.
18. Click the Arguments tab.
19. Enter the input and output paths.

20. Click the Apply button.
21. Click the OK button.
22. Click the Play button to execute the program.
23. The program is executed perfectly.
24. You can verify the output by visiting the web GUI.
25. Open the home page of the web GUI and click Browse the filesystem.
26. Click output2.
27. The part-r-00000 contains the output. Click the same to view the output of the program.
28. Thus we have successfully demonstrated the program using Writable class.

QUIZ 1

HDFS stands for ____________.

a. Hadoop Distributed Folder System

b. Hadoop Distributed Feature System

c. Hadoop Distributed File System

d. Hadoop Direct File System

 36

QUIZ 2

Which of the following files is responsible for decommissioning the nodes in a Hadoop cluster?

a. exclude

b. start-all.sh

c. stop-all.sh

d. start-dfs.sh

 38

QUIZ 3

Which of the following classes is responsible for decommissioning the nodes in a Hadoop cluster?

a. Text

b. IntWritable

c. FloatWritable

d. StringWritable

 40

QUIZ 4

Which of the following joins uses a reducer?

a. Reduce side join

b. Replicated join

c. Composite join

d. Cartesian product

 42

simplilearn

QUIZ 5

Which of the following allows Hadoop to read and write the data in a serialized form for transmission?

a. Distributed cache

b. HDFS API

c. Writable

d. Cartesian product

44

ANSWERS:

S.No.	Question	Answer & Explanation
1	HDFS stands for _____________.	c. HDFS stands for Hadoop Distributed File System.
2	Which of the following files is responsible for decommissioning the nodes in a Hadoop cluster?	a. An exclude file is responsible for decommissioning the nodes in a Hadoop cluster.
3	Which of the following classes is responsible for decommissioning the nodes in a Hadoop cluster?	d. StringWritable is responsible for decommissioning the nodes in a Hadoop cluster.
4	Which of the following joins uses a reducer?	a. A reduce side join uses a reducer that collects the values of each input group into temporary lists.
5	Which of the following allows Hadoop to read and write the data in a serialized form for transmission?	c. Writable defines a de/serialization protocol. Every data type in Hadoop is Writable.

Case Study

simplilearn

Scenario | Analysis | Solution

XY Invest provides investment advice to high net worth individuals and maintains stock market data of various exchanges. It handles stock market data critical for analysis and monitoring. This entails time-intensive processing of huge amounts of data, and vertical scaling is proving to be expensive. In its effort to find an alternate solution for processing, XY identifies that using Hadoop would reduce the company's cost and effort significantly. It wants to use the advanced features of MapReduce for data processing.

 50

Case Study

simplilearn

Scenario | Analysis | Solution

The company does research on Hadoop and finds that it is a popular solution for Big Data storage and processing. Hadoop has two components, HDFS for storage and MapReduce for processing.

A few advanced features of Hadoop:

1. It distributes processing so that jobs can be completed really quickly.
2. It is highly fault tolerant, so even if some tasks fail, they will be automatically retried.
3. It provides many writable interfaces for input and output.
4. It can distribute files needed for processing by using the distributed cache feature.

 51

Case Study

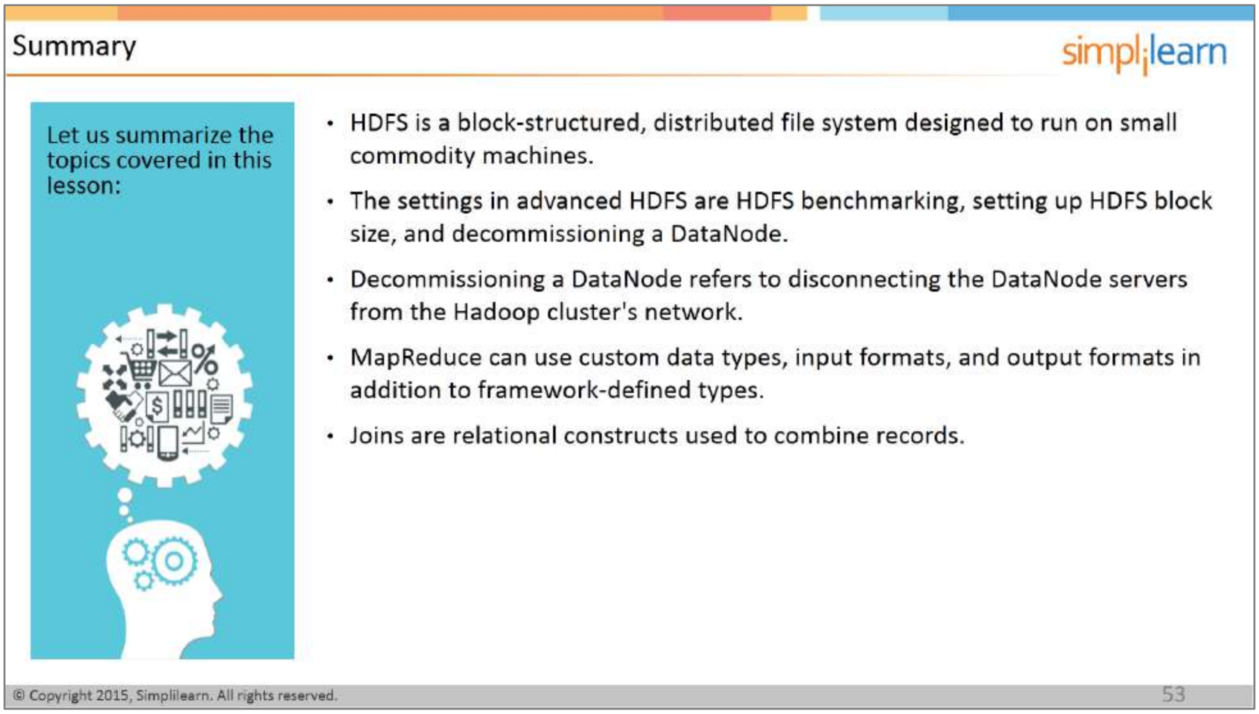
Scenario | Analysis | Solution

Perform the following steps to explore stock market data using MapReduce:

1. Get stock market data for Google.
2. Get the trade data for a customer (buys and sells of Google stock).
3. Store the trade data in HDFS.
4. Create a MapReduce program that uses distributed cache to distribute the stock market data. The MapReduce program also uses various writable interfaces of Hadoop.
5. Process the trade data and analyze the customer's profit/loss.

hadoop

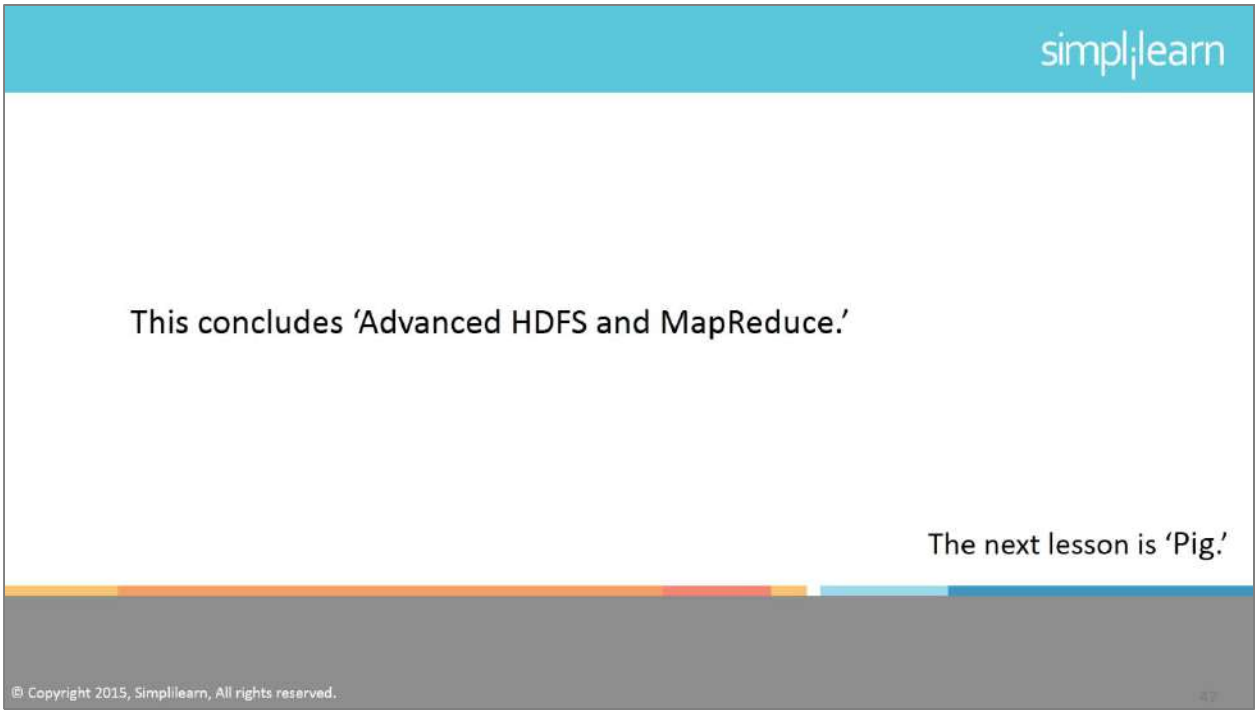
52



Summary

Let's summarize the topics covered in this lesson:

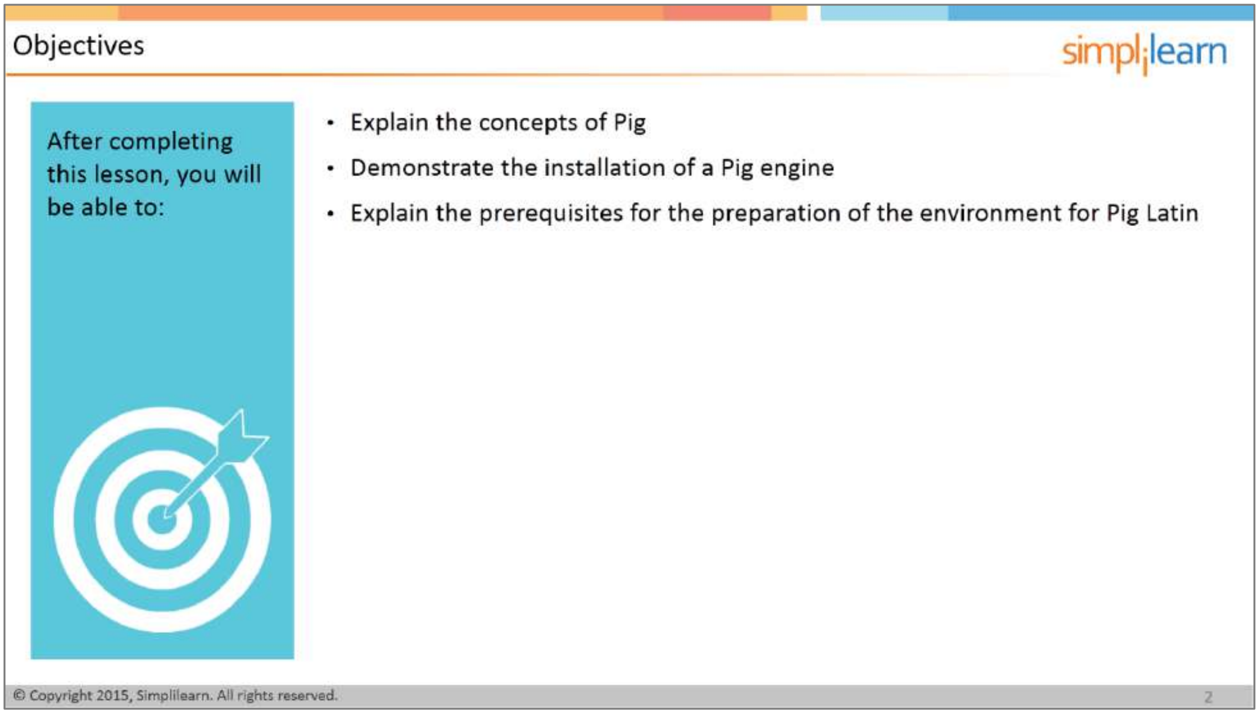
- HDFS is a block-structured distributed file system designed to run on small, commodity machines.
- The settings in advanced HDFS are HDFS benchmarking, setting up HDFS block size, and decommissioning a DataNode.
- Decommissioning a DataNode refers to disconnecting the DataNode servers from the Hadoop cluster's network.
- MapReduce can use custom data types, input formats, and output formats in addition to framework-defined types.
- Joins are relational constructs used to combine records.



Conclusion

This concludes 'Advanced HDFS and MapReduce.' In the next lesson, we will focus on 'Pig.'

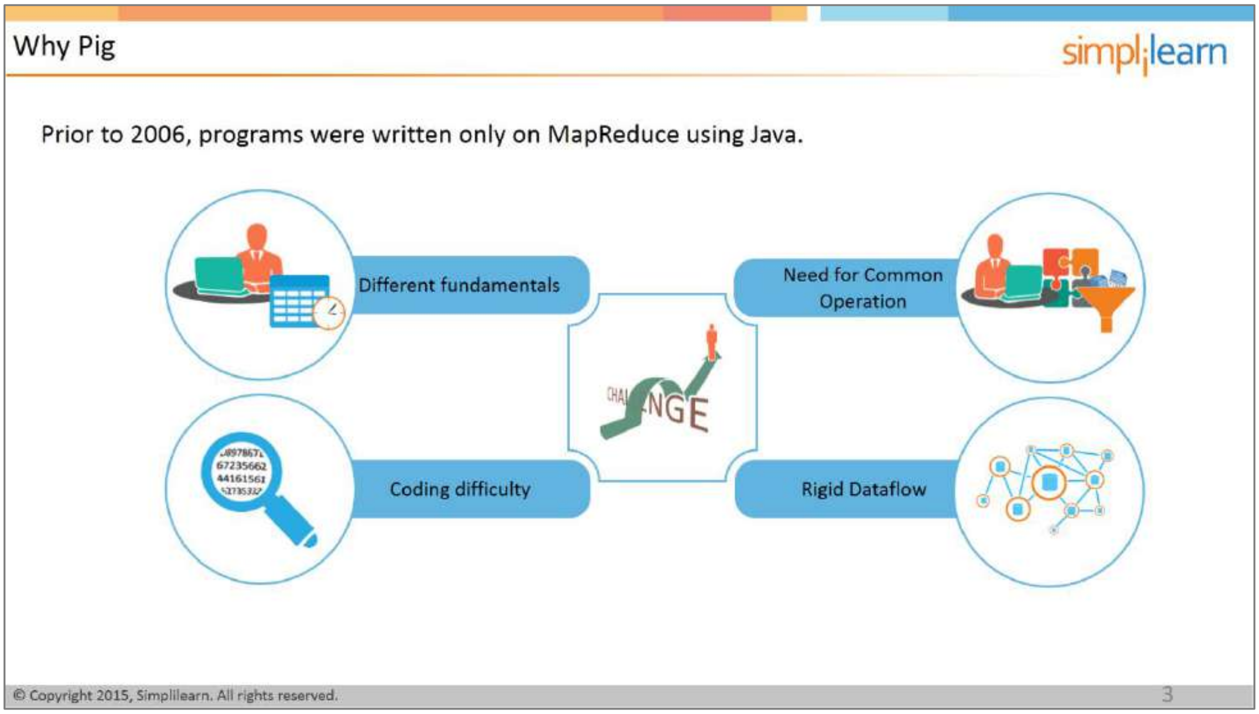
Lesson 6—Pig



Objectives

After completing this lesson, you will be able to

- Explain the concepts of Pig
- Demonstrate the installation of a Pig engine
- Explain the prerequisites for the preparation of the environment for Pig Latin

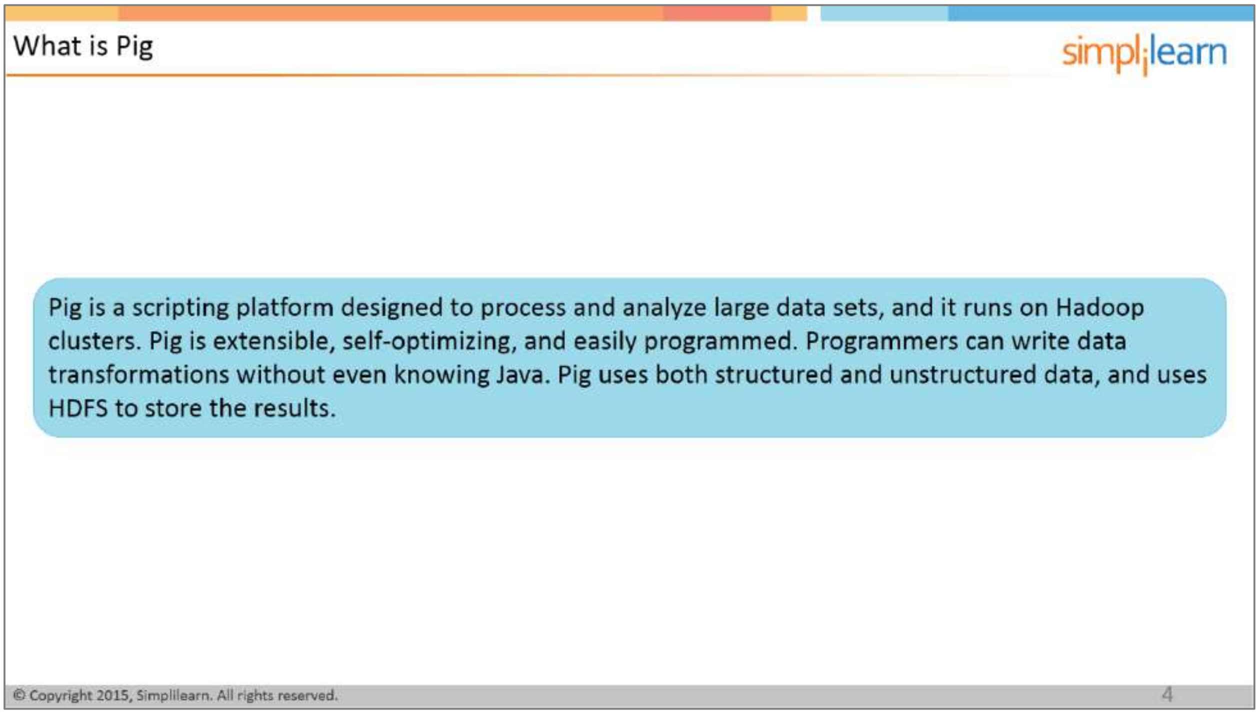


Why Pig

Prior to 2006, programs were written only on MapReduce using Java.

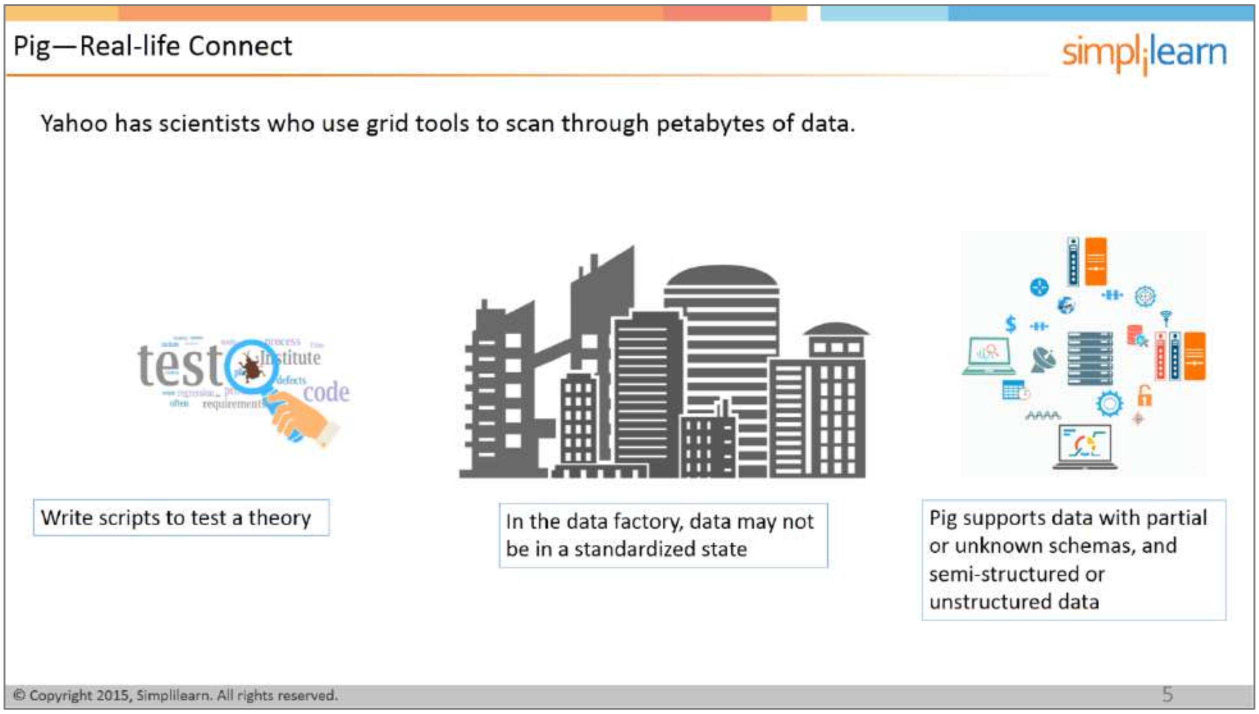
A developer had to mind the map, split, and reduce fundamentals while creating a program, for which they needed common operations such as joining, filtering, and so on. There was increasing difficulty faced while maintaining, optimizing, and extending the code. Consequently, the production time increased. The dataflow in MapReduce is quite rigid. You cannot use the output of a task as the input of another.

Pig was created in late 2006 by Yahoo researchers. It later became an Apache open-source project. It is another language besides Java, in which MapReduce programs can be written.



What is Pig

Pig is a scripting platform designed to process and analyze large data sets, and it runs on Hadoop clusters. Pig is extensible, self-optimizing, and easily programmed. Programmers can write data transformations without even knowing Java. Pig uses both structured and unstructured data, and uses HDFS to store the results.

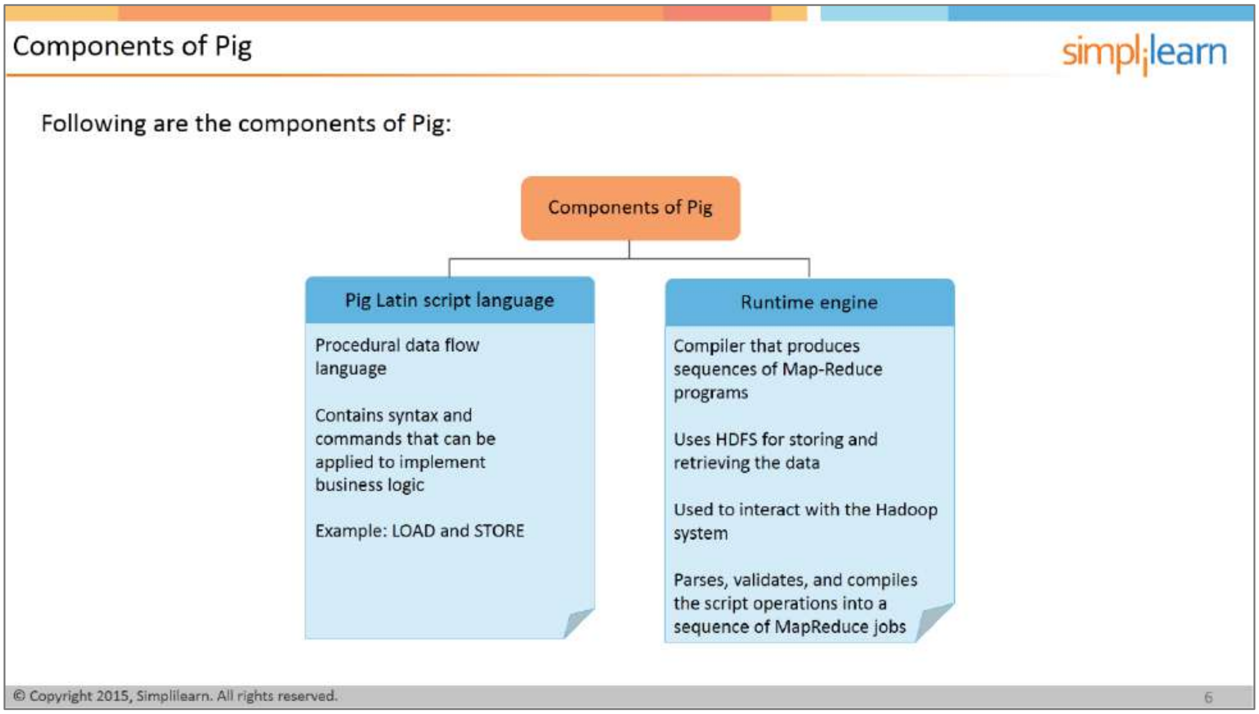


Real-life Connect

Yahoo has scientists who use grid tools to scan through petabytes of data. Many of them write scripts to test a theory or gain deeper insights.

But in the data factory, data may not be in a standardized state.

This makes Pig a good option as it supports data with partial or unknown schemas, and semi-structured or unstructured data.

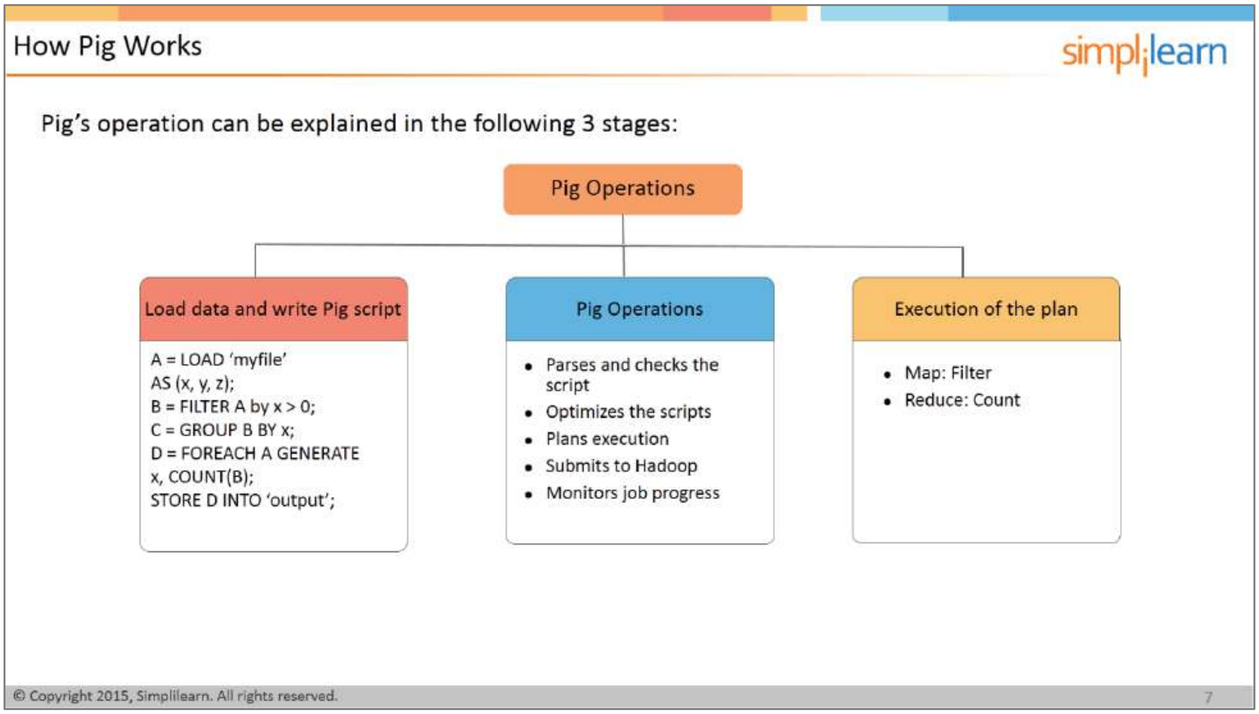


Components of Pig

The two major components of Pig are the Pig Latin script language and a runtime engine.

The Pig Latin script is a procedural data flow language. It contains syntax and commands that can be applied to implement business logic. Examples of Pig Latin are LOAD and STORE.

The runtime engine is a compiler that produces sequences of MapReduce programs. It uses HDFS for storing and retrieving data. It is also used to interact with the Hadoop system (HDFS and MapReduce). It parses, validates, and compiles the script operations into a sequence of MapReduce jobs.



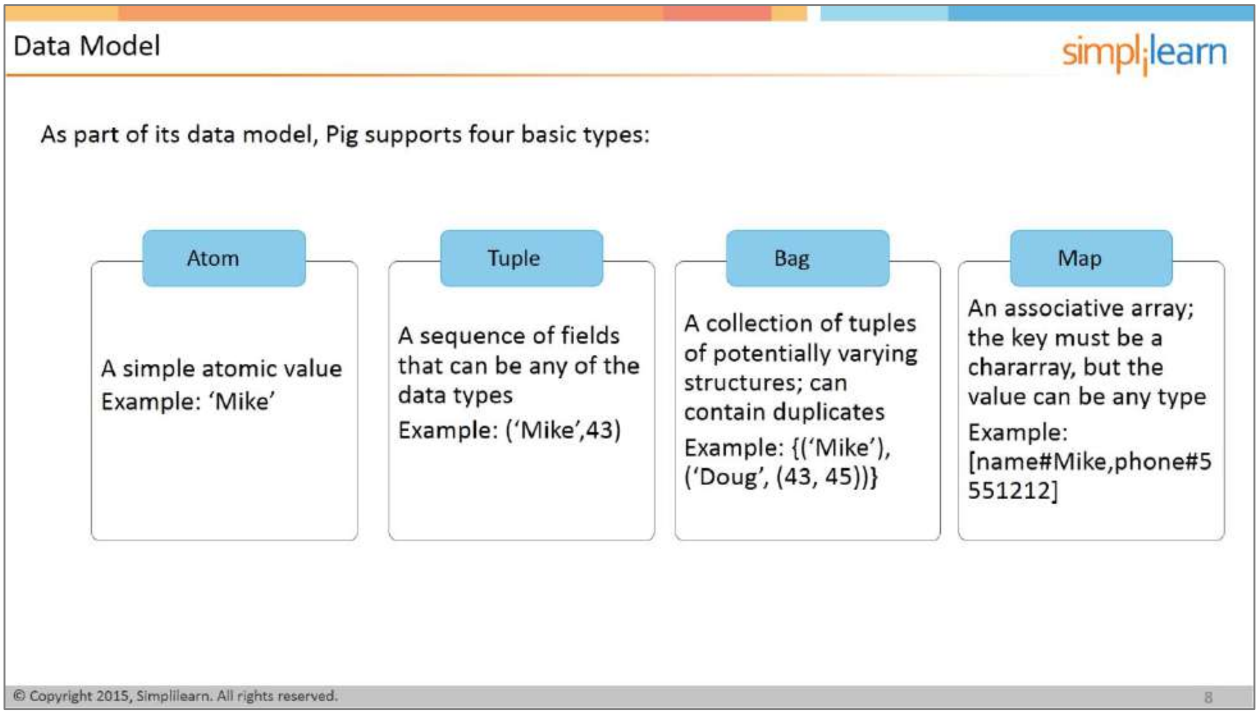
How Pig Works

Pig's operation can be explained in three stages.

1. In the first stage, the data is loaded, and a Pig script is written.

2. In the second stage, the Pig execution engine parses and checks the script. If it passes, the script is optimized, and a logical and physical plan is generated for execution. The job is submitted to Hadoop as a job comprising Map and Reduce tasks. Pig monitors the status of jobs using Hadoop API and reports the status to its client.

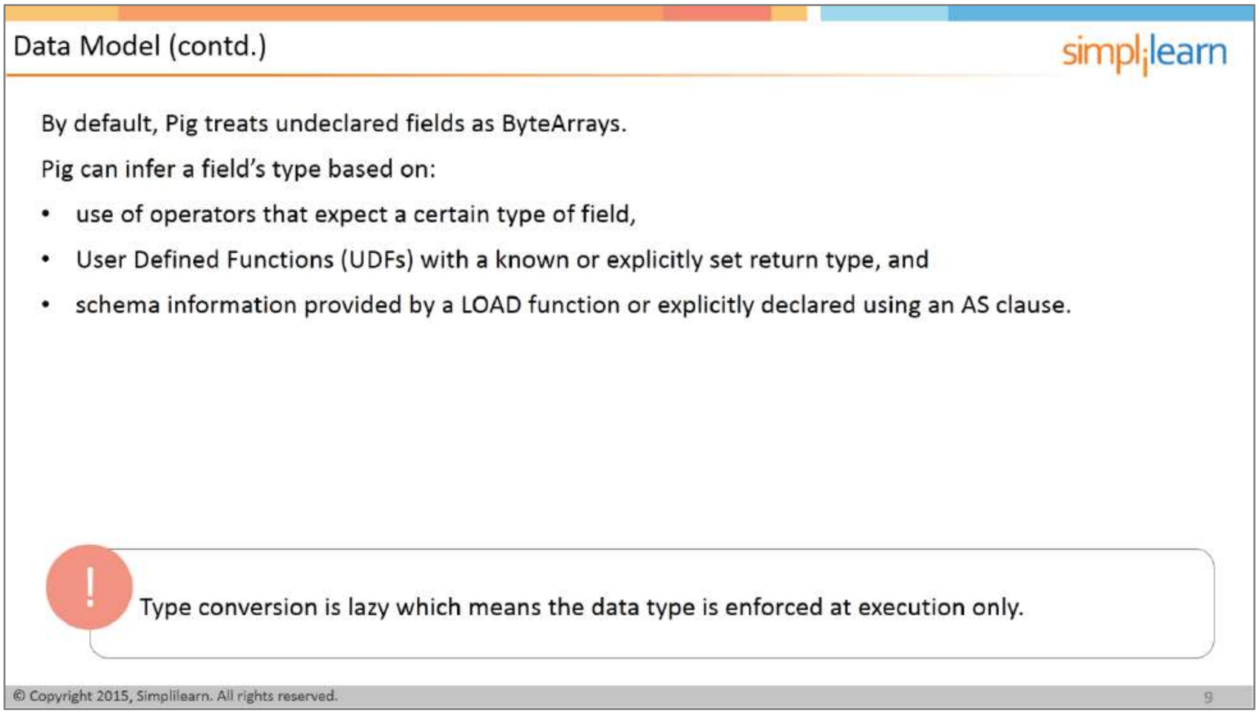
3. In the execution stage, the results are dumped on the screen or are stored in HDFS depending on the user command.



Data Model

As part of its data model, Pig supports four basic types.

1. The first is Atom, which is a simple atomic value like int, long, double, or string.
2. The Second is Tuple, which is a sequence of fields that can be of any data type.
3. The third is Bag, which is a collection of tuples of potentially varying structures and can contain duplicates.
4. Finally, there is Map, which is an associative array. The key must be a chararray, but the value can be of any type.

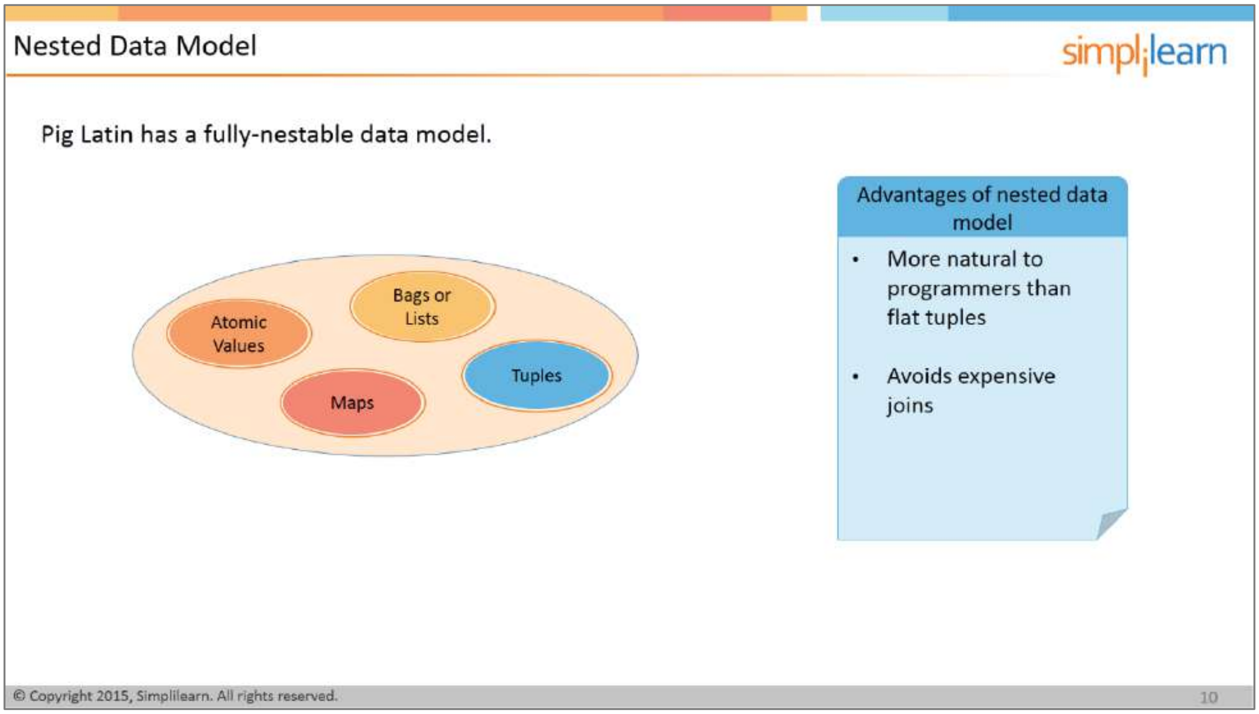


Data Model (contd.)

By default, Pig treats undeclared fields as bytearrays, which are collections of uninterpreted bytes.

Pig can infer a field's type based on the use of operators that expect a certain type of field. It can also use User Defined Functions, or UFDs, with a known or explicitly set return type. Furthermore, it can infer the field type based on schema information provided by a LOAD function or explicitly declared using an AS clause.

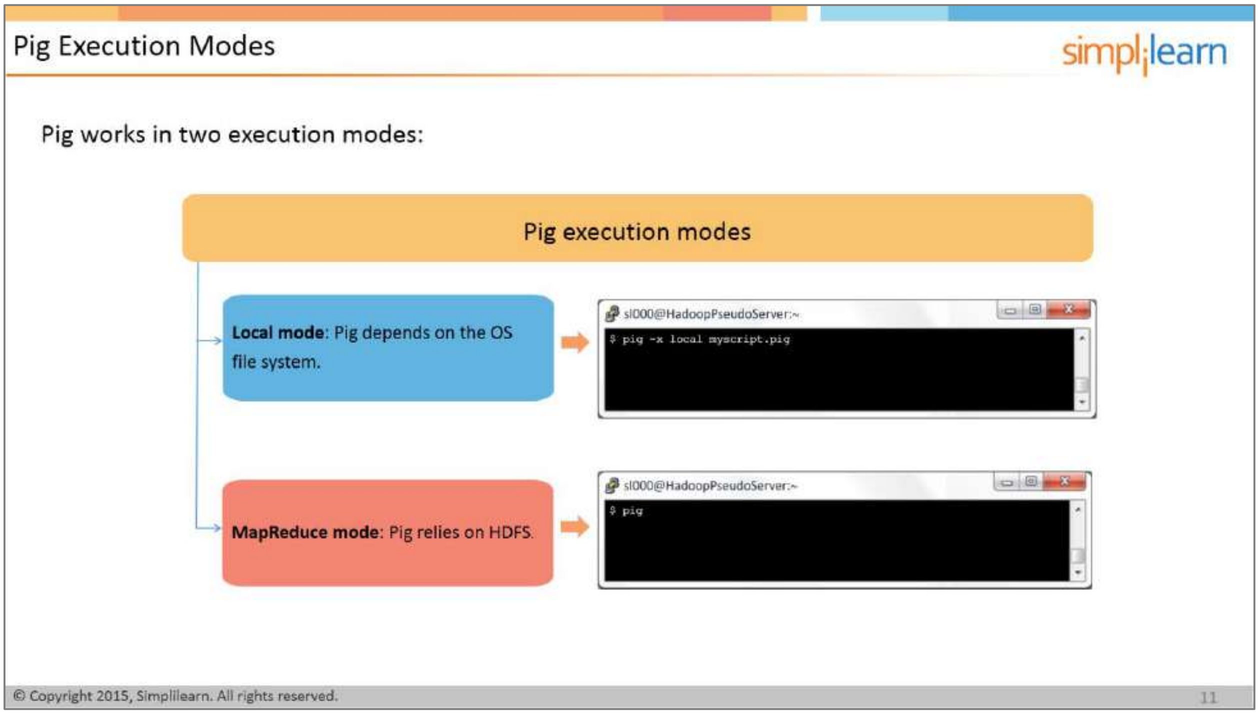
Please note that type conversion is lazy, which means the data type is enforced at the point of execution only.



Nested Data Model

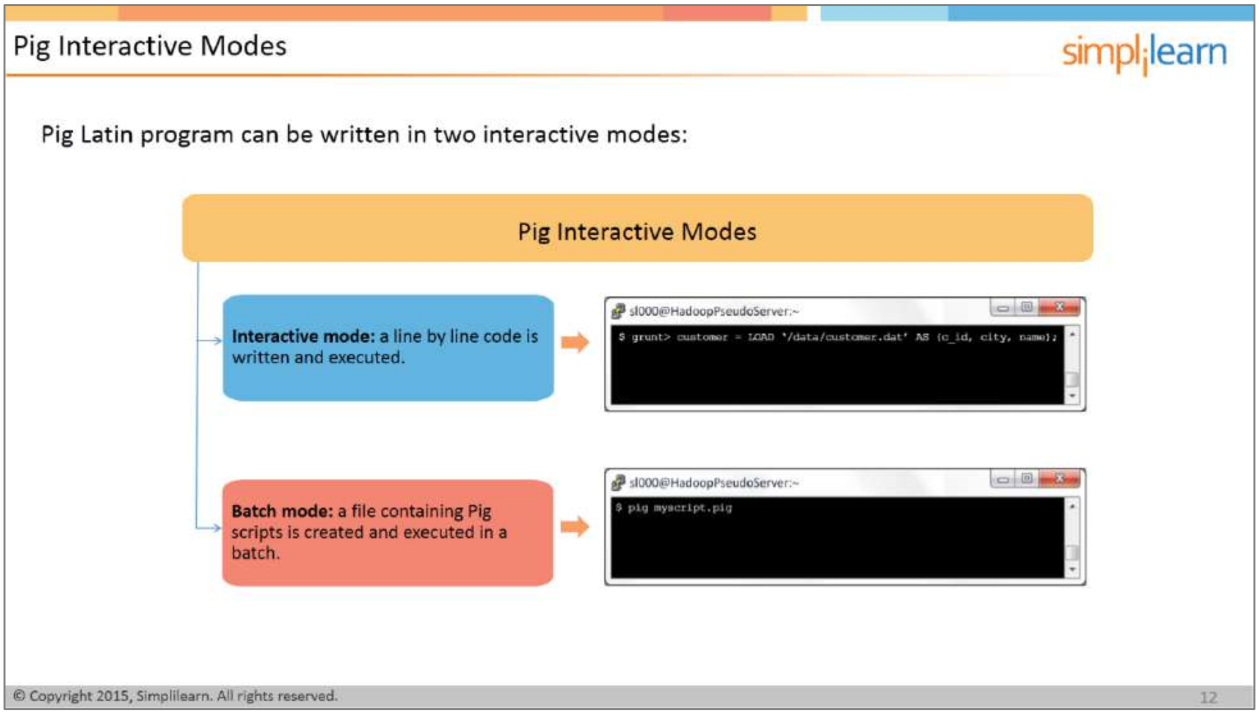
Pig Latin has a fully-nestable data model with Atomic values, Tuples, Bags or lists, and Maps. This implies one data type can be nested within another, as shown in the image.

The advantage is that this is more natural to programmers than flat Tuples. Also, it avoids expensive joins.



Pig Execution Modes

Pig works in two execution modes: Local and MapReduce. In local mode, Pig engine takes input from Linux file system, and the output is stored in the same file system. In MapReduce mode, Pig engine directly interacts and executes in HDFS and MapReduce.

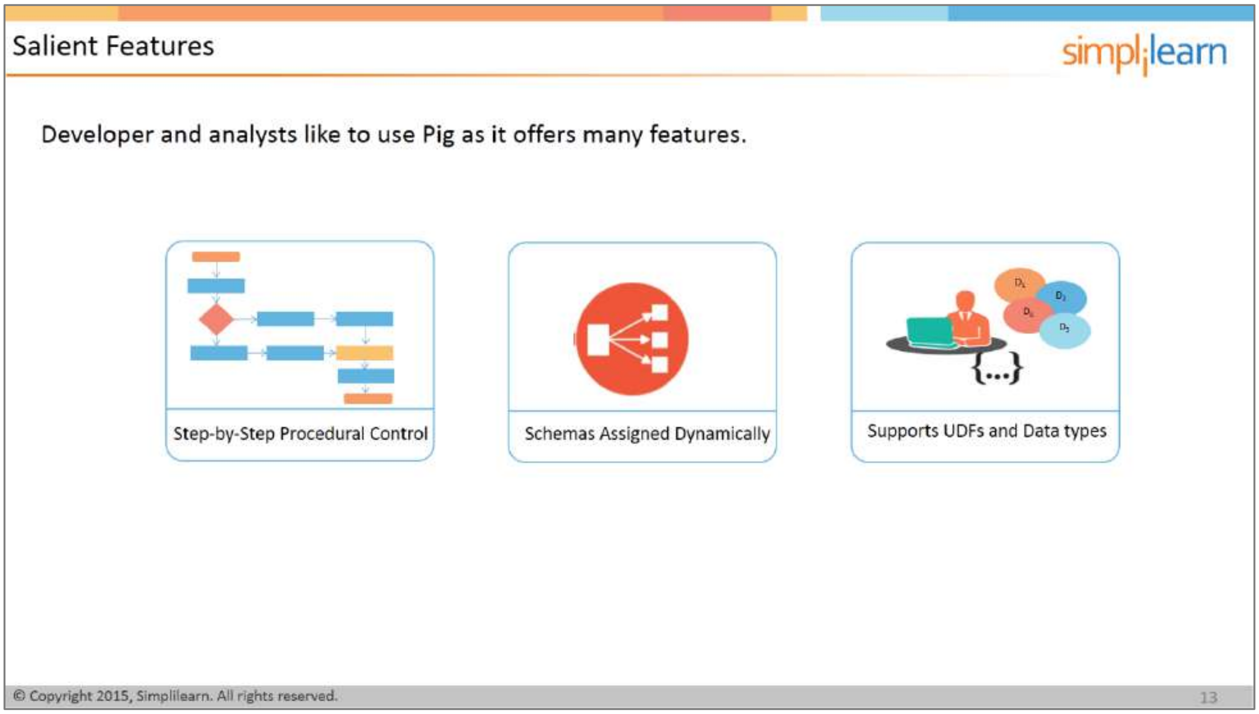


Pig Interactive Modes

The two modes in which a Pig Latin program can be written are Interactive and Batch.

Interactive mode means coding and executing the script, line by line, as shown in the image.

In Batch mode, all scripts are coded in a file with the extension .pig, and the file is directly executed.



Salient Features

Developer and analysts like to use Pig as it offers many features. Some of the features are as follows:

- It provides step-by-step procedural control and the provision to operate directly over files.
- It has schemas that, though optional, can be assigned dynamically.

It also supports User Defined Functions, or UDFs, and various data types.

Pig vs. SQL

The differences between Pig and SQL are given below:

Difference	Pig	SQL
Definition	Scripting language used to interact with HDFS	Query language used to interact with databases
Query Style	Step-by-step	Single block
Evaluation	Lazy evaluation	Immediate evaluation
Pipeline Splits	Pipeline splits are supported	Requires the join to be run twice or materialized as an intermediate result

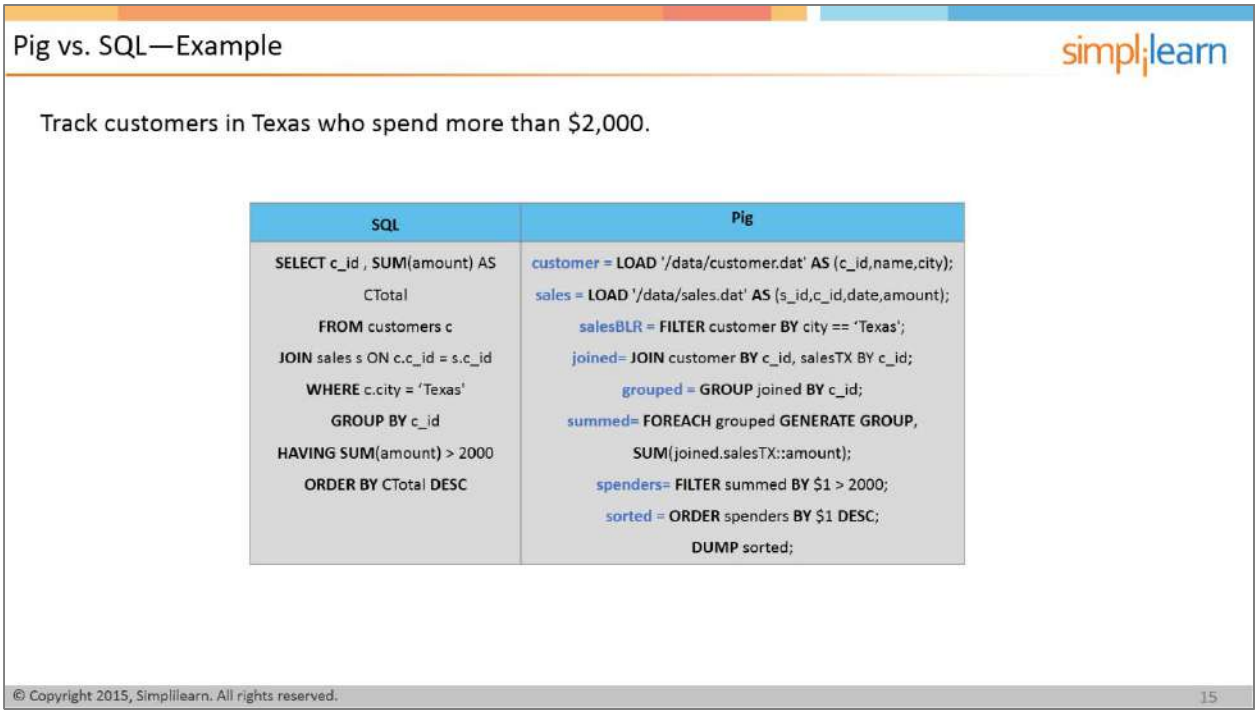
Pig vs. SQL

The first difference between Pig and SQL is that Pig is a scripting language used to interact with HDFS. SQL is a query language used to interact with databases residing in the database engine.

In terms of query style, Pig offers a step-by-step execution style compared to the single block execution style of SQL.

Pig does a lazy evaluation, which means that data is processed only when the STORE or DUMP command is encountered. Whereas, SQL offers immediate evaluation of a query.

Pipeline Splits are supported in Pig, but in SQL, you may need to run the “join” command twice for the result to be materialized as an intermediate result.



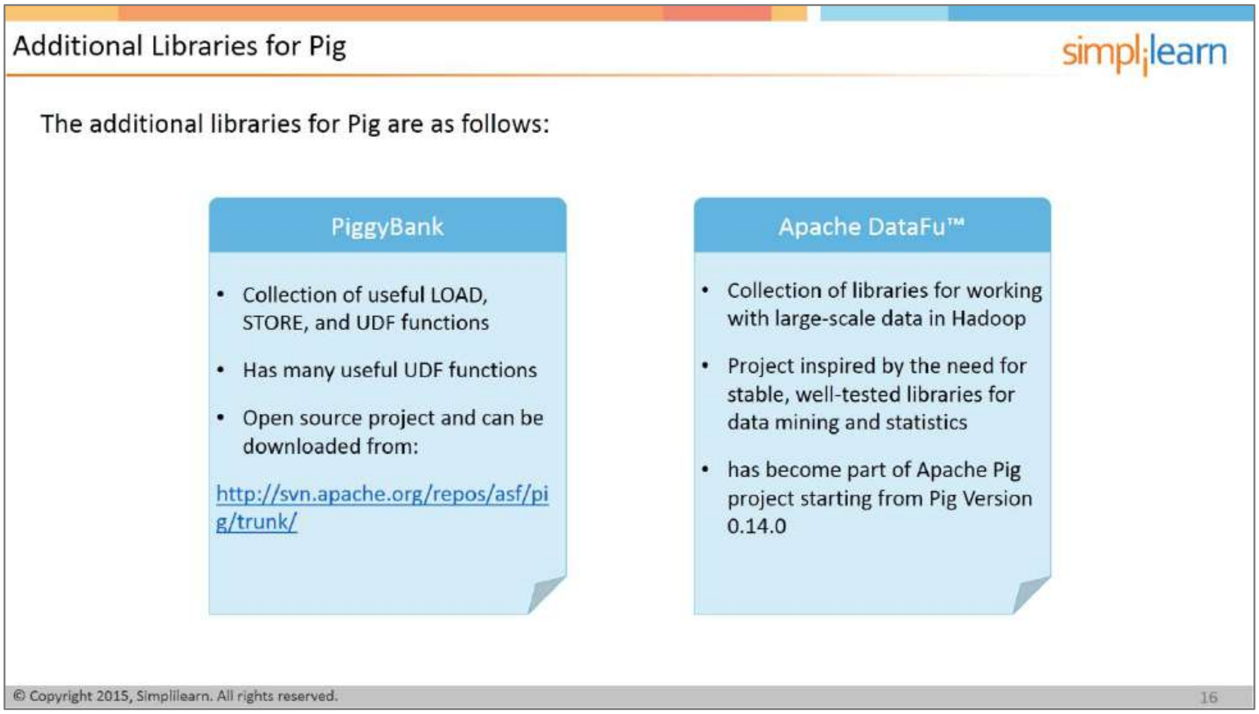
SQL	Pig

SELECT c_id , SUM(amount) AS CTotal
FROM customers c
JOIN sales s ON c.c_id = s.c_id
WHERE c.city = 'Texas'
GROUP BY c_id
HAVING SUM(amount) > 2000
ORDER BY CTotal DESC
	customer = LOAD '/data/customer.dat' AS (c_id,name,city);
	sales = LOAD '/data/sales.dat' AS (s_id,c_id,date,amount);
	salesBLR = FILTER customer BY city == 'Texas';
	joined= JOIN customer BY c_id, salesTX BY c_id;
	grouped = GROUP joined BY c_id;
	summed= FOREACH grouped GENERATE GROUP,
	SUM(joined.salesTX::amount);
	spenders= FILTER summed BY $1 > 2000;
	sorted = ORDER spenders BY $1 DESC;
	DUMP sorted;

Pig vs. SQL – –Example

The illustration given is an example to help you understand the SQL command and its Pig equivalent command script. The SQL command focuses on the customer table with columns c_id and CTotal, which is the sum of the amounts. It joins the sales table with reference to c_id, where the c.city is Texas. The grouping of c_id is performed by ensuring the sum of the amounts is greater than 2000 ordered in descending order.

Now, examine the same function using Pig. In Pig, you create two entities—customer and sales—where you load equivalent data with the schema. You filter the customers based on location, for example, Texas. Both data are joined together using the c_id row. The sum of the amounts of individual c_ids is calculated. Isolate those customers who spend more than $2,000. Later, sort the customers in descending order.



Additional Libraries for Pig

The additional libraries for Pig are as follows:

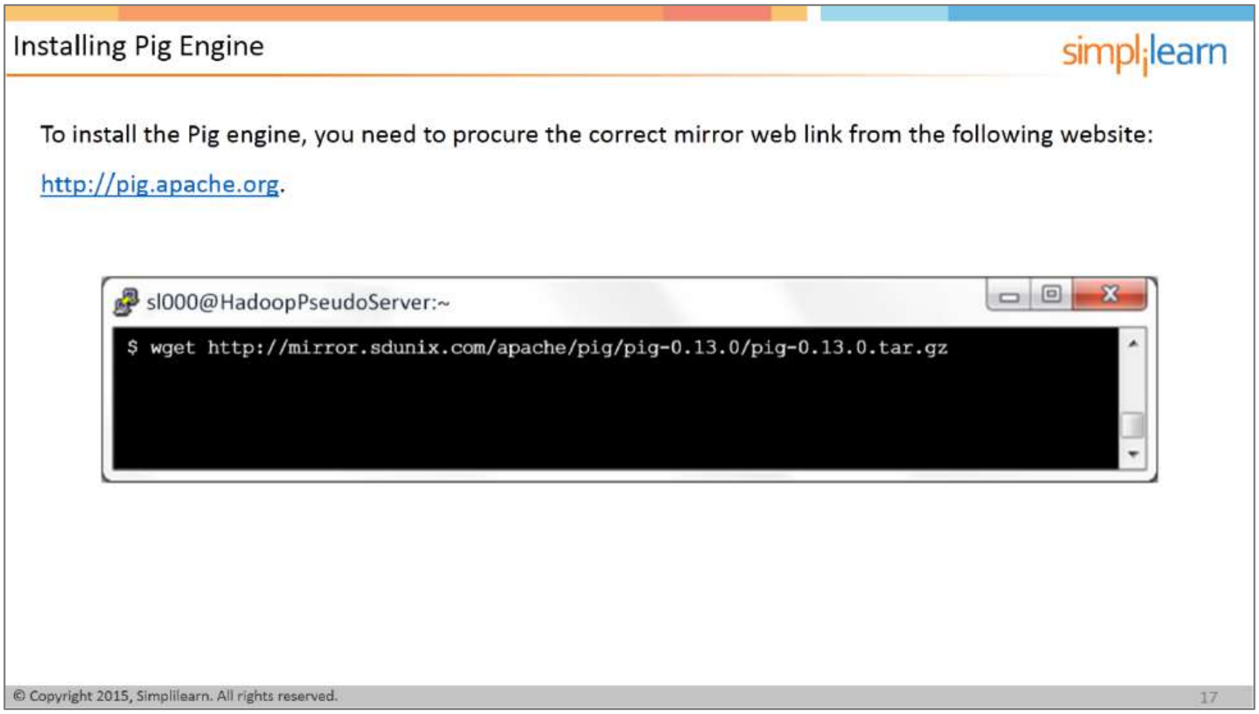
PiggyBank is a collection of useful LOAD, STORE, and UDF functions.

PiggyBank has many useful UDF functions for performing string, mathematical and storage operations.

It is open source project and can be downloaded from link shown.

Apache DataFu™ is a collection of libraries for working with large-scale data in Hadoop. The project was inspired by the need for stable, well-tested libraries for data mining and statistics.

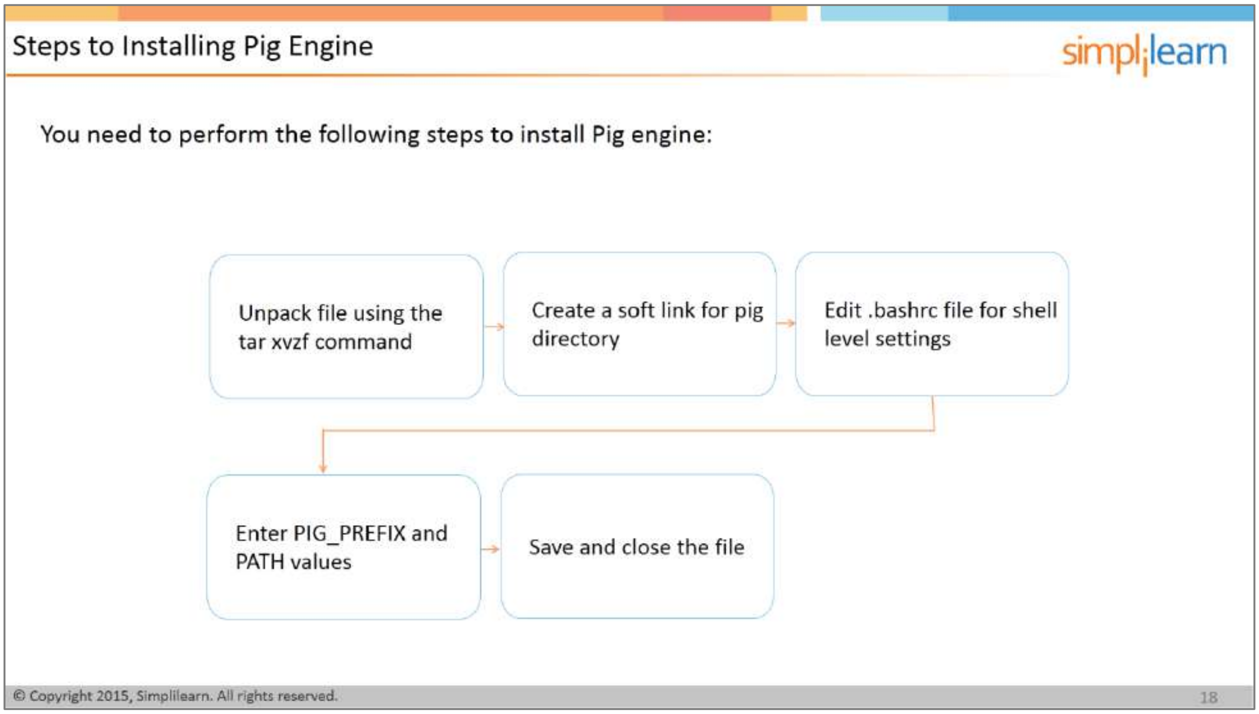
Apache DataFu has become part of Apache Pig project starting from Pig Version 0.14.0



Installing Pig Engine

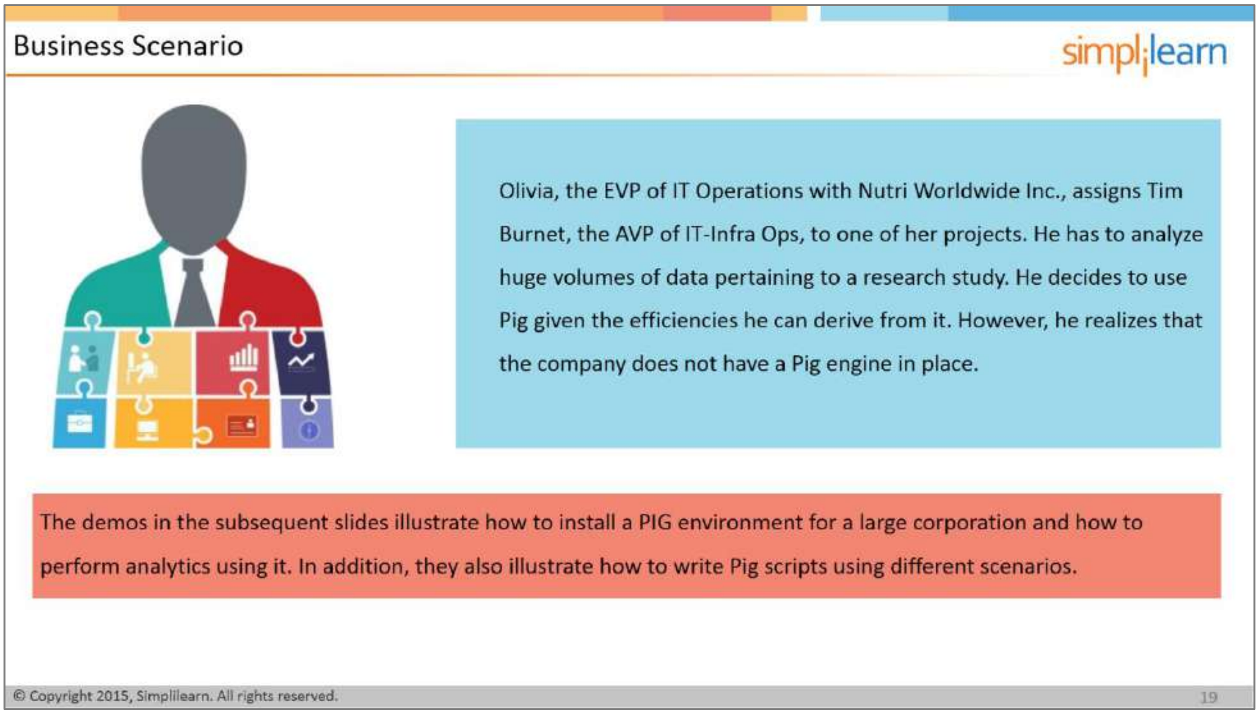
To install the Pig engine, you need to acquire the correct mirror web link from the URL.

Use the command as shown to download the Pig 0.13.0 tar file, and store it in the server system using the wget command.



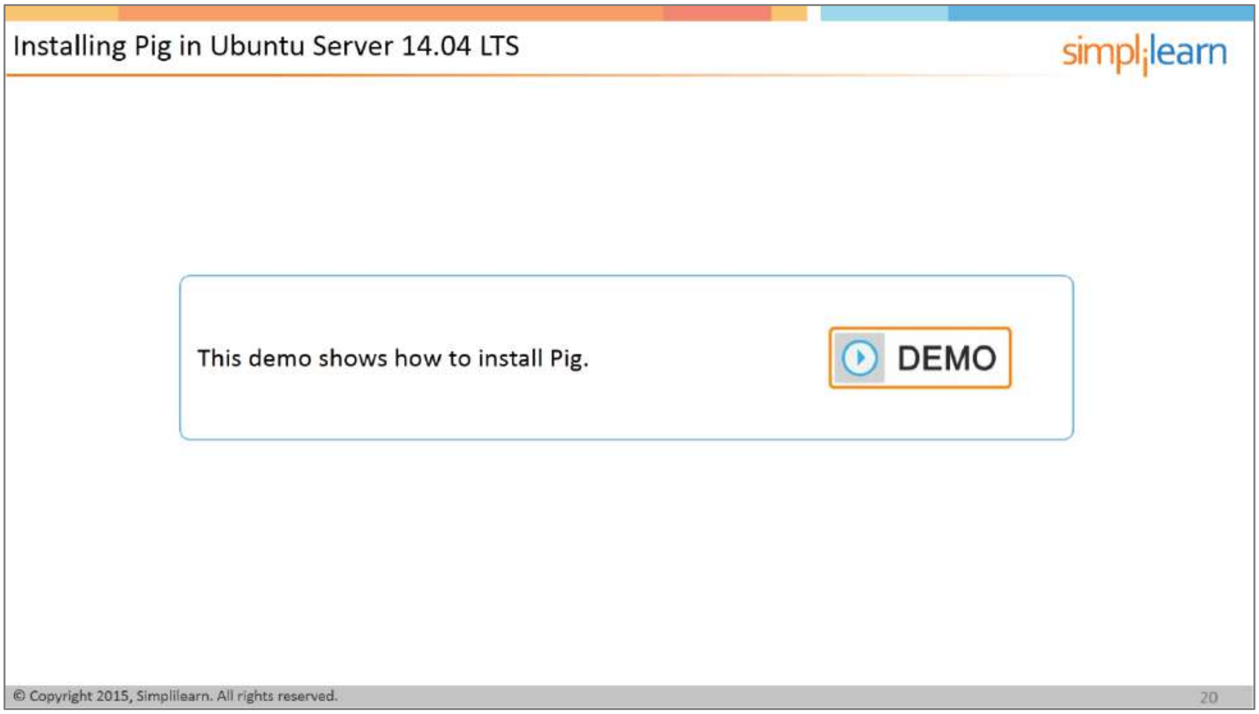
Steps to Installing Pig Engine

You need to perform the following steps to install the Pig engine. Unpack file using the tar xvzf command. Create a soft link for pig directory. Edit .bashrc file for shell level settings. Enter PIG_PREFIX and PATH values. Save and close the file.



Business Scenario

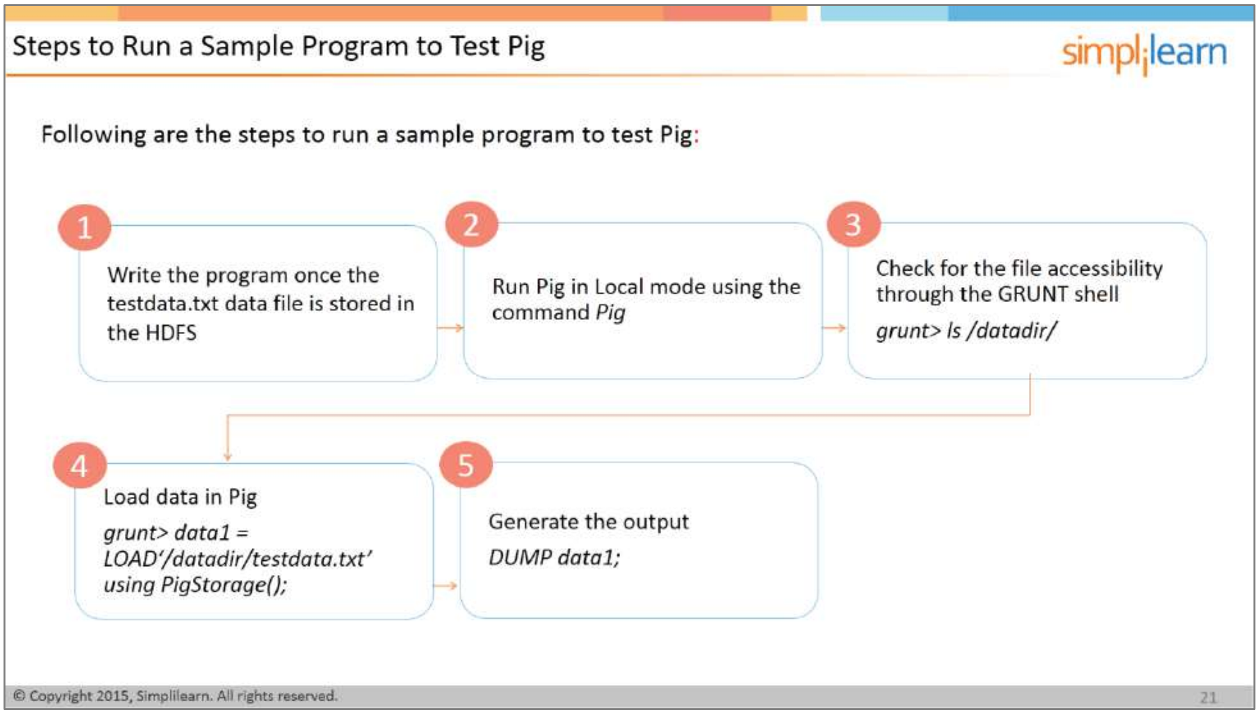
Olivia, the EVP of IT Operations with Nutri Worldwide Inc., assigns Tim Burnet, the AVP of IT-Infra Ops, to one of her projects. He has to analyze huge volumes of data pertaining to a research study. He decides to use Pig given the efficiencies he can derive from it. However, he realizes that the company does not have a Pig engine in place.



Installing Pig in Ubuntu Server 14.04

1. To begin Pig installation, let's first download Pig.
2. To download Pig you can visit the URL given. You can also visit Pig's website on Apache to download
3. Click on the nearest mirror site.
4. Select 'latest' release
5. Right-click binary tar file to copy the link address.
6. Note the location of the link.
7. Download Pig using wget command.
8. Download time may vary depending on the network speed.
9. Use the ls -l command to verify download.
10. Note the pig downloaded file.
11. Unpack file using the tar xvzf command.
12. List the directory contents to verify the unpacked files.
13. Notice a pig directory is created.
14. Create a soft link for pig directory at the location shown.
15. Verify link by listing it.
16. Edit .bashrc file for shell level settings.
17. Enter PIG_PREFIX and PATH values.
18. Save the file.
19. Close the file.

20. Refresh shell using the command shown.
21. Use pig command at shell prompt to launch pig.
22. Note that there is a grunt prompt which indicates that you have launched the pig shell successfully.
23. Close pig shell.
24. Type jps to see currently running java process.
25. Note the current java process running using jps command.
26. Open mapred-site.xml to enter MR job history configuration. Pig relies on this to run jobs.
27. Enter more property values related to job history server.
28. Enter the server or IP address and the port number.
29. Save the file.
30. Close the file.
31. Start MapReduce job history server.
32. Verify using jps command.
33. Note the job history server in list of processes.
34. Launch pig
35. Let's run some pig script. First copy a file to HDFS using Hadoop FS commands directly from grunt shell. Don't prefix with hadoop fs.
36. Use ls command to verify the file has been copied.
37. Same steps can be performed using hadoop fs command with arguments from shell prompt outside grunt.
38. Load file in a pig relation.
39. Process the file using pig transformations.
40. Dump the result on screen.
41. Note the result indicates the successful setup and running of pig in the Hadoop cluster.



Steps to Run a Sample Program to Test Pig

Let's look at the steps to run a sample program to test Pig.

- Once you upload the data in the Hadoop server in pseudo-distributed mode or single-node server, write the program. Save the data as testdata.txt.
- Run Pig in Local mode using the command **Pig.**
- Check for the file accessibility through the GRUNT shell using the command shown.
- Load data in Pig using the command shown.
- Generate the output using the command shown.

Getting Datasets for Pig Development

Use the following URLs to download different datasets for Pig development:

Datasets	URL
Books	www.gutenberg.org (war_and_peace.txt)
Wikipedia database	http://dumps.wikimedia.org/enwiki/
Variant datasets	www.infochimps.com/datasets
Open data base from Amazon S3 data	http://aws.amazom.com/datasets
Open database from National climate data	http://cdo.ncdc.noaa.gov/qclcd_ascii/

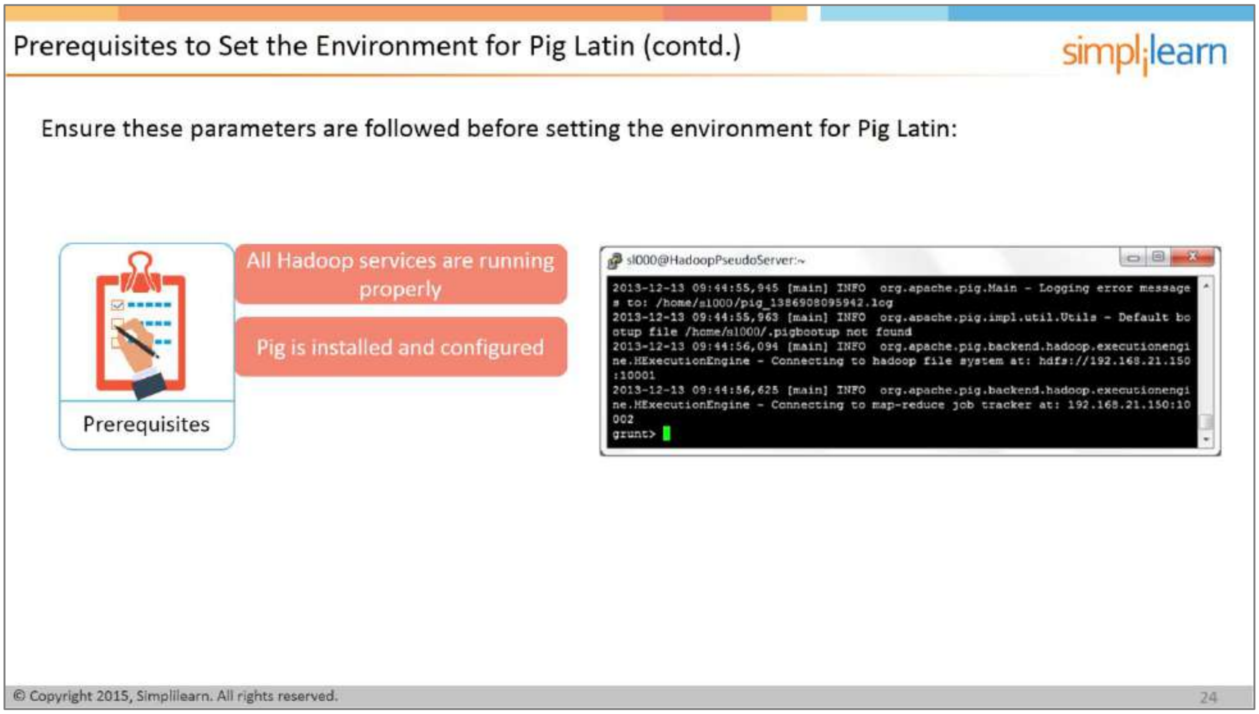
Getting Datasets for Pig Development

Use the URLs shown to download different datasets for Pig development.

Prerequisites to Set the Environment for Pig Latin

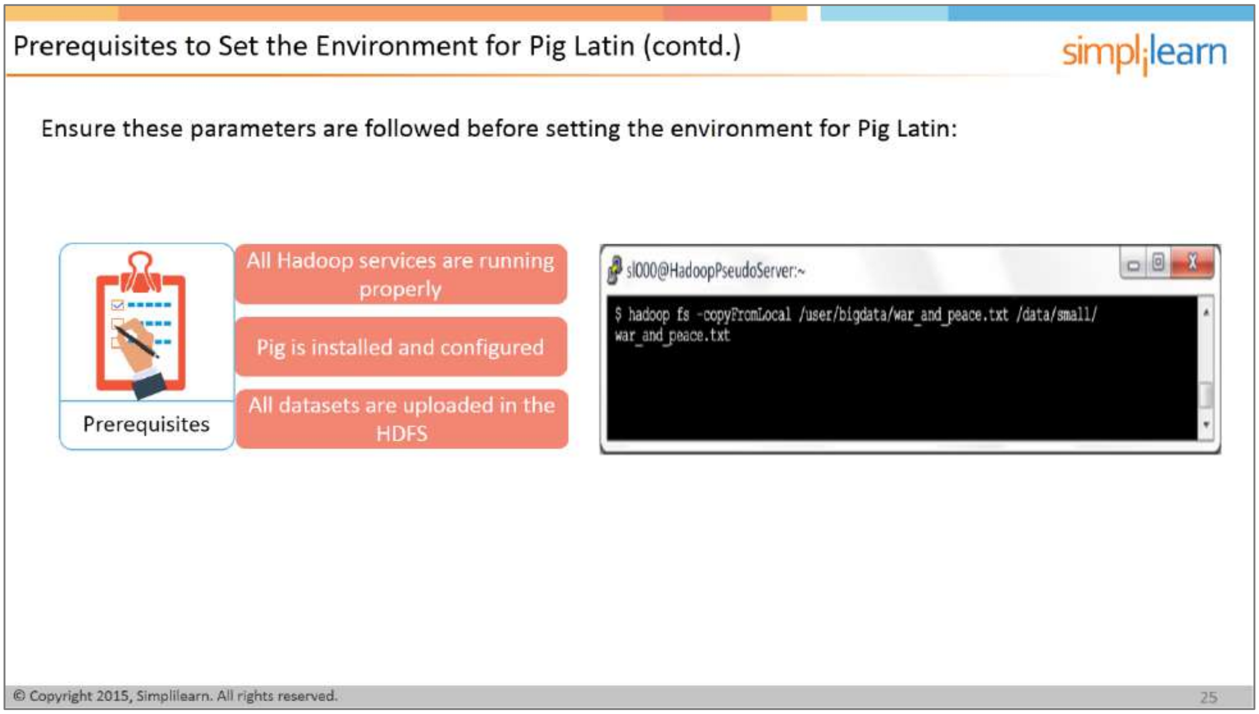
Ensure these parameters are followed before setting the environment for Pig Latin:

All Hadoop services are running properly. This can be checked by typing *jps* in the command console.



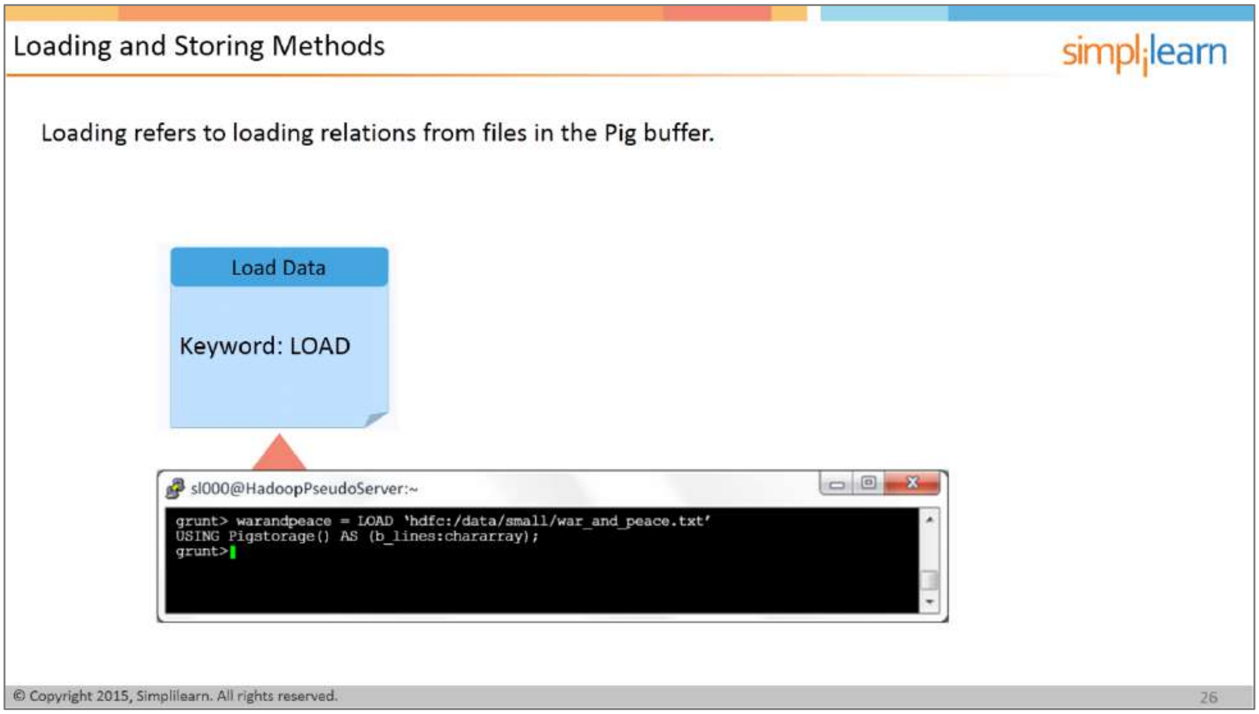
Prerequisites to Set the Environment for Pig Latin

Pig is installed and configured by running Pig in MapReduce mode.



Prerequisites to Set the Environment for Pig Latin (contd.)

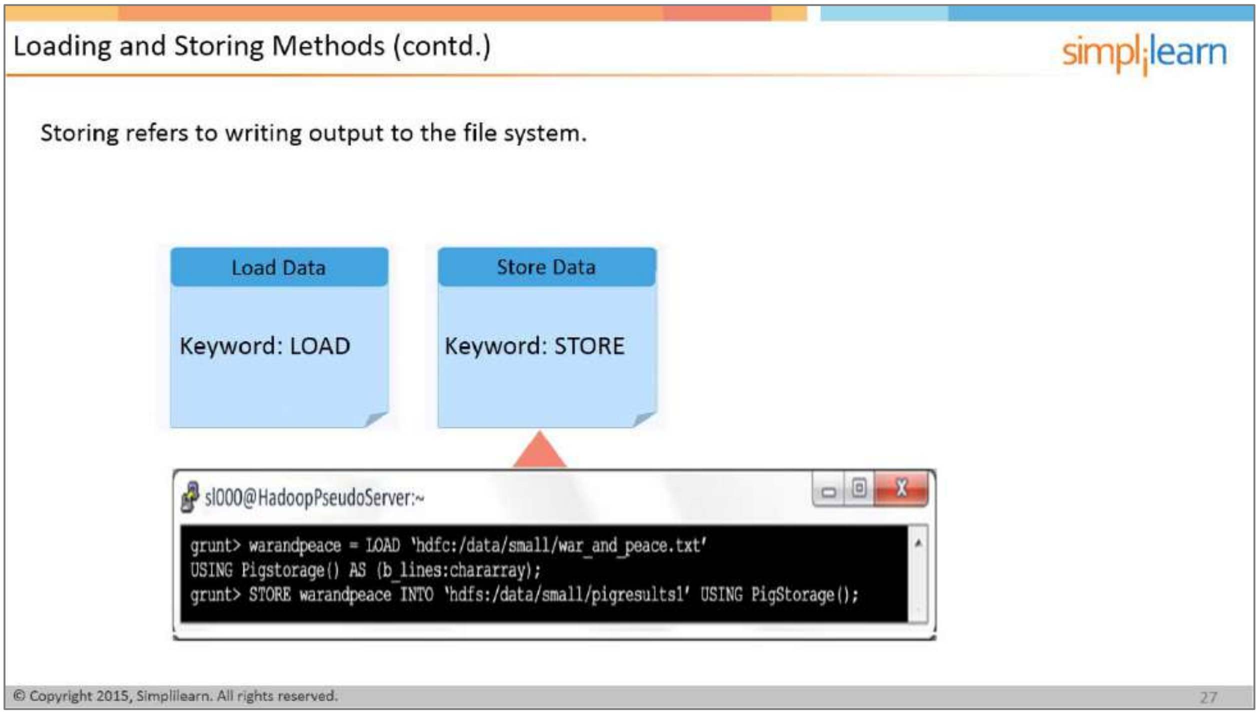
All required datasets are uploaded in the HDFS using the command shown



Loading and Storing Methods

Let's now look at how to load and store data in the Pig engine using the command console.

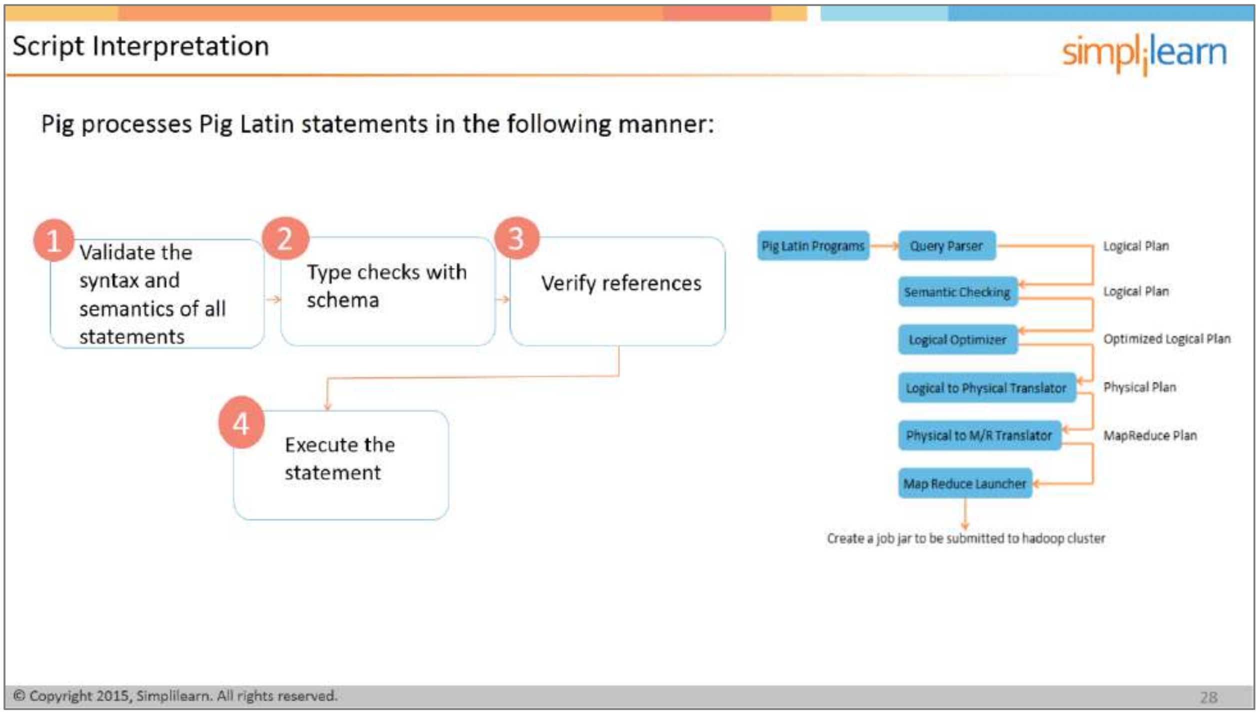
Loading refers to loading relations from the files in the Pig buffer. This is done using the keyword LOAD followed by the name of the variable whose data is to be loaded. A series of 'transformation' statements processes the data.



Loading and Storing Methods (contd.)

Storing refers to writing output to the file system. This is done using the keyword STORE followed by the name of the variable whose data is to be stored and the location of storage.

You can use the keyword DUMP to display the output on the screen.

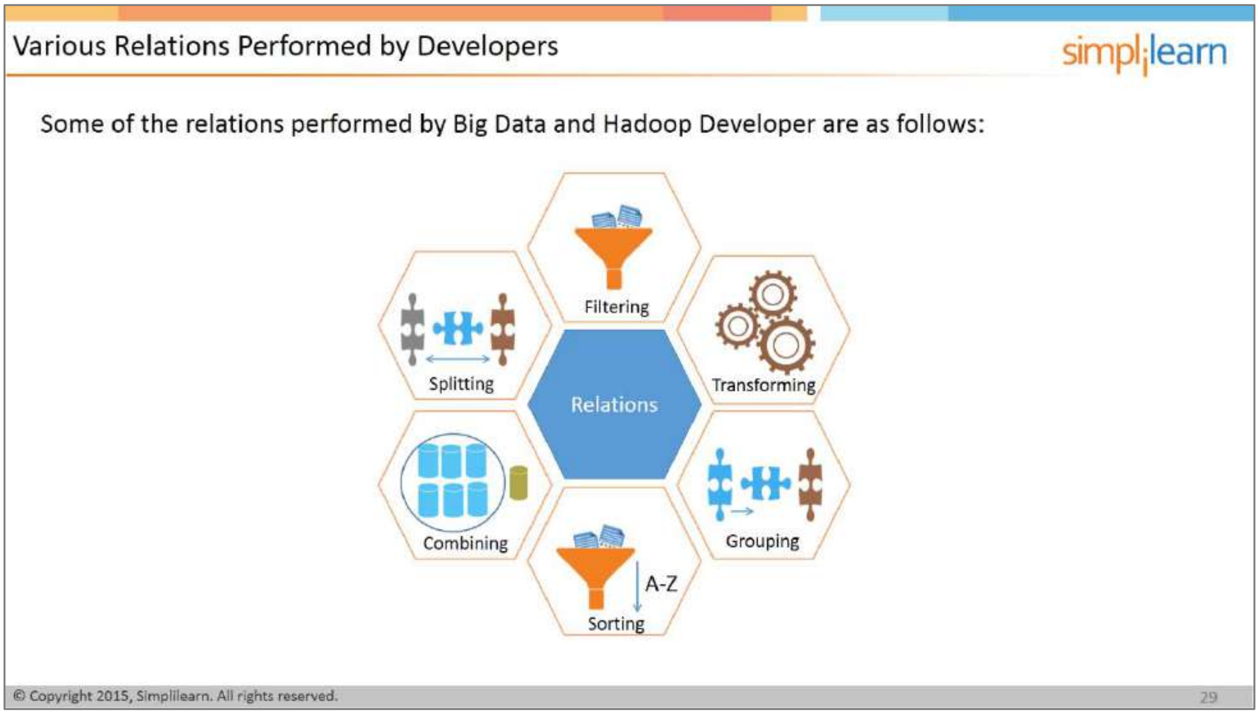


Script Interpretation

Pig processes Pig Latin statements in the following manner:

- First, Pig validates the syntax and semantics of all statements.
- Second, it type checks with schema.
- Third, it verifies references. Pig performs limited optimization before execution.
- Fourth, if Pig encounters a DUMP or STORE, it will execute the statements.

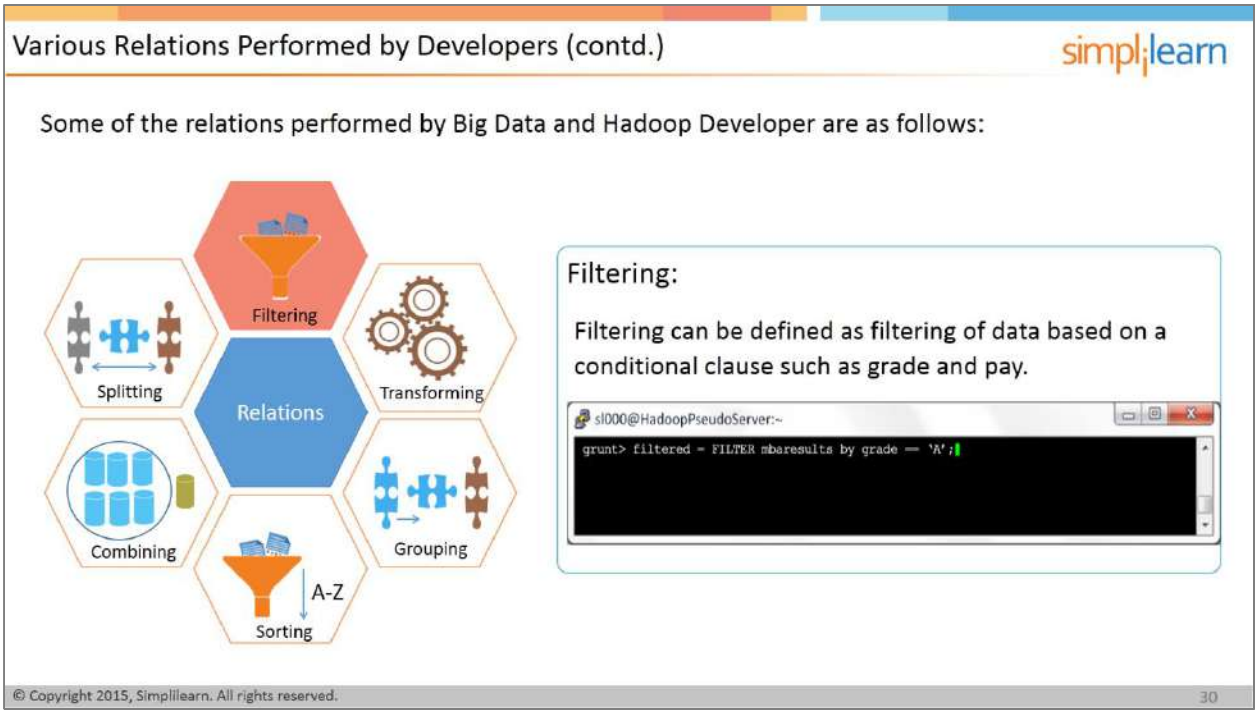
A Pig Latin script execution plan consists of the logical, optimized logical, physical, and MapReduce plan.



Various Relations Performed by Developers

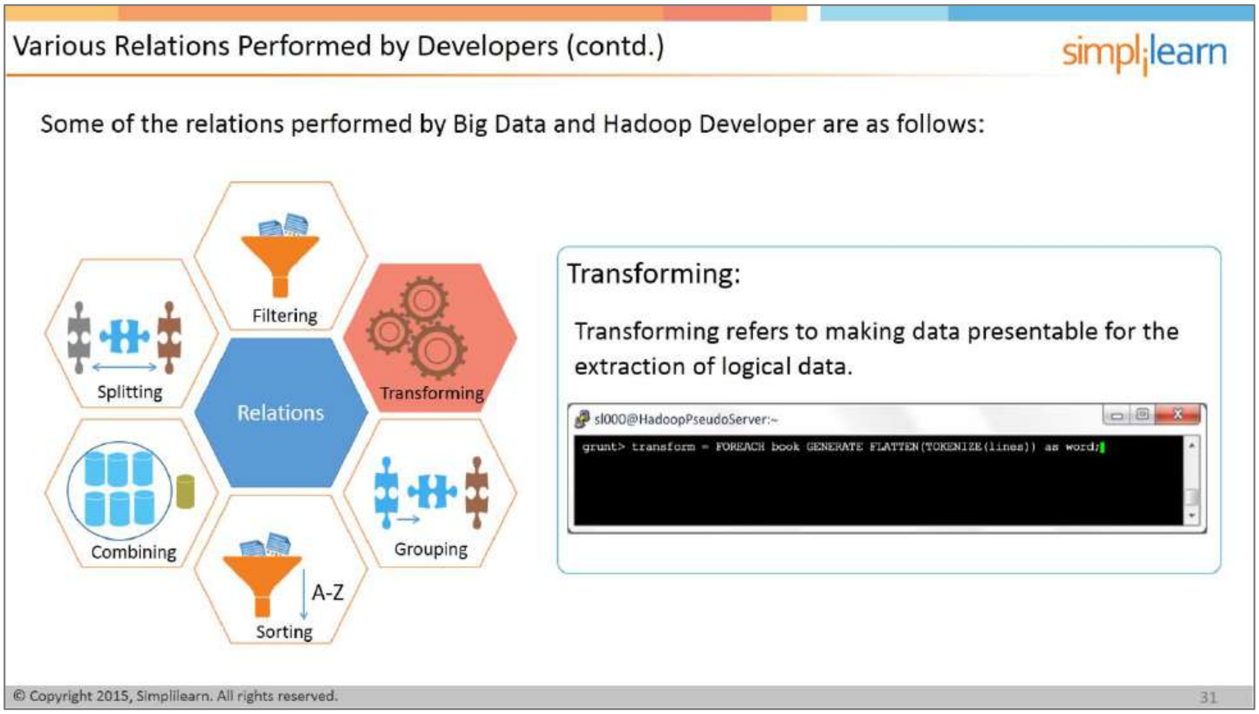
Some of the various relations performed by Big Data and Hadoop Developer are as follows:

- Filtering
- Transforming
- Grouping
- Sorting
- Combining
- Splitting



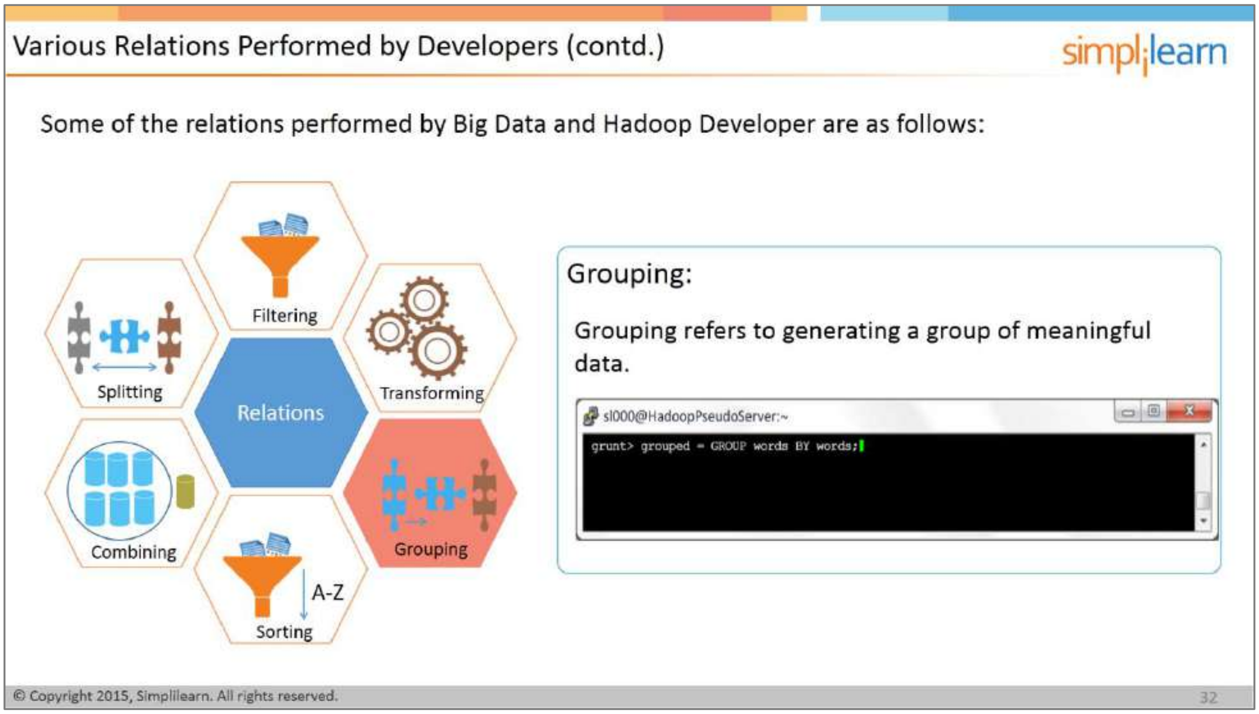
Various Relations Performed by Developers (contd.)

Filtering can be defined as filtering of data based on a conditional clause such as grade and pay.



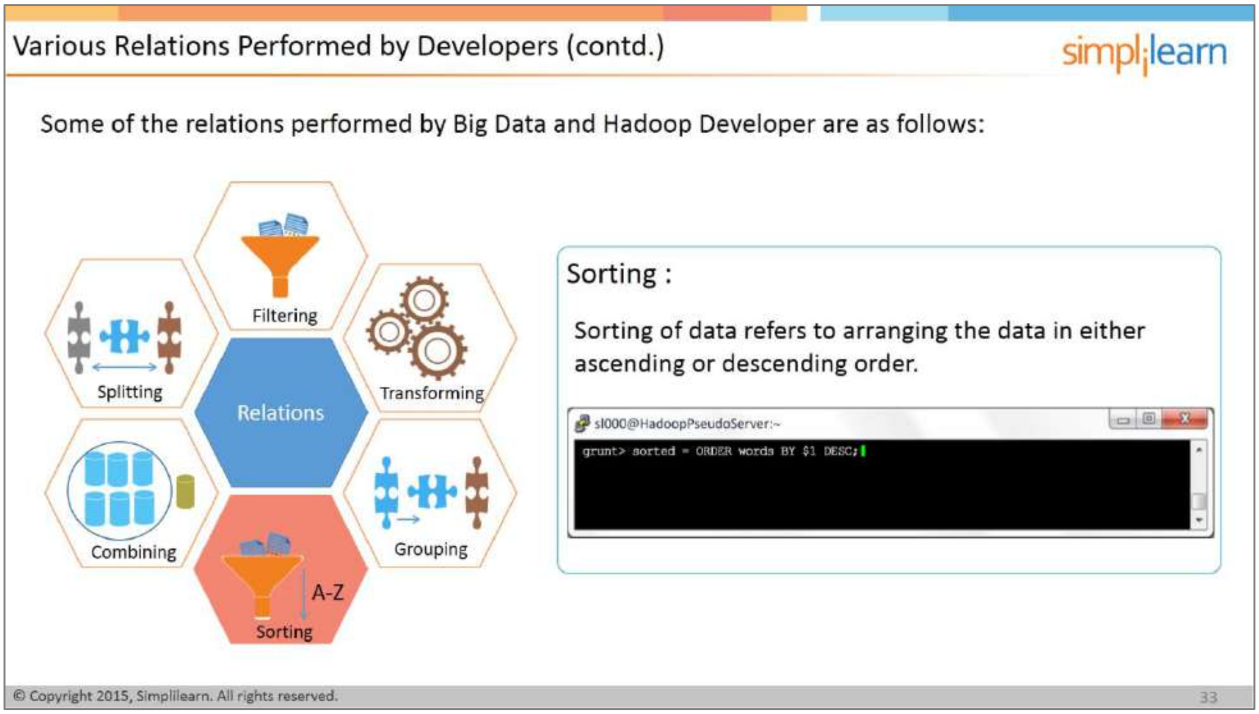
Various Relations Performed by Developers (contd.)

Transforming refers to making data presentable for extracting logical data.



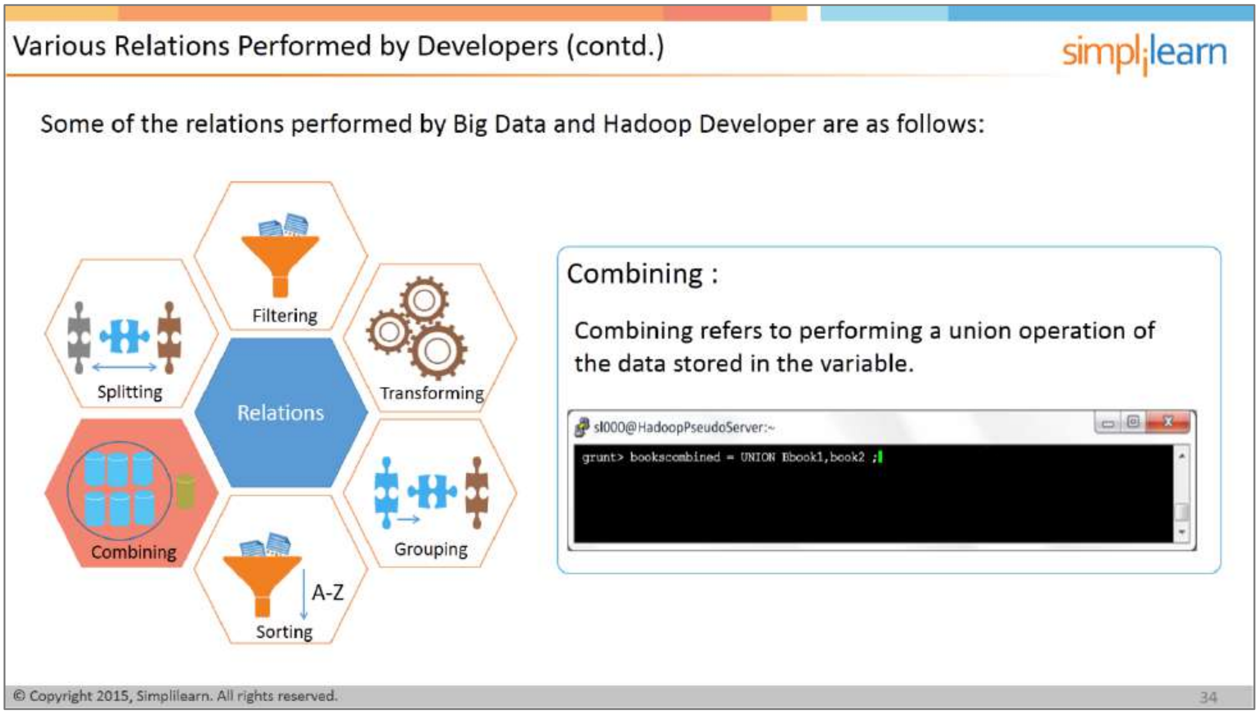
Various Relations Performed by Developers (contd.)

Grouping refers to generating a group of meaningful data.



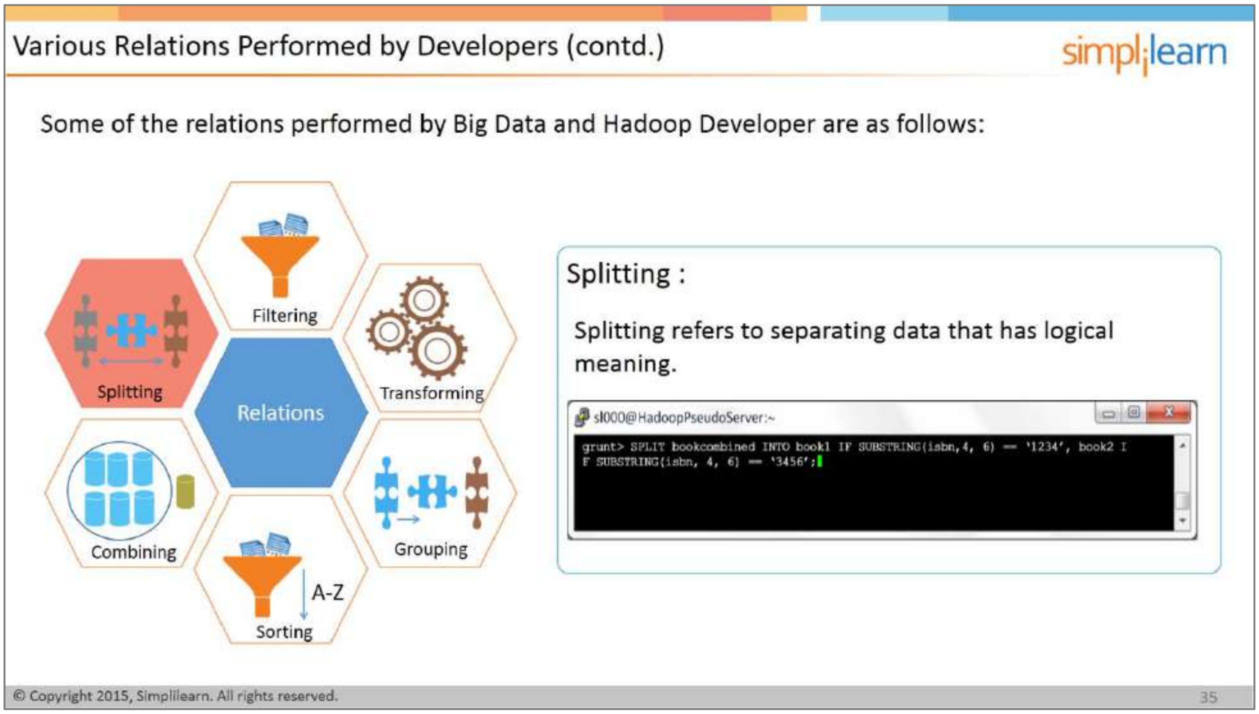
Various Relations Performed by Developers (contd.)

Sorting refers to arranging the data in ascending or descending order.



Various Relations Performed by Developers (contd.)

Combining refers to performing a union operation of the data stored in the variable.



Various Relations Performed by Developers (contd.)

Splitting refers to separating the data with a logical meaning.

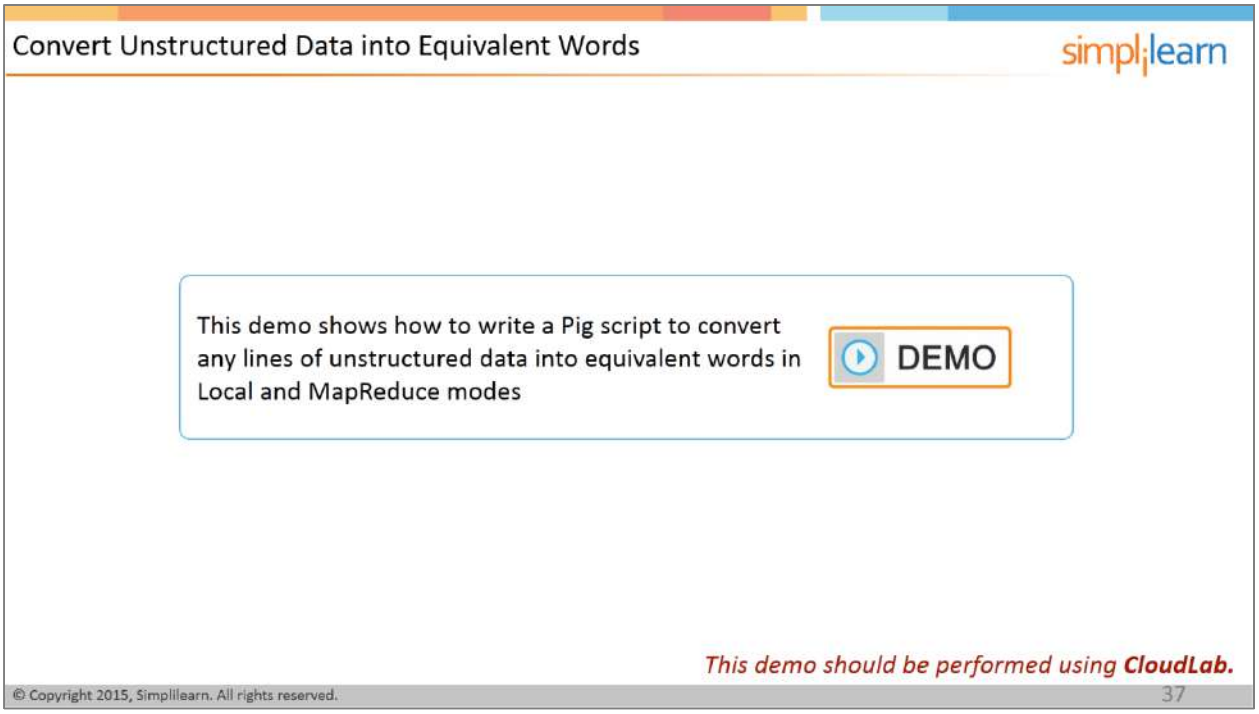
Various Pig Commands

Following are some of the Pig commands:

Pig command	What it does
Load	Reads data from file system
Store	Writes data to file system
foreach	Applies expression to each record and outputs one or more records
filter	Applies predicate and removes records that do not return true
group/cogroup	Collects records with the same key from one or more inputs
Join	Joins two or more inputs based on a key
Order	Sorts records based on a key
Distinct	Removes duplicate records
Union	Merges two data sets
Split	Splits data into 2 or more sets, based on filter conditions
Stream	Sends all records through a user-provided binary
Dump	Writes output to stdout
limit	Limits the number of records

Various Pig Commands

Some of the Pig Commands are given.

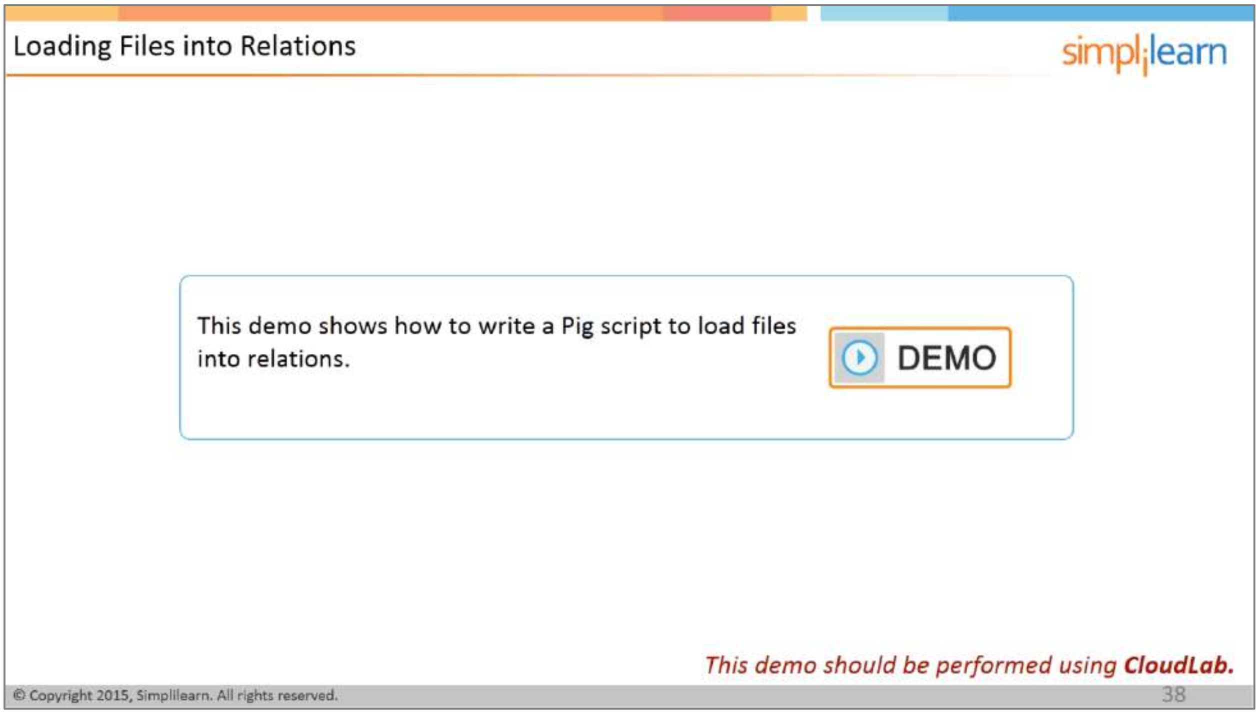


Convert Unstructured Data into Equivalent Words

This demo shows how to write a Pig script to convert any lines of unstructured data into equivalent words in Local and MapReduce modes.

1. Ensure that you have your sample data in the system.
2. Initially, run the program in MapReduce mode. Upload the data in HDFS using the command shown.
3. To start pig in MapReduce mode, the command used is pig.
4. Type pig.
5. Since you need to specify the location of the data in the program, ensure that your data is on the location.
6. The first line of code is responsible to load the data from HDFS to a temporary Pig storage for processing. The keyword LOAD is responsible to pick the data from the specified location. PigStorage() is the temporary data location such as a cache which will be used during the processing period. Variable named lines will be used with a character array data type, chararray. Ensure that you add a semi colon at the end of statement.
7. The next line will convert lines into words. This is done using a looping statement, FOREACH which will GENERATE words by FLATTENing the lines, which means removing all characters other than 0 to 9, A to Z, and a to z. TOKENIZE will ensure that each word will be a separate entity which will be stored in an array as a word.
8. List the output using the keyword DUMP. The words generated are seen which are stored in variable 'words'. Type dump words;

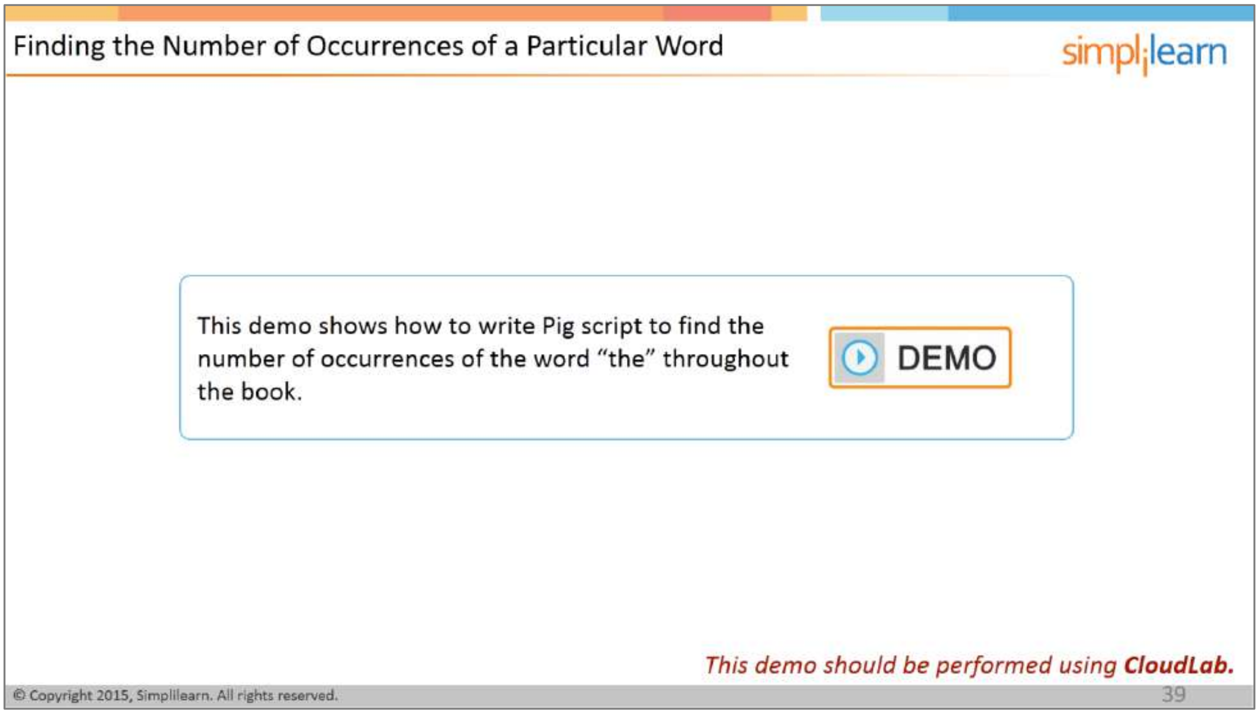
9. The highlighted part shows the output that is generated and stored in the variable words.
10. Let's try the same program in the local mode.
11. Start pig in local mode using the command pig -x local.
12. Type the same line to load the data with a difference that the data is loaded from ext file system and not from the HDFS..
13. The next line converts lines into words. This is done using a looping statement, FOREACH, which will GENERATE words by FLATTENing the lines which means removing all characters other than 0 to 9, A to Z and a to z. TOKENIZE will ensure that each word will be a separate entity which will be stored in array as word.
14. List the output using the keyword DUMP. Type dump words;
15. The words generated are seen which are stored in variable 'words'.
16. The highlighted part shows the output of the program.
17. Note, it states that the program is run in local mode using local resources.



Loading Files into Relations

This demo shows how to write Pig script to load files into relations.

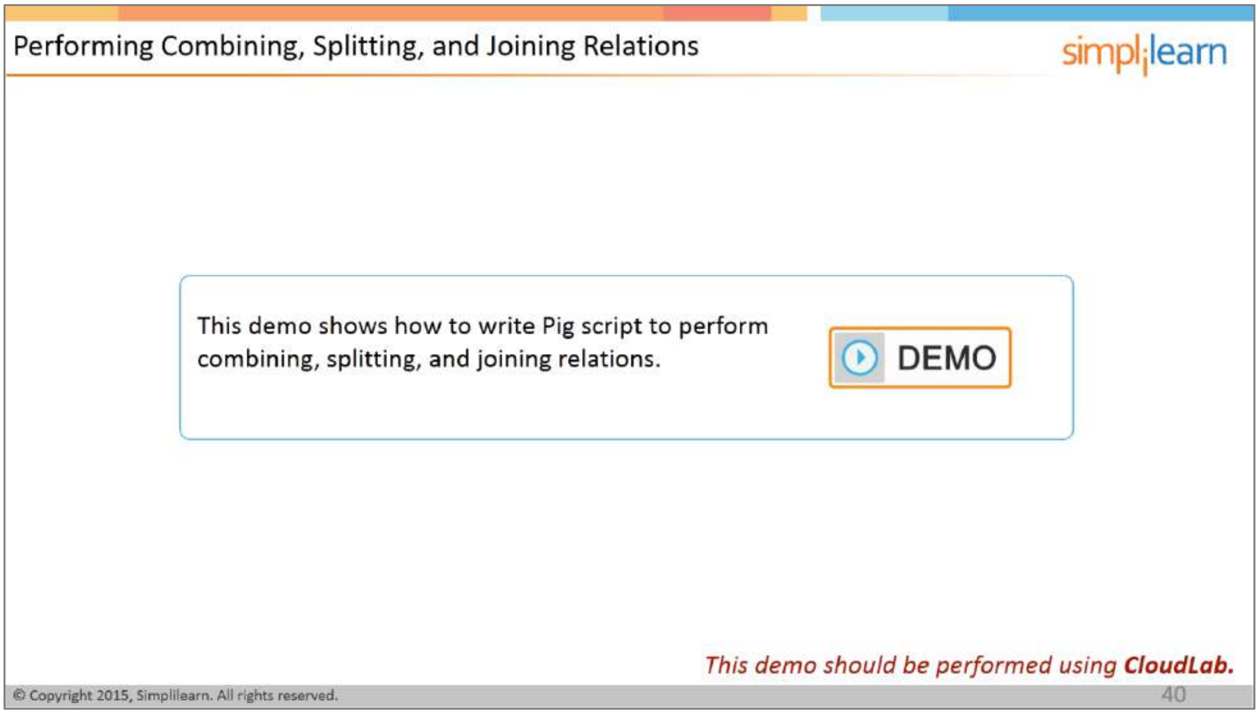
1. First, create a sample data set called login dot txt.
2. Enter records in the form of user, date-time stamp.
3. Each record is present in a new line and different data elements are separated by a comma.
4. Save and exit.
5. Use the cat command to view the file contents.
6. Copy the file to HDFS. In this case, copy it to a learning directory.
7. Launch Pig.
8. If pig is already installed, enter pig on command prompt.
9. Once the table is created, enter a row in the table with values like 1 comma USA.
10. You will see grunt. Pig's command line interface is grunt.
11. LOAD login.txt in user Login relation with user and date-timestamp separated by a comma. Specify user as 'char' array datatype and date-timestamp as 'long'. Notice the usage of handler Pig Storage along with which you specify comma delimiter.
12. DUMP user Login on screen to verify whether the loading is successful.
13. 'DUMP' and 'Store' in Pig initiate MapReduce job, firing off as lazy execution a series of steps covered in the Pig script preceding 'Dump' or 'Store'.
14. Once the MapReduce job is complete, the output is shown.



Finding the Number of Occurrences of a Particular Word

This demo shows how to write a write Pig script to find the number of occurrences of the word the throughout the book.

1. Start Pig in MapReduce mode.
2. Load testdata.txt using the command file =
3. Type the command to load testdata.txt file.
4. To convert the lines into words, use the command shown.
5. Create a group of words.
6. Aggregate and count the words.
7. Dump the output on the screen.
8. You can see the output of the program in the highlighted part. It shows each word with the number of occurrences.
9. Let's try the same program in the Local Mode.
10. Start pig in the local mode. Load testdata.txt.
11. Convert the lines to words.
12. You have to create a group of words.
13. Let's aggregate and count the words.
14. Dump the output on the screen. You can see the output of the program in the highlighted part. It shows each word with the number of occurrences.
15. You have successfully executed the program.

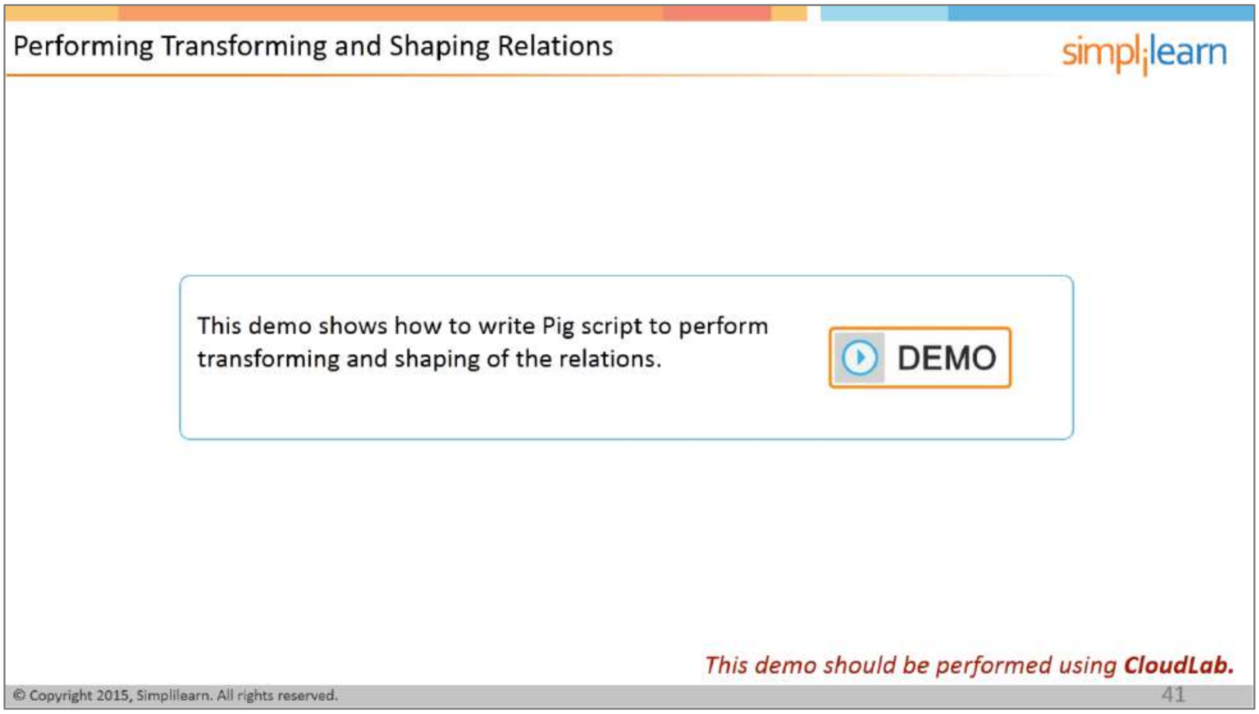


Performing Combining, Splitting, and Joining Relations

This demo shows how to write Pig script to perform combining, splitting, and joining relations.

1. Load the January data.
2. Load the February data.
3. Load the March data.
4. Load April data.
5. Let's perform combine operation in which the data of all four months will be combined.
6. Let's see the result of combining the data.
7. Execution is done successfully. Check for success line.
8. Let's see the output. Go to NameNode GUI. The address is: http://192.168.21.153:50070
9. Click the Browse the filesystem link.
10. Click the data link.
11. Click the Big link.
12. Click the pigresults link.
13. Click month_quad file.
14. The output is displayed.
15. Let's perform splitting operation.
16. Let's see the output of the split.
17. The program will execute successfully.
18. Let us check the output for split. Go to NameNode GUI. The address is http://192.168.21.153:50070

19. Click the Browse the filesystem link.
20. Click the data link.
21. Click the big link.
22. Click the pigresults4 link.
23. Click the jan link.
24. The list shows the output for the same.
25. You have successfully performed the program.



Performing Transforming and Shaping Relations

This demo shows how to write Pig script to perform transforming and shaping of the relations.

1. Let's first understand the program flow.
2. Load the data for months Jan, Feb, Mar, and Apr.
3. Perform the union operation of the months.
4. Perform the splitting operation of aggregated data.
5. Perform the filtering of data based on CLEARSKY parameter.
6. This line stores the output of the transformed table.
7. Let us execute this program in batch mode.
8. The program execution starts. Since the data is 2GB, it will take around 15 minutes to perform execution of the same.
9. Let us see the output for the same. Go to namenode GUI using the address http://192.168.21.153:50070
10. Click the data link.
11. Click the big link.
12. Click the pigresults5 link.
13. Click the shaped link.
14. The output is generated successfully.
15. Click anyone file to view the output.
16. Thus we have performed the program successfully.

Spot the Error

Paul is working in ABX corporation Ltd., as a Hadoop developer. His manager asks him to install Pig engine. He follows the installation steps. However, the installation of Pig engine fails due to an error in the code. The Project Manager asks Paul to rectify this issue. Can you assist Paul in identifying his error? *Spot the incorrect code.*

sl000@HadoopPseudoServer:~

```
$ tar -xvzf pig-0.13.0.tar.gz
$ sudo mv pig-0.13.0 /usr/local/Pig
$ export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-i386
$ Import Pig_PREFIX=/usr/local/Pig
```


Spot the Error

The incorrect code is: `$ Import Pig_PREFIX=/usr/local/Pig`

The correct code is: `$ export Pig_PREFIX=/usr/local/Pig`

QUIZ 1

Which version of Pig engine has the MapReduce feature in Local mode?

a. Pig 0.5.0

b. Pig 0.6.0

c. Pig 0.4.0

d. Pig 0.8.0

QUIZ 2

Which of the following commands is used to start Pig in MapReduce mode?

a. Pig

b. Pig -x MapReduce

c. Pig -x local

d. Both Pig and Pig -x MapReduce

QUIZ 3

Which of the following commands is used to start Pig in Local mode?

a. Pig -x local

b. Pig -x MapReduce

c. Pig

d. Both b and c

 50

QUIZ 4

Which of the following keywords in Pig scripting is used for displaying the output on the screen?

a. DUMP

b. STORE

c. LOAD

d. TOKENIZE

 52

QUIZ 5

Which of the following keywords in Pig scripting is used for accepting input files?

a. LOAD

b. STORE

c. FLATTERN

d. TOKENIZE

 54

QUIZ 6

Which of the following syntaxes is used to perform loading of data from file to create relations in Pig?

a. LOAD

b. STORE

c. GROUP

d. FOREACH

 56

QUIZ 7

Which of the following syntaxes is used to perform storing of the output data to the files in Pig?

a. GROUP

b. STORE

c. LOAD

d. FOREACH

 58

QUIZ 8

Which of the following syntaxes is used to perform looping in Pig?

a. STORE

b. LOAD

c. GROUP

d. FOREACH

 60

QUIZ 9

Which of the following keywords is used to perform combining in Pig?

a. LOAD

b. STORE

c. UNION

d. FOREACH

QUIZ 10

Which of the following keywords is used to perform sorting in Pig?

a. LOAD

b. UNION

c. ORDER

d. FOREACH

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which version of Pig engine has the MapReduce feature in Local mode?	d. Pig 0.8.0 and later versions have a provision to perform MapReduce functions in Local mode.
2	Which of the following commands is used to start Pig in MapReduce mode?	d. Pig or Pig -x MapReduce command can be used to run the Pig in MapReduce mode.
3	Which of the following commands is used to start Pig in Local mode?	a. Pig -x local command is used to start Pig in Local mode.
4	Which of the following keywords in Pig scripting is used for displaying the output on the screen?	a. DUMP is used to display the output on the screen.
5	Which of the following keywords in Pig scripting is used for accepting input files?	a. LOAD in Pig scripting is used for accepting input files.

ANSWERS:

S.No.	Question	Answer & Explanation
6	Which of the following syntaxes is used to perform storing of the output data to the files in Pig?	a. LOAD in Pig scripting is used to perform loading of data from file to create relations in Pig.
7	Which of the following files is responsible for decommissioning the nodes in a Hadoop cluster?	b. STORE is used to perform storing of the output data to the files in Pig.
8	Which of the following syntaxes is used to perform looping in Pig?	d. FOREACH is used to loop operations in Pig.
9	Which of the following keywords is used to perform combining in Pig?	c. UNION in Pig scripting is used for combining data in Pig.
10	Which of the following keywords is used to perform sorting in Pig?	c. The keyword ORDER in Pig scripting is used for sorting data.

Case Study

Scenario | Analysis | Solution

XY Networks provides network security support to lot of organizations. It has system generated log files critical for security analysis and monitoring. The company has chosen to implement Hadoop to store the log files. But XY's data scientists are finding it difficult to use Hadoop MapReduce as they are not Java programmers. For any data analysis, they have to run to the Java developers to code the MapReduce programs. In case there are any mistakes, they have to send the programs back to developers. Data Scientists then find that they can use a Hadoop tool, Pig, which can use an interactive scripting language.

 67

Case Study

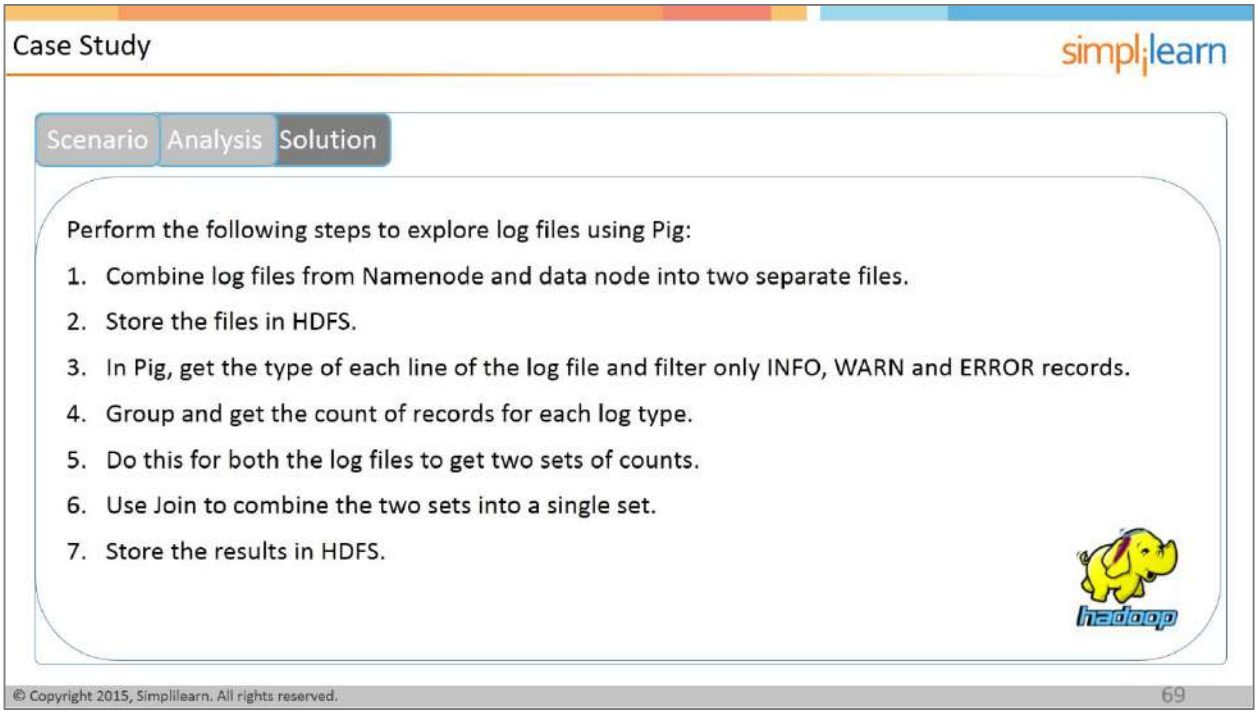
Scenario | Analysis | Solution

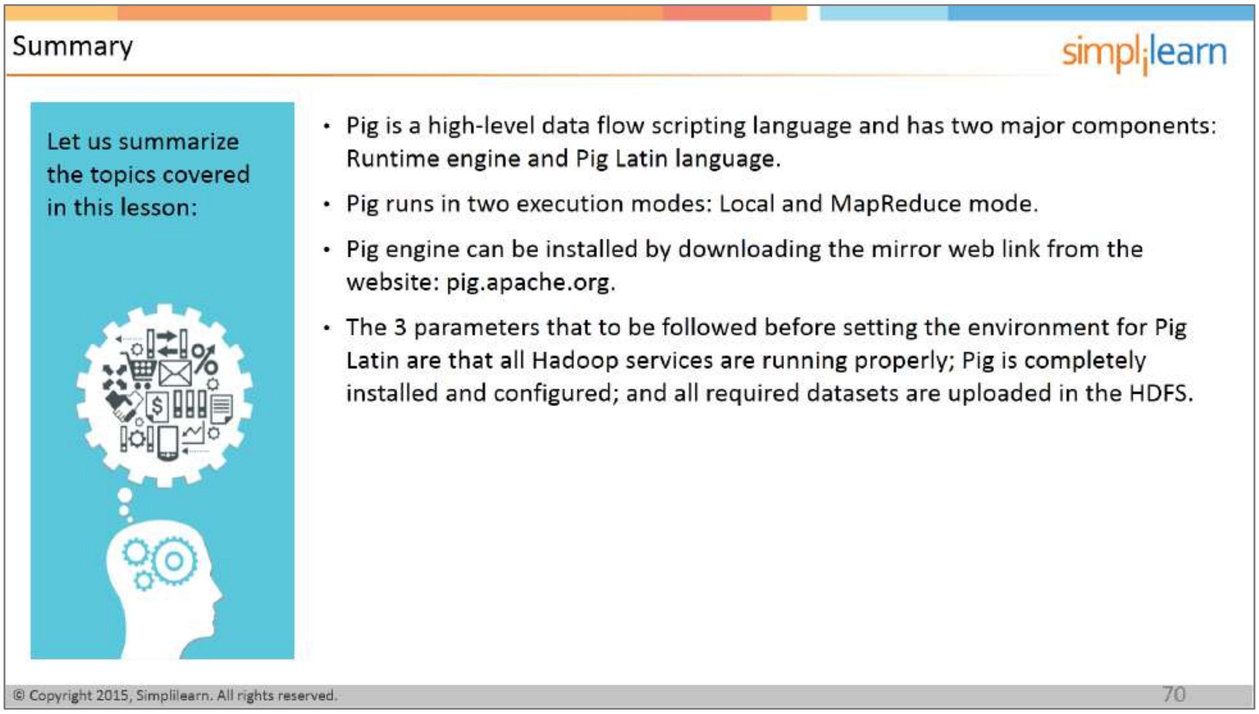
The team does research on Apache Pig and find that it is popular among data scientists to analyze Big Data. Also, Pig is suitable for analyzing unstructured data.

Advantages of Pig over using MapReduce:

1. No need for Java programming.
2. Easy-to-learn scripting language.
3. Can process unstructured data-like log files.
4. Provides joins of multiple sources of data which is difficult to do in MapReduce.
5. Can look at intermediate data while processing data.

 68

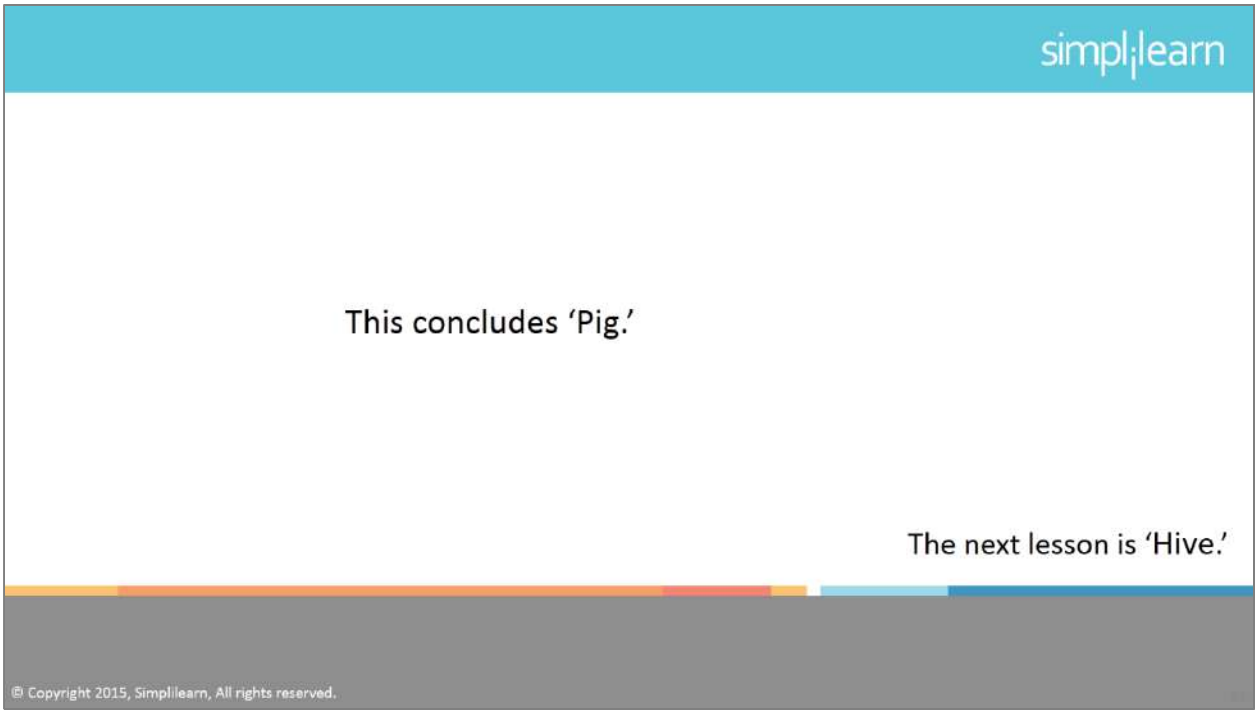

Case Study
simplilearn
Scenario
Analysis
Solution
Perform the following steps to explore log files using Pig:
1. Combine log files from Namenode and data node into two separate files.
2. Store the files in HDFS.
3. In Pig, get the type of each line of the log file and filter only INFO, WARN and ERROR records.
4. Group and get the count of records for each log type.
5. Do this for both the log files to get two sets of counts.
6. Use Join to combine the two sets into a single set.
7. Store the results in HDFS.
hadoop
© Copyright 2015, Simplilearn. All rights reserved.
69



Summary

Let us summarize the topics covered in this lesson:

- Pig is a high-level data flow scripting language and has two major components: Runtime engine and Pig Latin language.
- Pig runs in two execution modes: Local and MapReduce mode.
- Pig engine can be installed by downloading the mirror web link from the website: pig.apache.org.
- The 3 parameters that need to be followed before setting the environment for Pig Latin are that all Hadoop services are running properly; Pig is completely installed and configured; and all required datasets are uploaded in the HDFS.

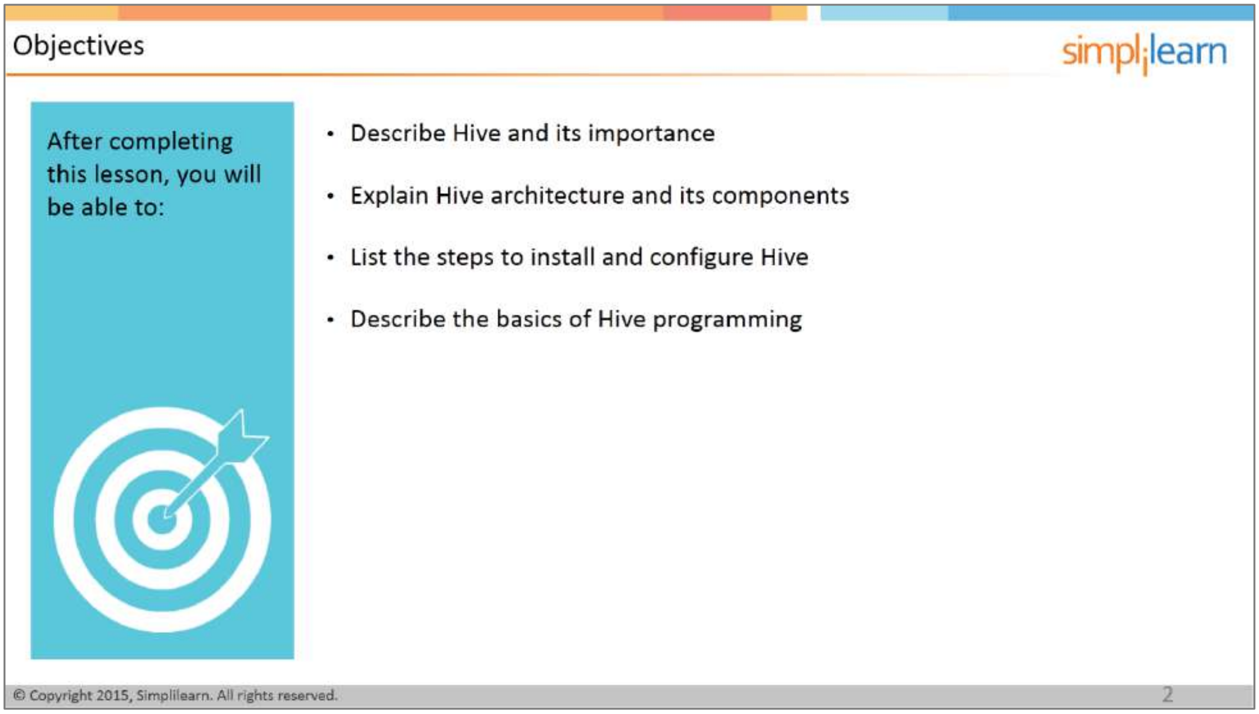


Conclusion

This concludes "Pig."

In the next lesson we will focus on "Hive."

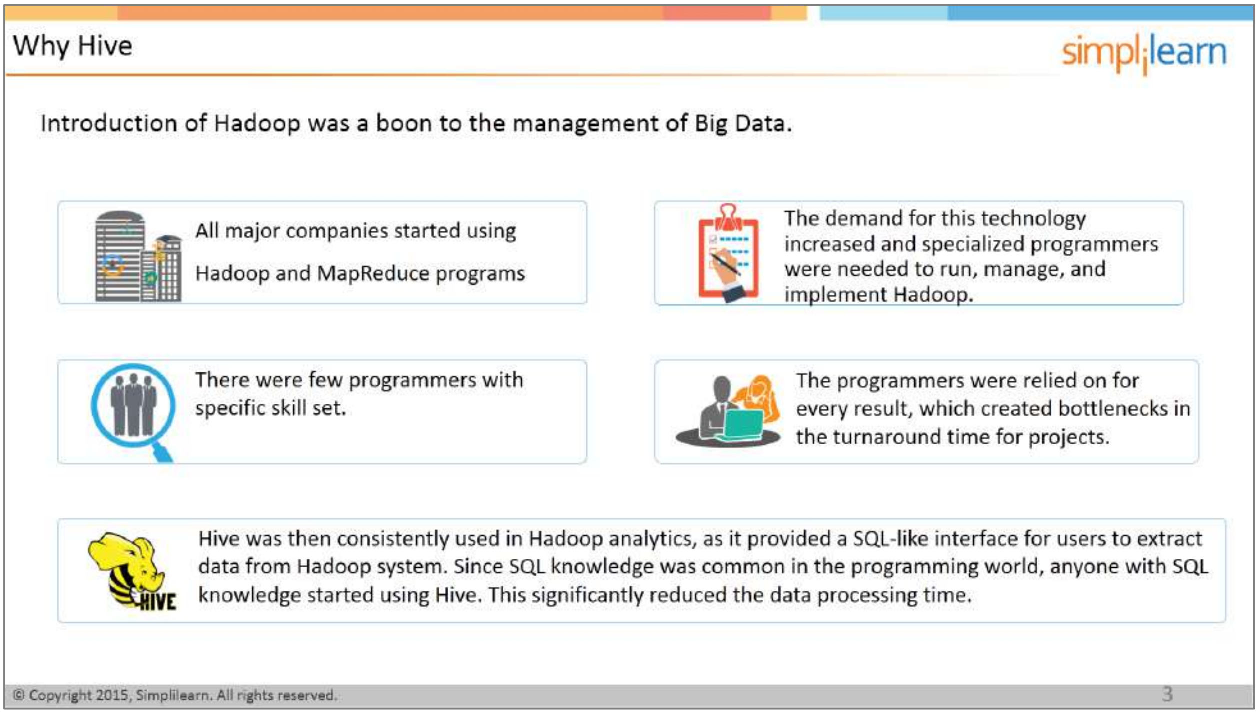
Lesson 7—Hive



Objectives

After completing this lesson, you will be able to:

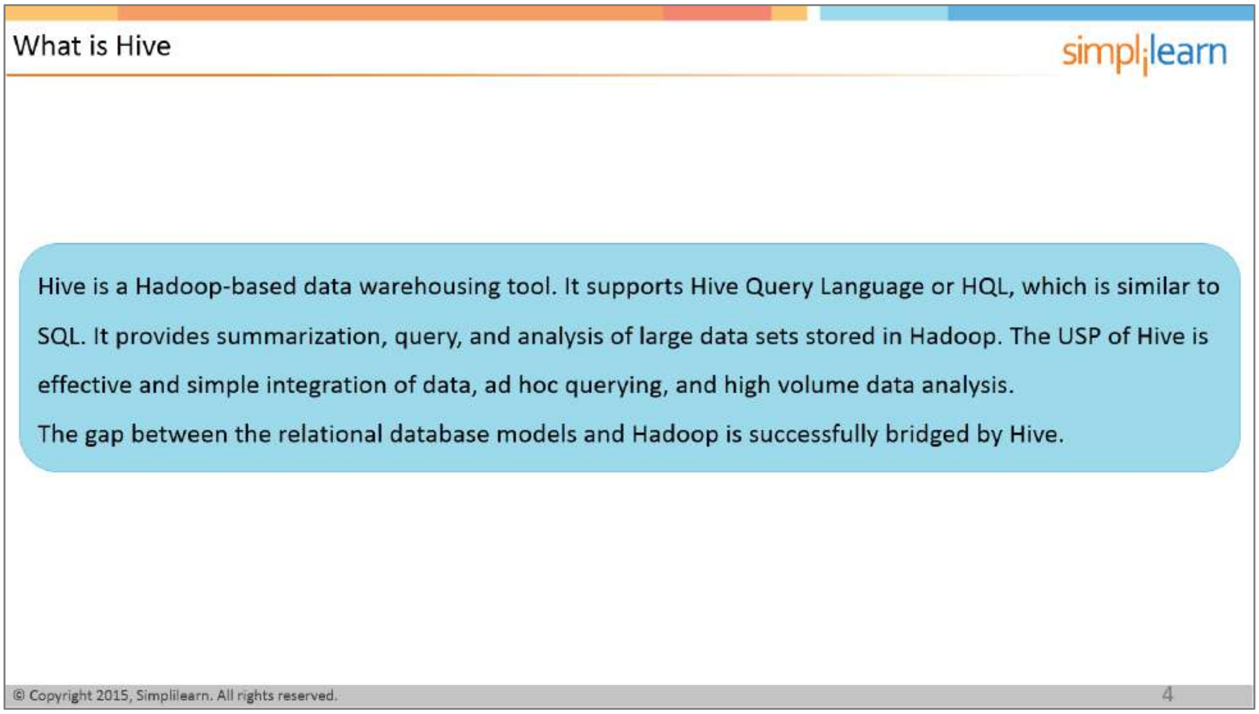
- Describe Hive and its importance
- Explain Hive architecture and its components
- List the steps to install and configure Hive
- Describe the basics of Hive programming



Why Hive

Introduction of Hadoop was a boon to the management of Big Data. Gradually, all major companies started using Hadoop and MapReduce programs. The demand for this technology increased and specialized programmers were needed to run, manage, and implement Hadoop. However, there were few programmers with specific skill set. The programmers were relied on for every result, which created bottlenecks in the turnaround time for projects.

Hive was then consistently used in Hadoop analytics, as it provided a SQL-like interface for users to extract data from Hadoop system. Since SQL knowledge was common in the programming world, anyone with SQL knowledge started using Hive. This significantly reduced the data processing time.

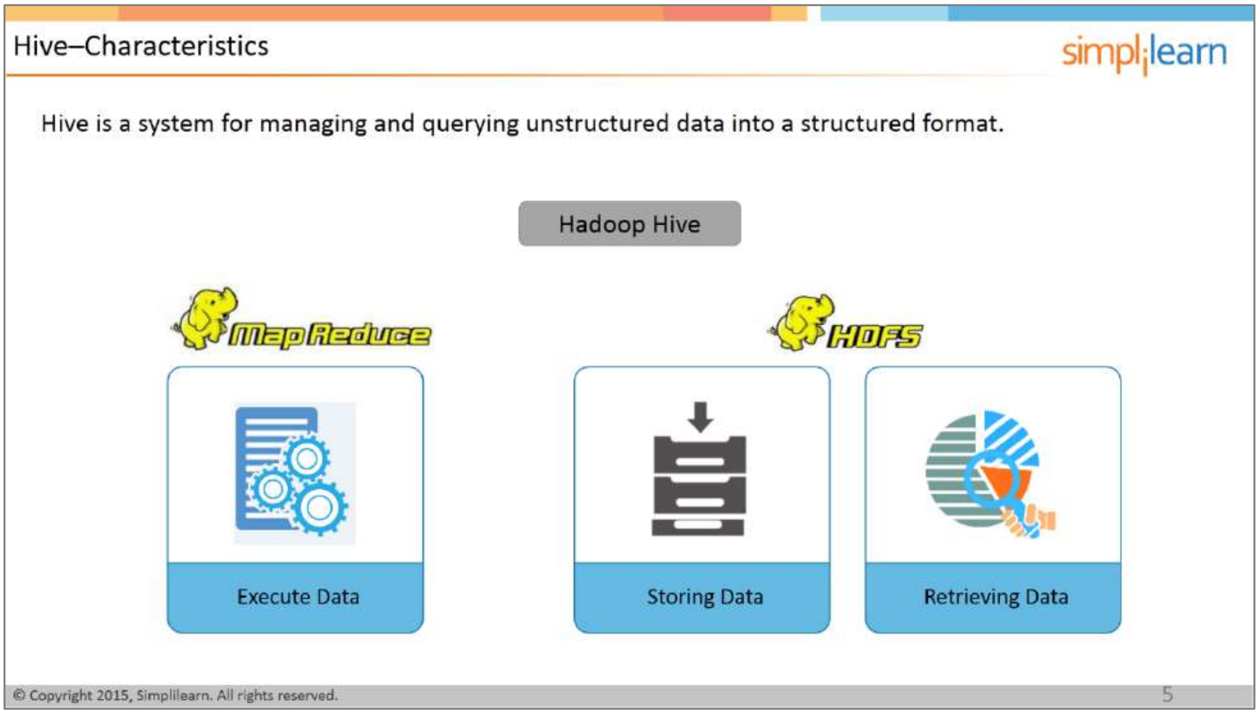


What is Hive

Hive is a Hadoop-based data warehousing tool. It supports Hive Query Language or HQL, which is similar to SQL. It provides summarization, query, and analysis of large data sets stored in Hadoop. The USP of Hive is effective and simple integration of data, ad hoc querying, and high volume data analysis.

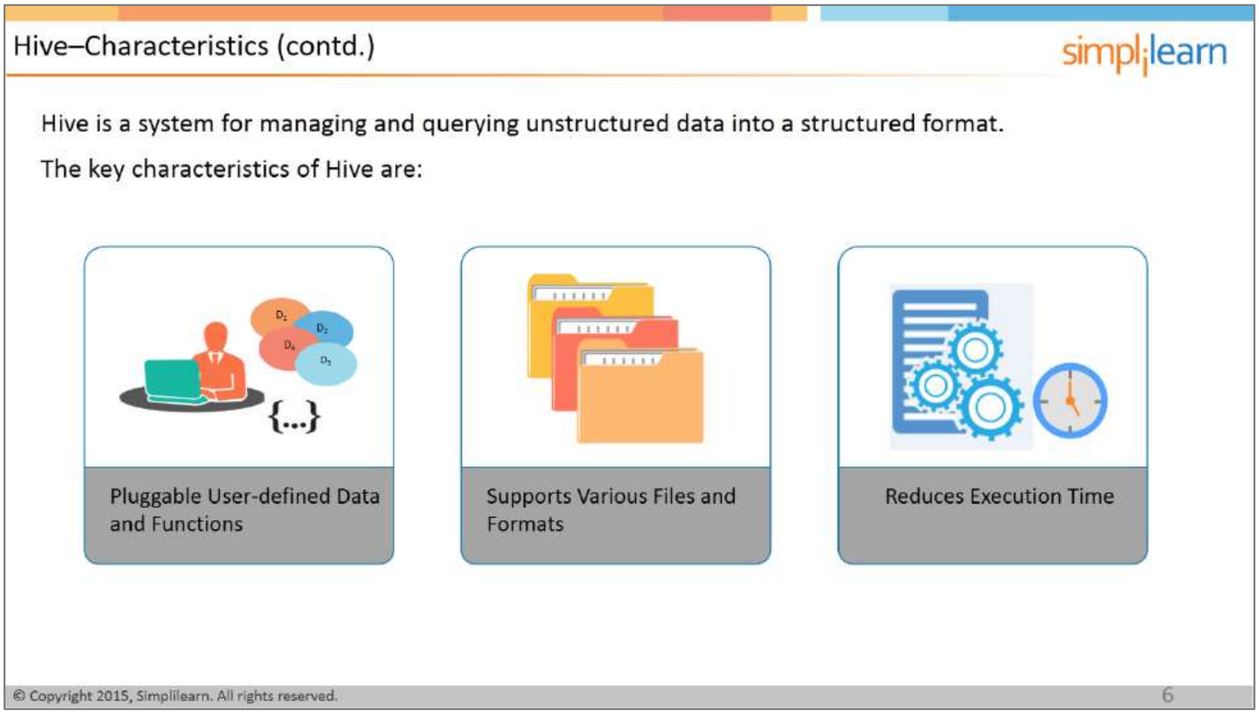
The gap between the relational database models and Hadoop is successfully bridged by Hive.

Recently, Hive has been provided with Apache Tez, which improved query latency significantly.



Hive–Characteristics

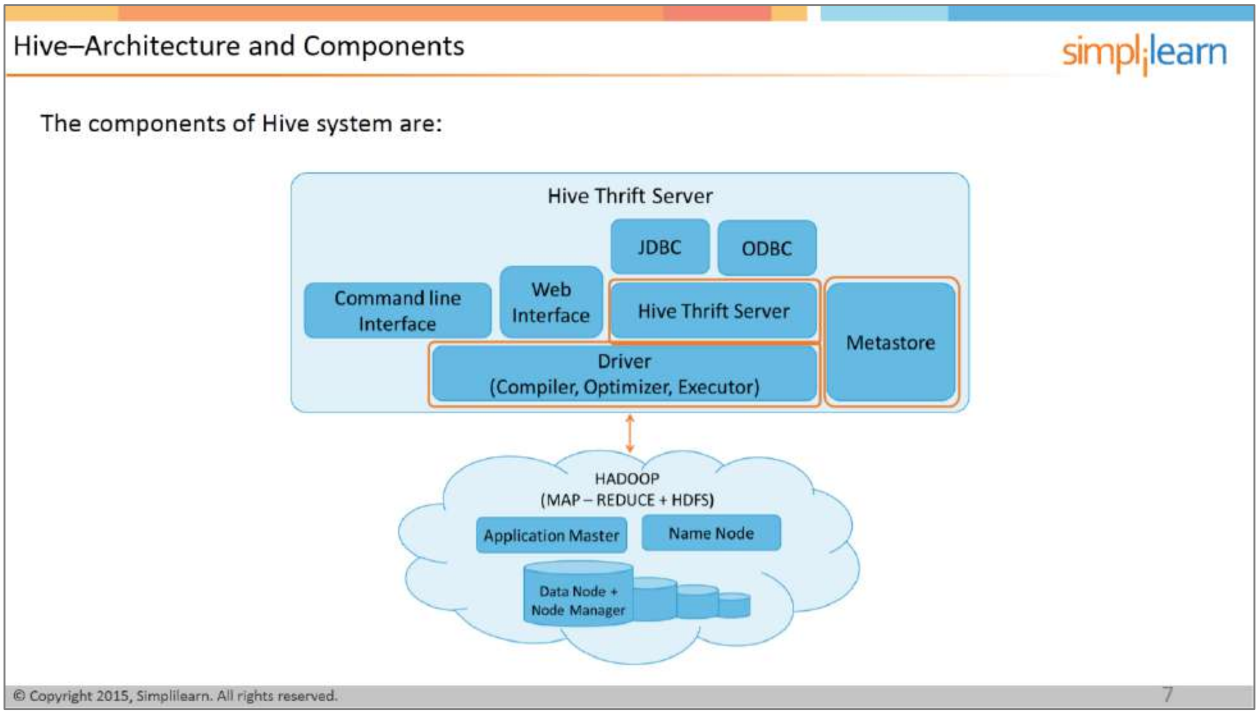
Hive is a system for managing and querying unstructured data into a structured format. It uses the MapReduce concept for the execution of its scripts and HDFS for the storage and retrieval of data.



Hive–Characteristics (contd.)

Following are the key characteristics of Hive:

- It contains extensive, pluggable MapReduce scripts that include user-defined data types and functions.
- It has an extensible framework to support different files and data formats.
- It reduces the execution time using best built-in script; thus, enabling high output in less time with better performance.

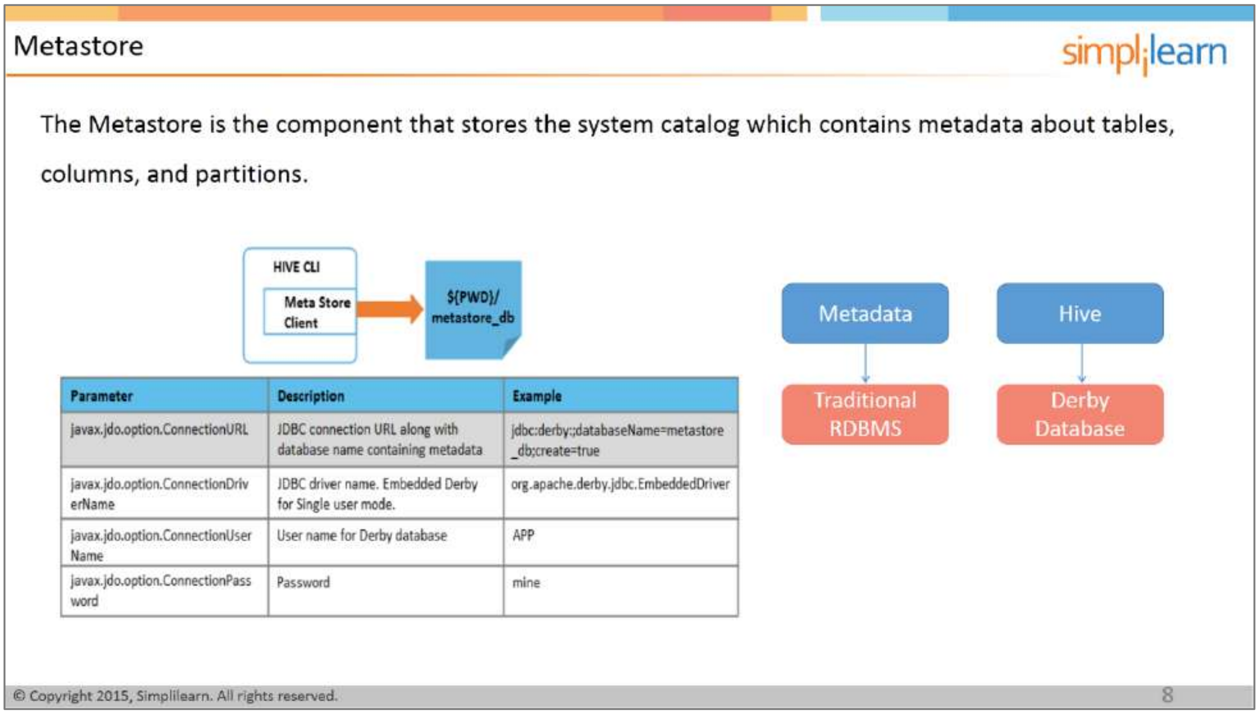


Hive–Architecture and Components

Let's now look at Hive architecture, its components, and role in the development process. The components of Hive system are:

- Metastore,
- Driver,
- Hive Thrift Server, and
- Client Components.

We'll now discuss each component in detail.



Parameter	Description	Example
javax.jdo.option.ConnectionURL	JDBC connection URL along with database name containing metadata	jdbc:derby:;databaseName=metastore_db;create=true
javax.jdo.option.ConnectionDriverName	JDBC driver name. Embedded Derby for Single user mode.	org.apache.derby.jdbc.EmbeddedDriver
javax.jdo.option.ConnectionUserName	User name for Derby database	APP
javax.jdo.option.ConnectionPassword	Password	mine

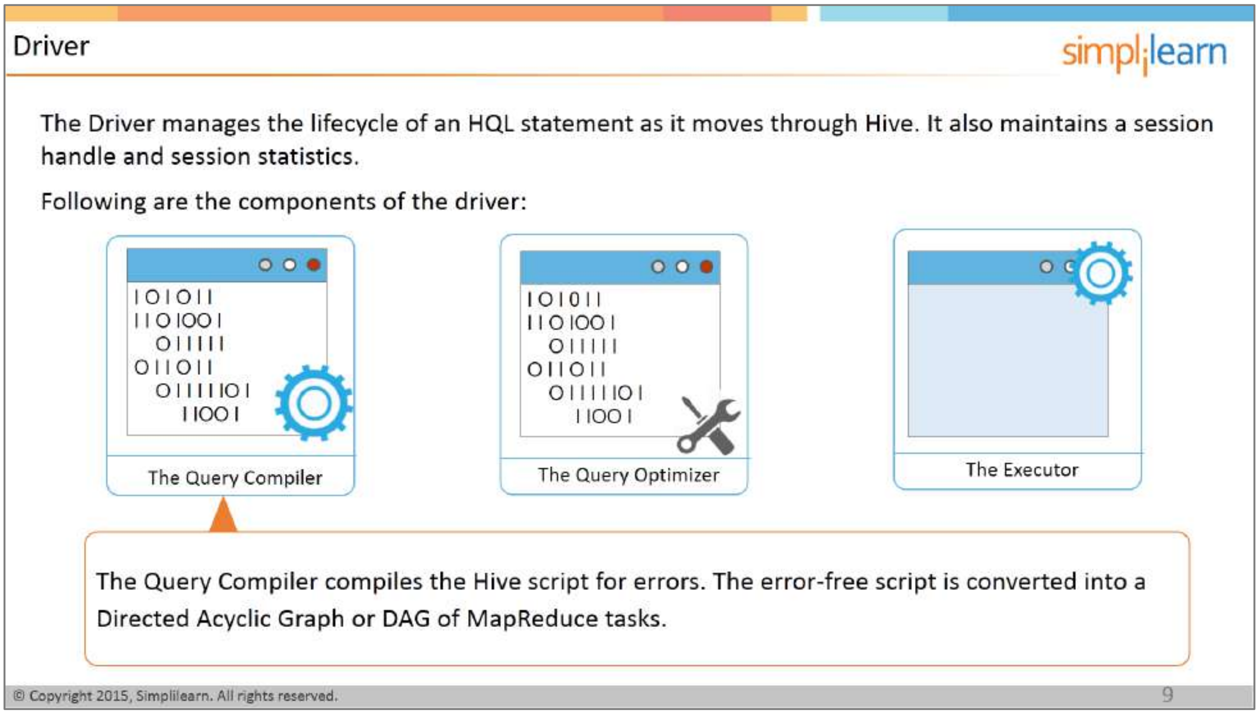
Metastore

The Metastore is the component that stores the system catalog which contains metadata about tables, columns, and partitions.

Usually, metadata is stored in a traditional RDBMS format. However, by default, Apache Hive uses the Derby database. Any Java Database Connectivity or JDBC compliant database like MySQL can be used for Metastore.

A number of primary attributes must be configured for Hive Metastore. Some of them are:

- Connection URL
- Connection driver
- Connection user ID
- Connection password



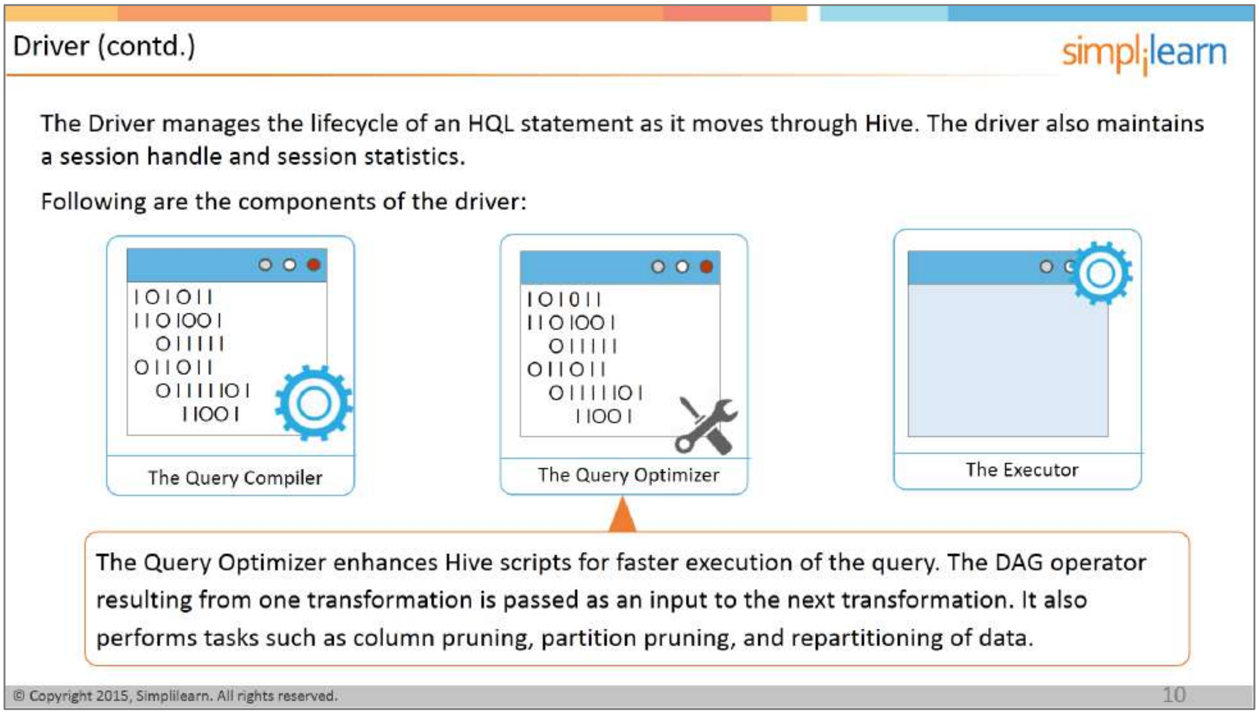
Driver

The Driver manages the lifecycle of an HQL statement as it moves through Hive. It also maintains a session handle and session statistics.

It includes three basic components:

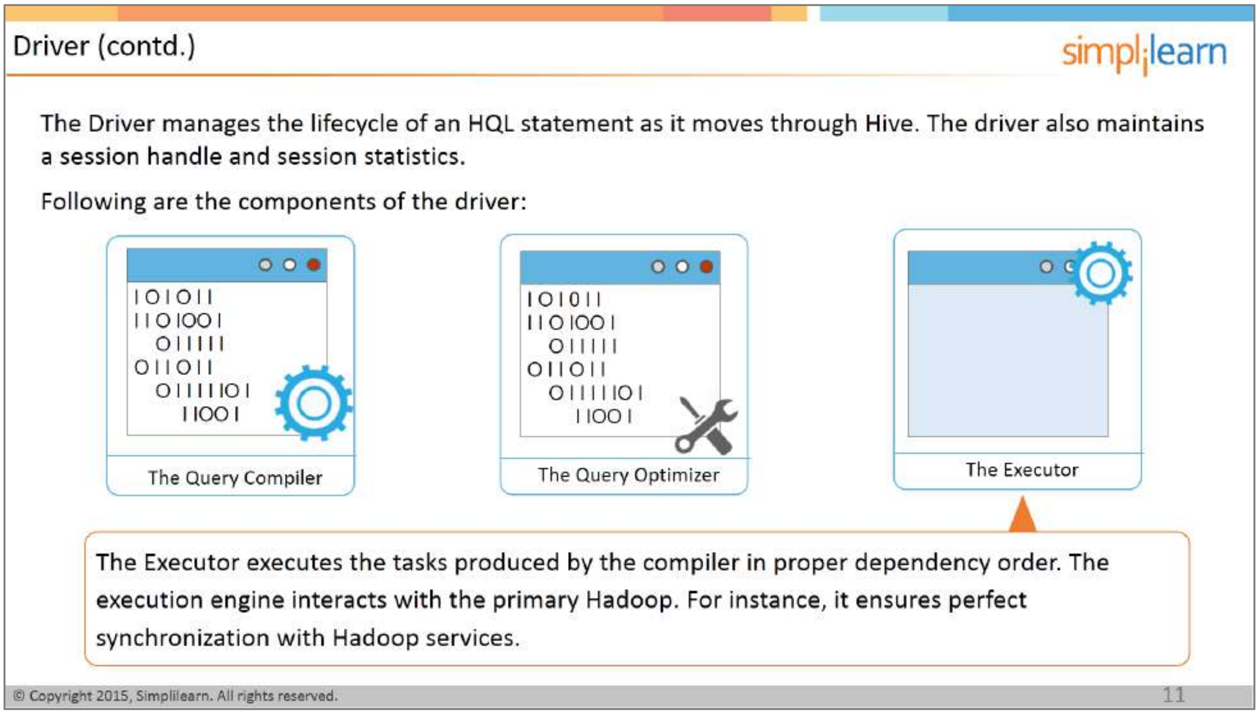
- the query compiler,
- the query optimizer, and
- the executor.

The Query Compiler compiles the Hive script for errors. The error-free script is converted into a Directed Acyclic Graph or DAG of MapReduce tasks.



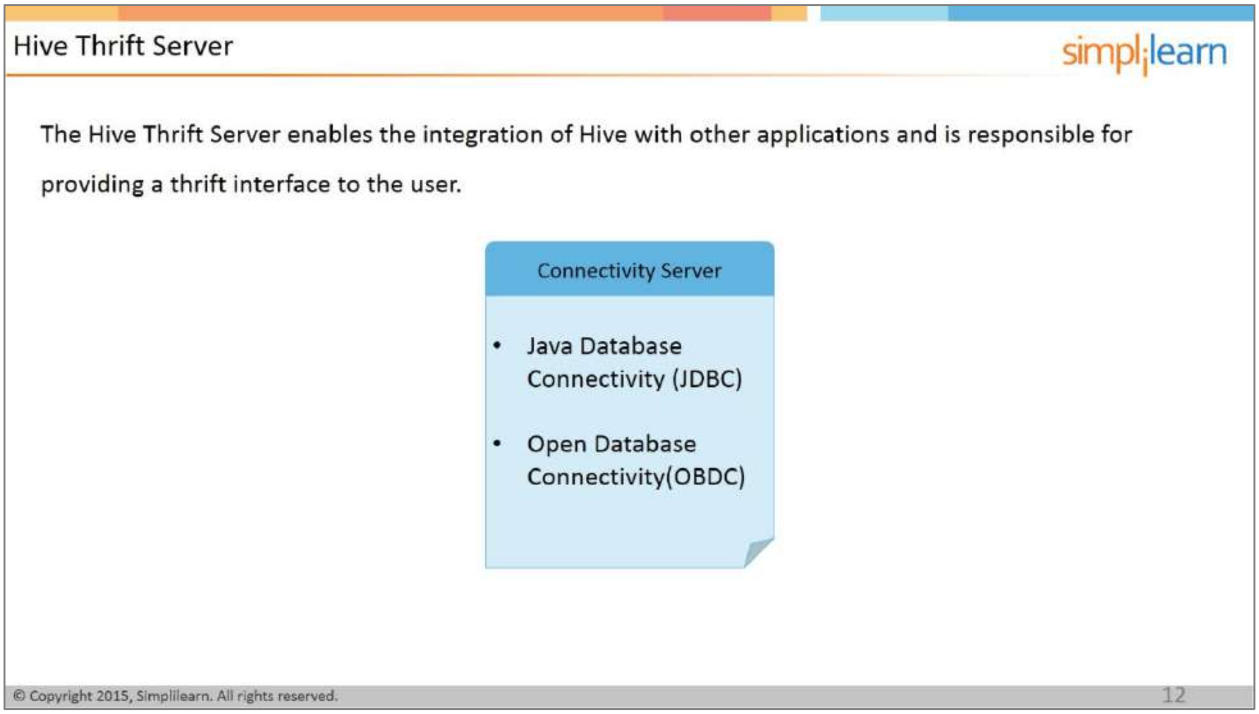
Driver (contd.)

The Query Optimizer enhances Hive scripts for faster execution of the query. The DAG operator resulting from one transformation becomes an input to the next transformation. This is known as chain of transformation. It also performs tasks such as column and partition pruning, and repartitioning of data.



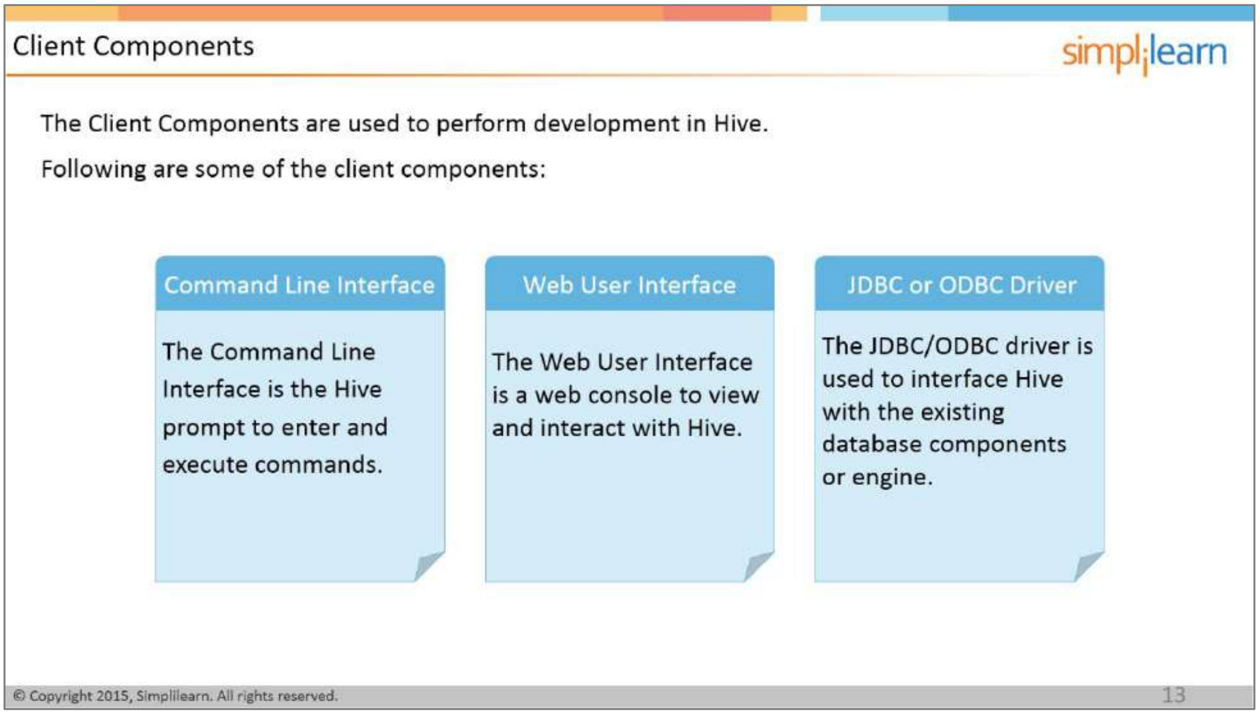
Driver (contd.)

The Executor executes the tasks produced by the compiler in a proper dependency order. The execution engine interacts with the primary Hadoop. For instance, it ensures perfect synchronization with Hadoop services.



Hive Thrift Server

The Hive Thrift Server enables the integration of Hive with other applications and is responsible for providing a thrift interface to the user. It also maintains connectivity in modules. It provides a Java Database Connectivity or JDBC, or Open Database Connectivity or ODBC server.



Client Components

The Client Components are used to perform development in Hive.

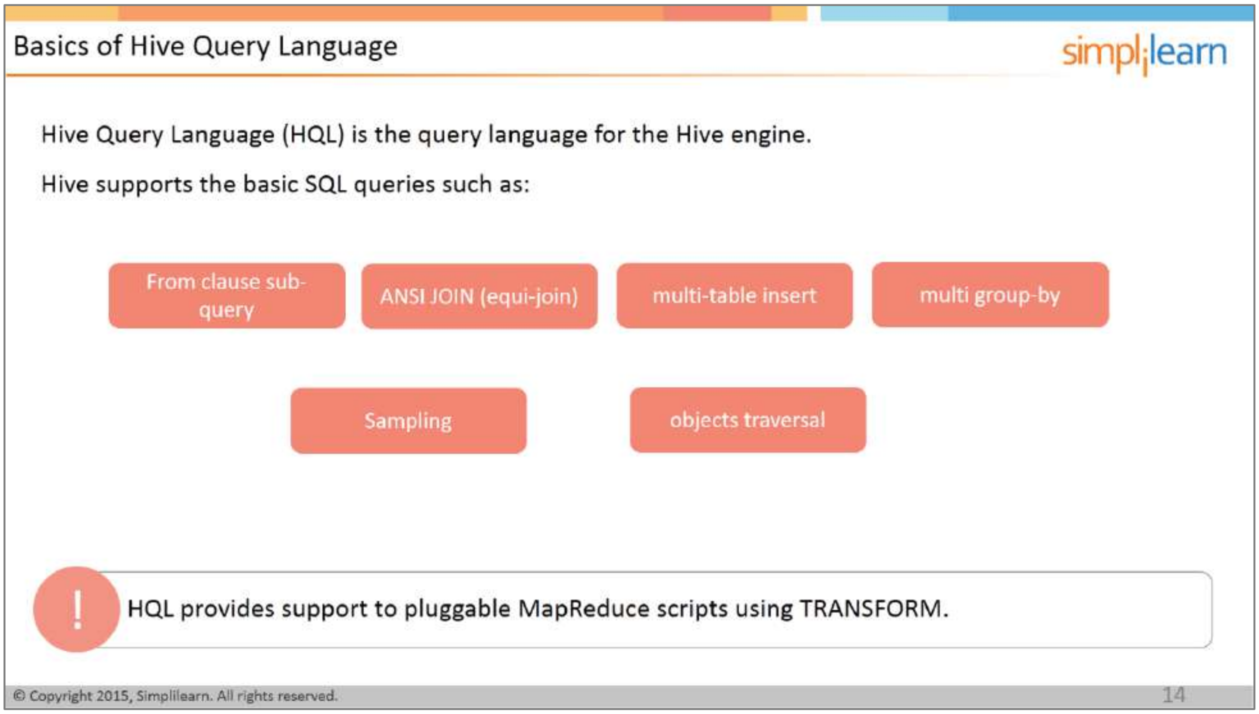
A few client components are:

- Command Line Interface or CLI,
- Web User Interface or UI, and
- the JDBC or ODBC driver.

The Command Line Interface is the Hive prompt to enter and execute commands.

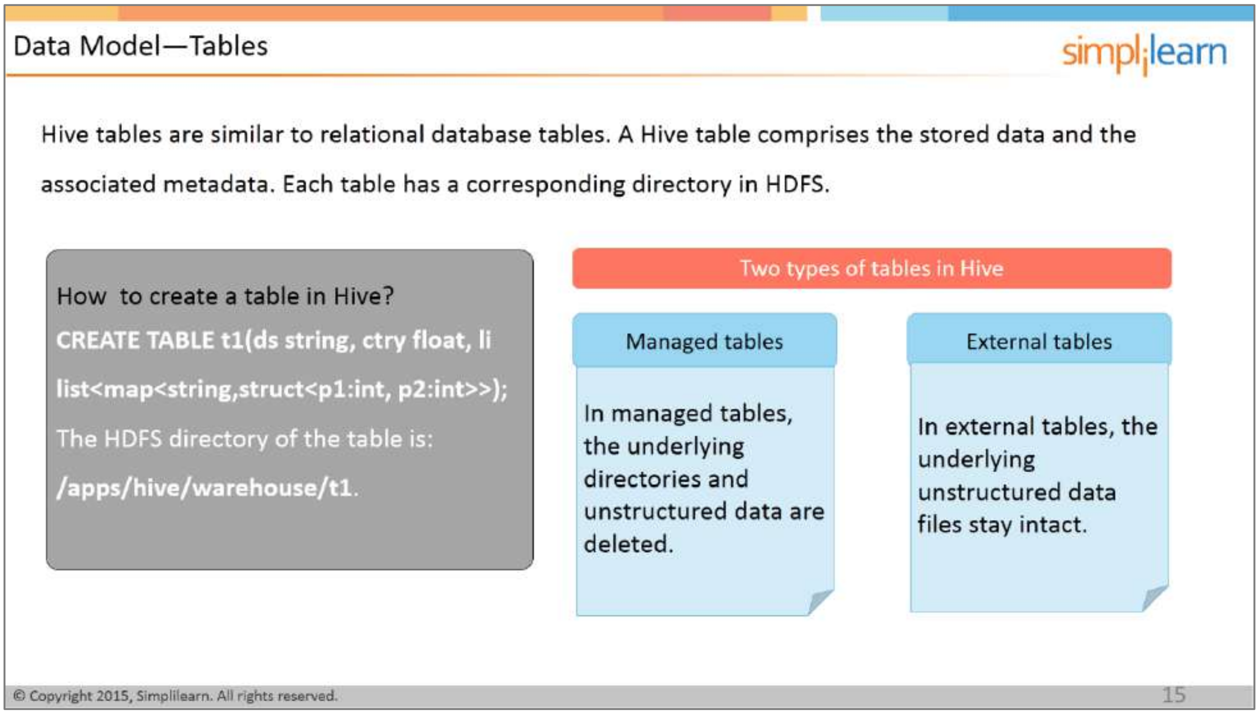
The Web User Interface is a web console to view and interact with Hive.

The JDBC or ODBC driver is used to interface Hive with the existing database components or engine.



Basics of the Hive Query Language

Hive Query Language or HQL is the query language for Hive engine. Hive supports the basic SQL queries such as the ‘From clause sub-query’, ‘ANSI JOIN’ such as ‘equi-join’, multi-table insert, multi group-by, sampling, and objects traversal. HQL provides support to pluggable MapReduce scripts using the command ‘TRANSFORM.’



Data Model—Tables

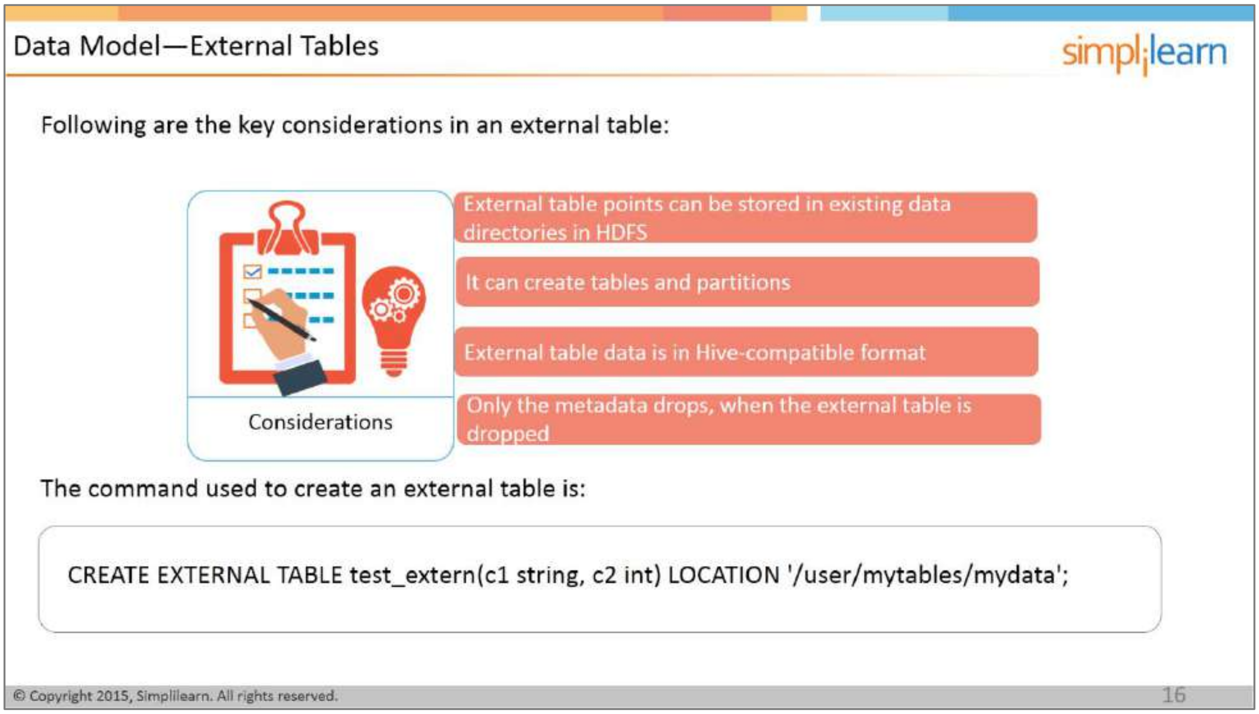
Hive tables are similar to relational database tables. A Hive table comprises the stored data and the associated metadata. Each table has a corresponding directory in HDFS.

Let’s discuss how to create this table in Hive.

You can create the table with the keyword ‘CREATE TABLE.’ This table should be located in HDFS in a directory.

There are two types of tables in Hive. They are managed tables and external tables.

In managed tables, the underlying directories and unstructured data are deleted. However, in external tables, these data files stay intact when the table is created.



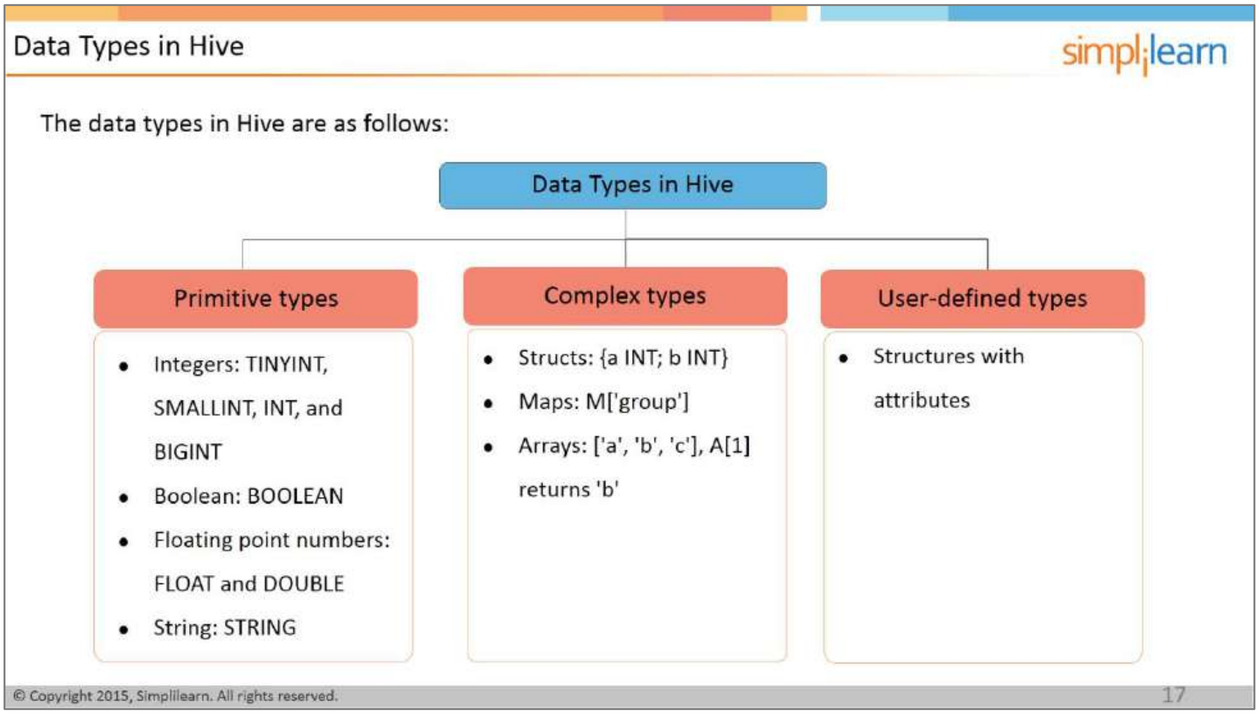
Data Model—External Tables

Following are the key considerations in an external table:

- External table points can be stored in existing data directories in HDFS
- It can create tables and partitions
- External table data is in Hive-compatible format
- Only the metadata drops, when the external table is dropped

You can create the external table using the keyword 'CREATE EXTERNAL TABLE' followed by its location.

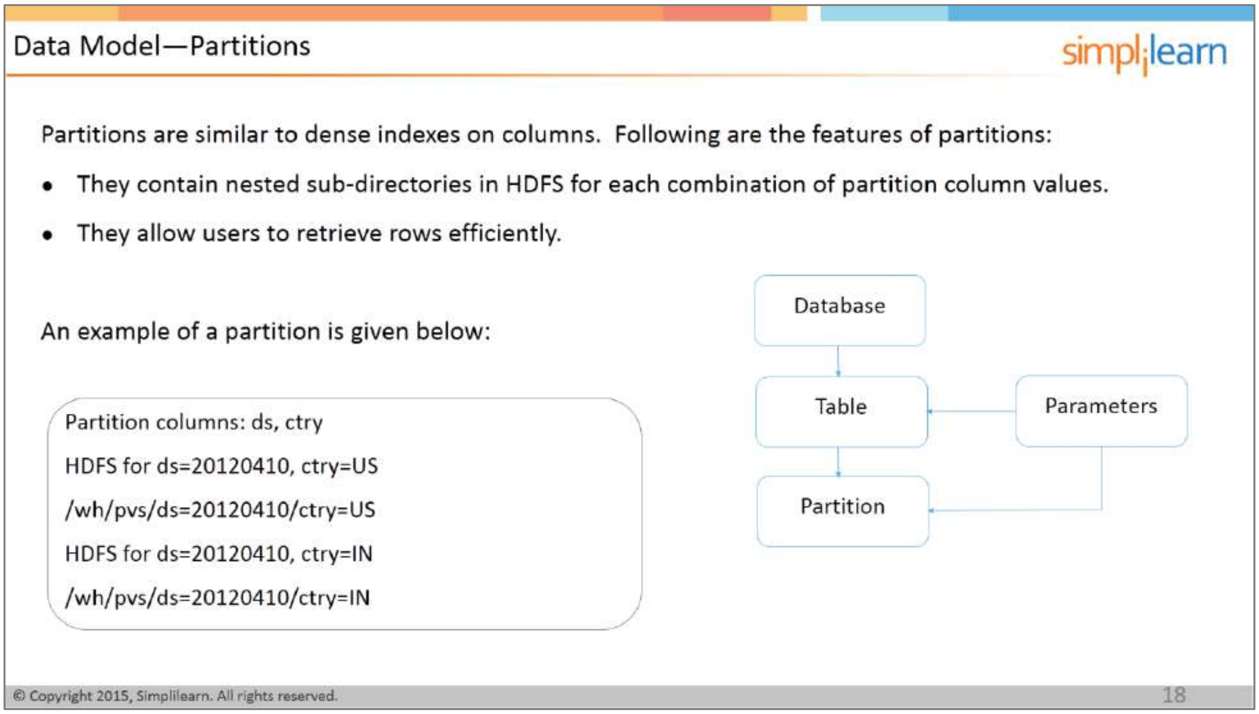
For example, the external table 'test' is created using the script shown. It uses the unstructured data available at the mentioned location.



Data Types in Hive

The three data types in Hive are primitive, complex, and user-defined.

- Primitive data types include integers such as TINYINT, SMALLINT, INT, and BIGINT. They also include BOOLEAN, floating point numbers type such as FLOAT and DOUBLE, and STRING.
- Complex types include Structs, Maps, and Arrays.
- User-defined types include structures with attributes of any type.



Data Model—Partitions

Partitions are similar to dense indexes on columns.

Following are the features of partitions:

- They contain nested sub-directories in HDFS for each combination of partition column values.
- They allow users to retrieve rows efficiently.

Data Model—Queries Used to Create Partitions

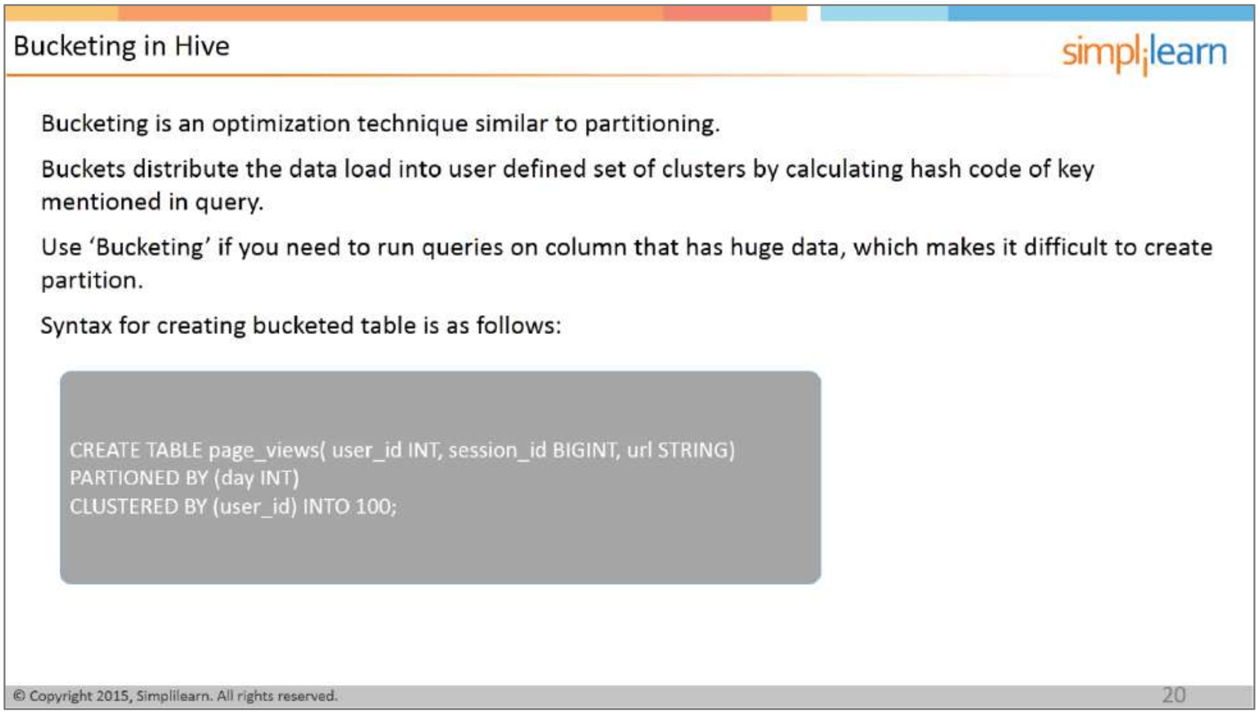
Following are some of the queries used to create partitions:

- CREATE TABLE test_part(ds string, hr int) PARTITIONED BY (ds string, hr int)
- INSERT OVERWRITE TABLEtest_part PARTITION(ds='2009-01-01', hr=12)
- SELECT * FROM t
- ALTER TABLE test_part ADD PARTITION(ds='2009-02-02', hr=11)
- SELECT * FROM test_part WHERE ds='2009-01-01' (this query will only scan all the files within the user/hive/warehouse/test_part/ds=2009-01-01 directory)
- SELECT * FROM test_part WHERE ds='2009-02-02' AND hr=11 (this query will only scan all the files within the /user/hive/warehouse/test_part/ds=2009-02-02/hr=11 directory)

 19

Data Model—Queries Used to Create Partitions

You can use the listed queries to create partition and insert data into them.



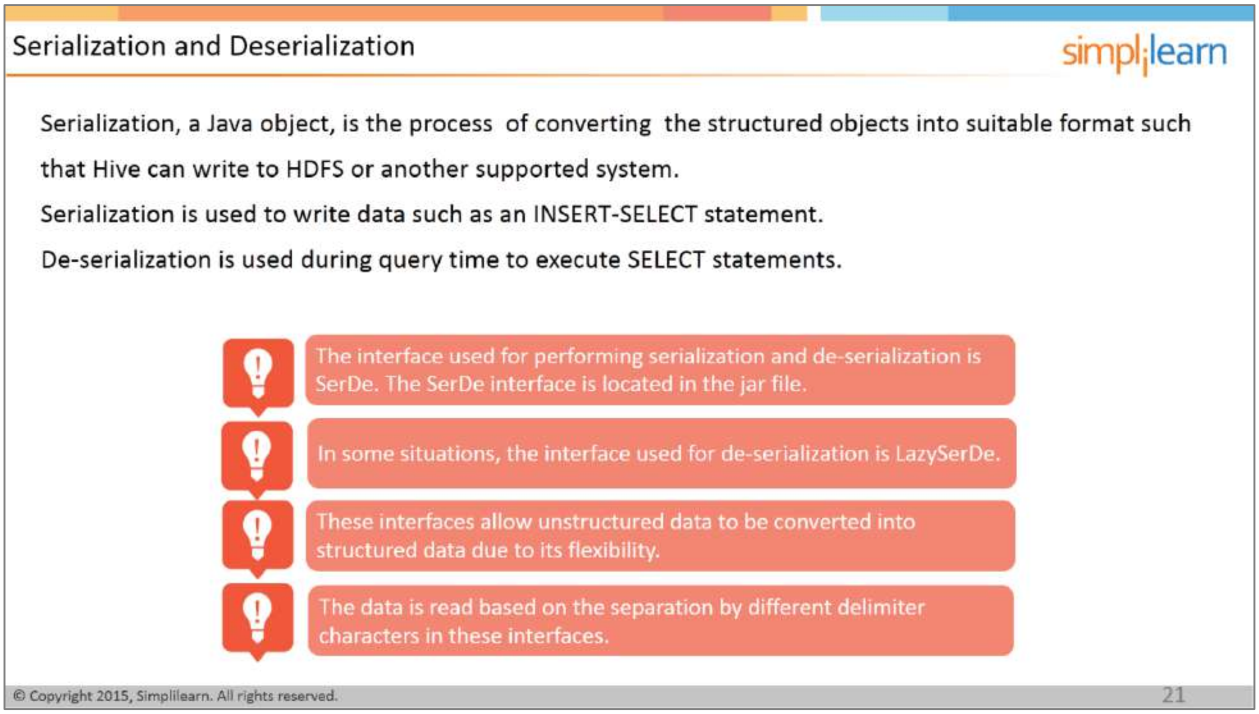
Bucketing in Hive

Bucketing is an optimization technique similar to partitioning. Buckets distribute the data load into user defined set of clusters by calculating hash code of the key mentioned in the query.

You can use the concept 'Bucketing' if you need to run queries on column that has huge data which makes it difficult to create partition.

Syntax for creating bucketed table is given.

Here the data would be classified depending on hash number of user_id into 100 buckets and the processor will first calculate the hash number of the user_id in query and will look for only that bucket.



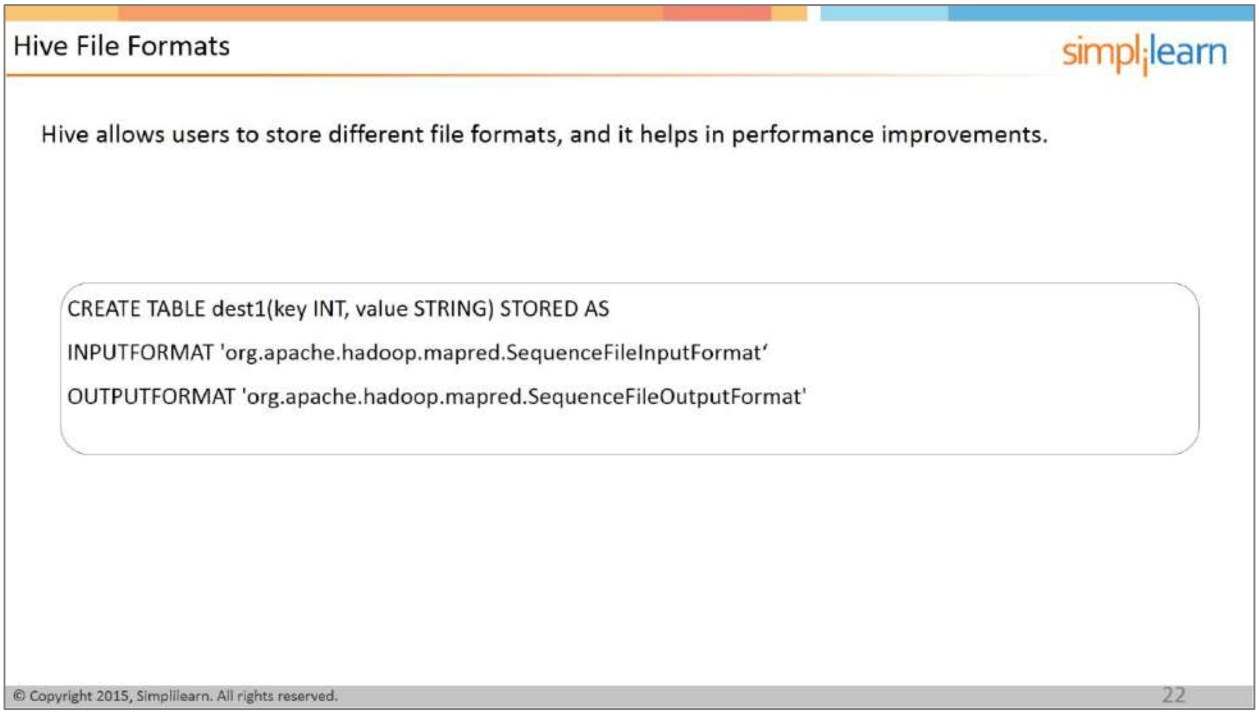
Serialization and Deserialization

Serialization, a Java object, is the process of converting the structured objects into suitable format such that Hive can write to HDFS or another supported system.

Serialization is used to write data such as an INSERT-SELECT statement. De-serialization is used during query time to execute SELECT statements.

Other facts related to serialization and deserialization are:

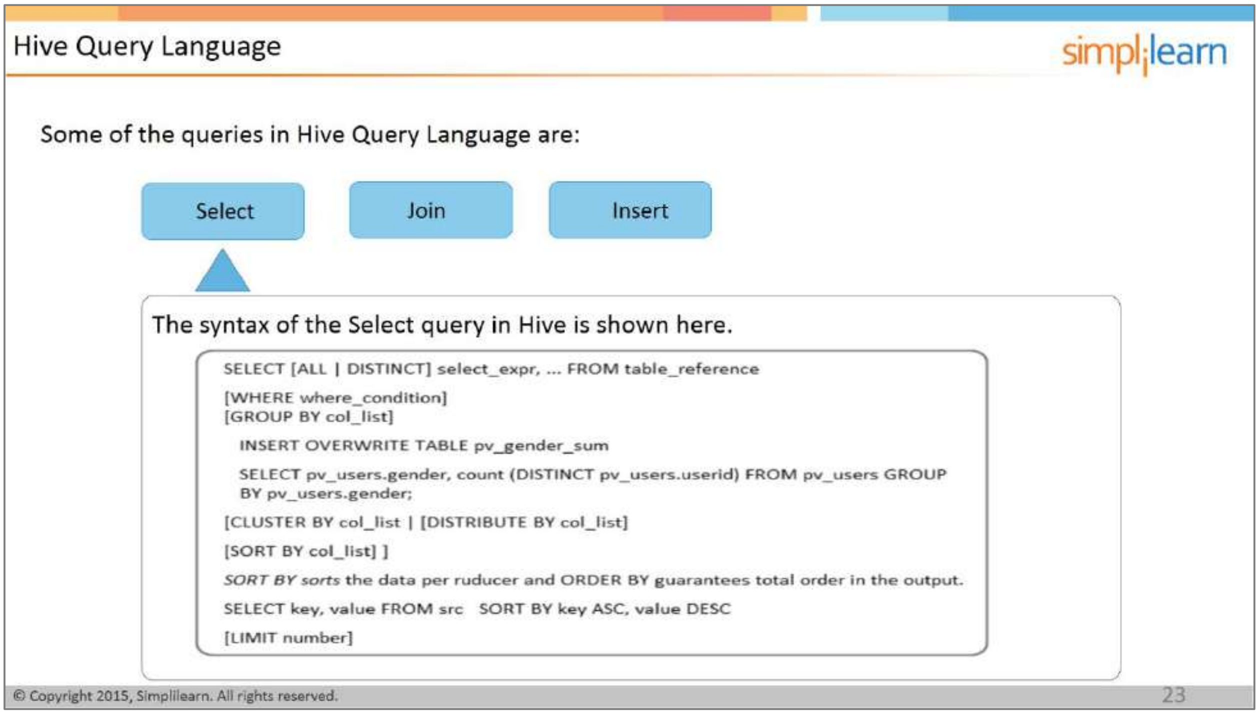
- The interface used for performing serialization and de-serialization is SerDe. The SerDe interface is located in the jar file.
- In some situations, the interface used for de-serialization is LazySerDe.
- These interfaces allow unstructured data to be converted into structured data due to its flexibility.
- The data is read based on the separation by different delimiter characters in these interfaces.



Hive File Formats

Hive allows users to store different file formats, and it helps in performance improvements with respect to the data operations such as storing and analyzing.

The example given shows how SQL is used to perform a file format operation in which the SequenceFileInputFormat class is used for input. The SequenceFileOutputFormat class is used for output.



Hive Query Language

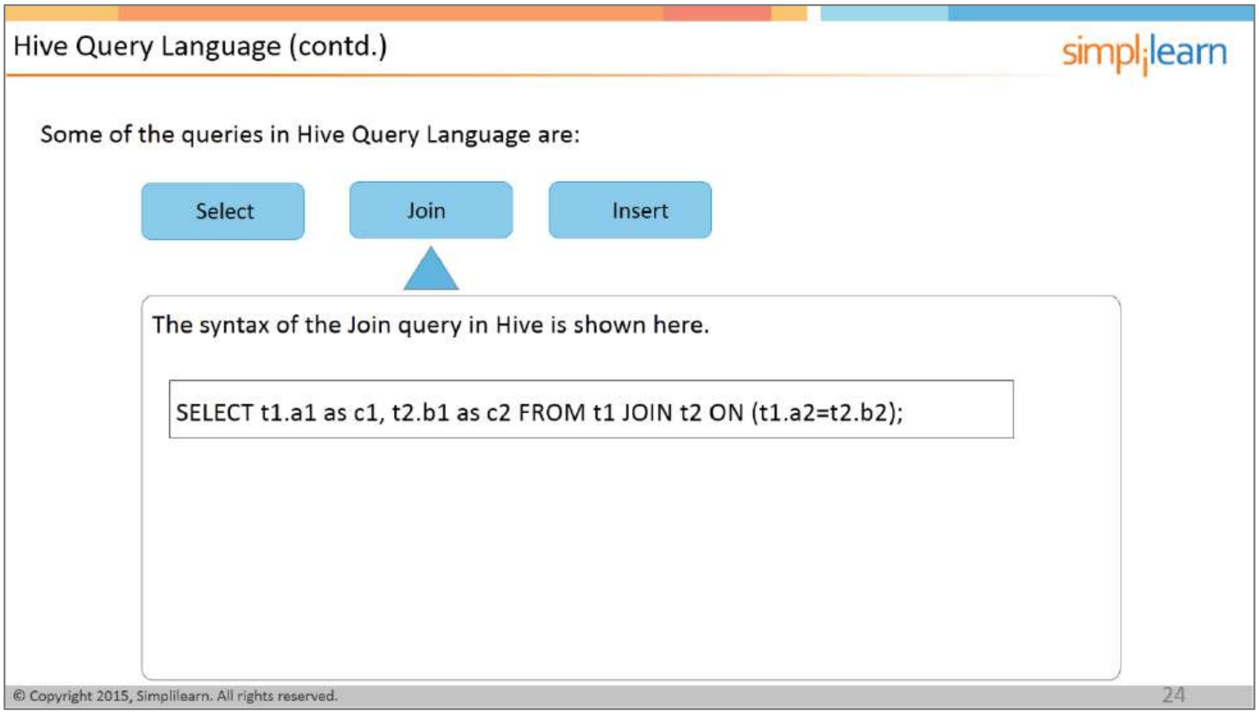
Let’s now discuss a few queries such as select, join, and insert in Hive Query Language.

Select:

The syntax of the Select query in Hive is shown.

Similar to a relational database, the Select query has WHERE, GROUP BY, SORT BY, and LIMIT clauses.

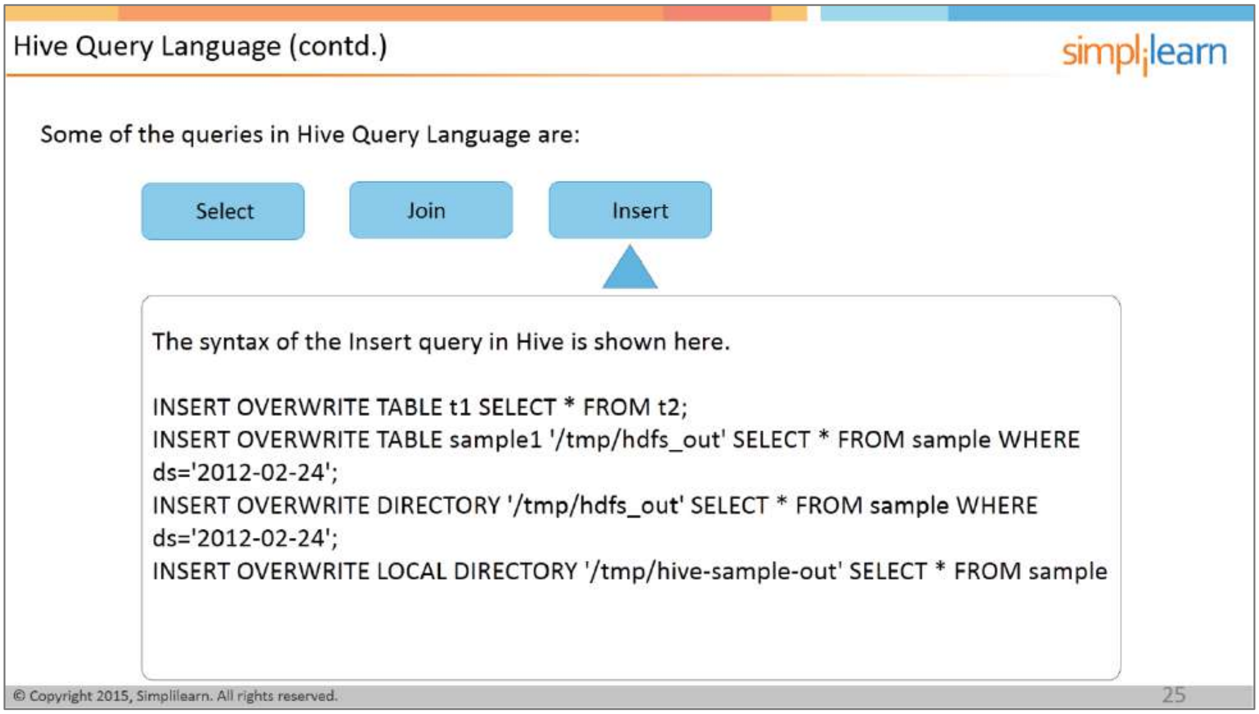
The Select query also supports nested queries.



Hive Query Language (contd.)

Join:

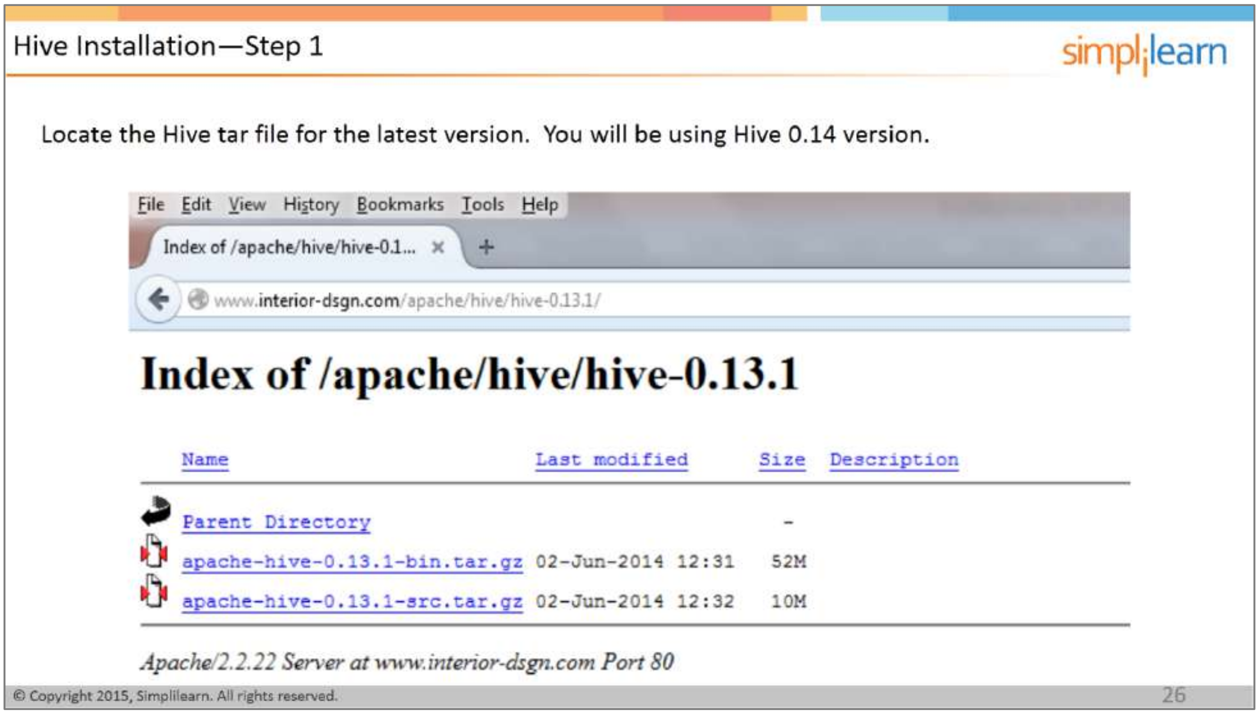
HQL supports JOIN commands. The Keyword used to join two queries is ‘JOIN.’



Hive Query Language (contd.)

Insert:

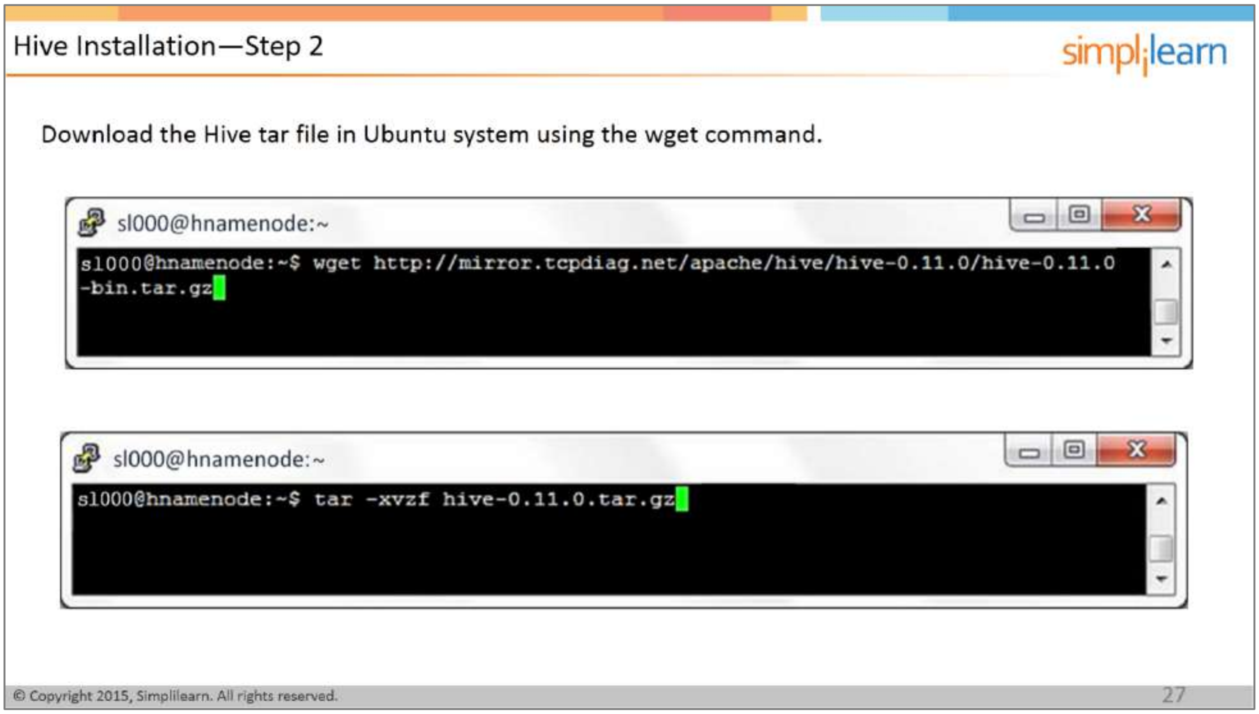
HQL supports INSERT commands as well. The syntax and some examples of INSERT command are shown here.



Hive Installation—Step 1

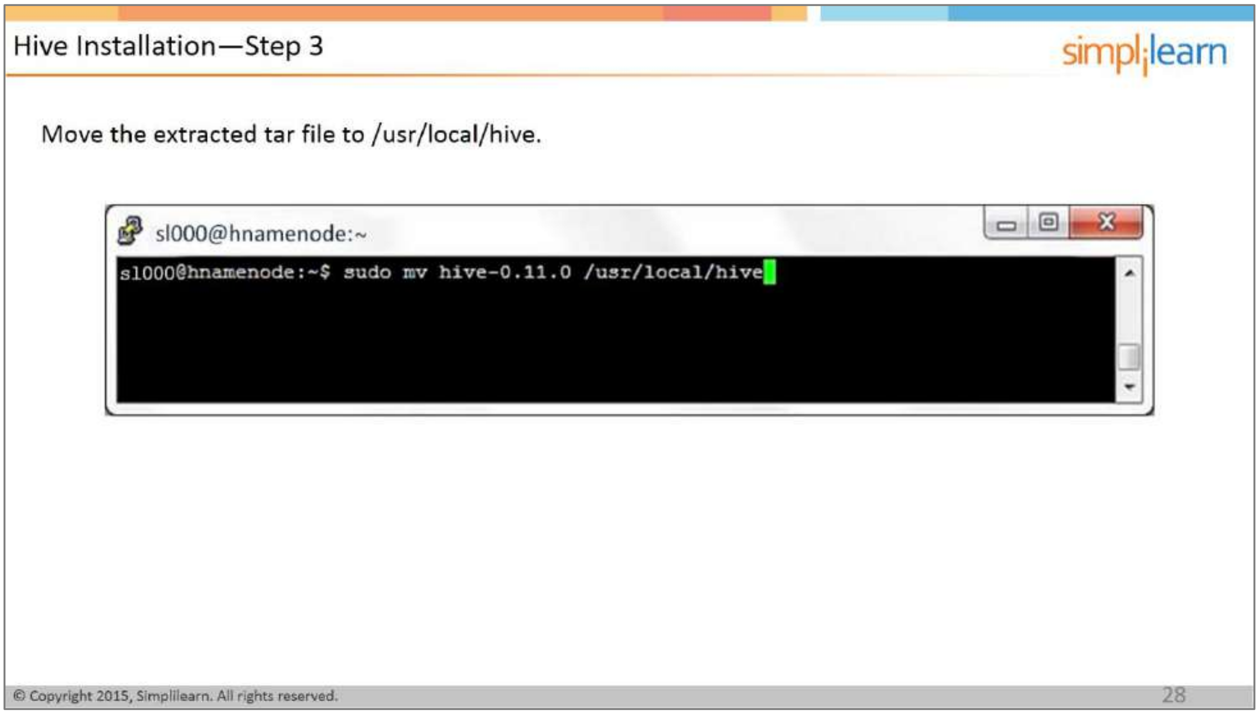
You can install Hive using four simple steps.

You need to locate the Hive tar file. You will be using Hive version 0.14. You can obtain the download location by visiting the URL shown in the image.



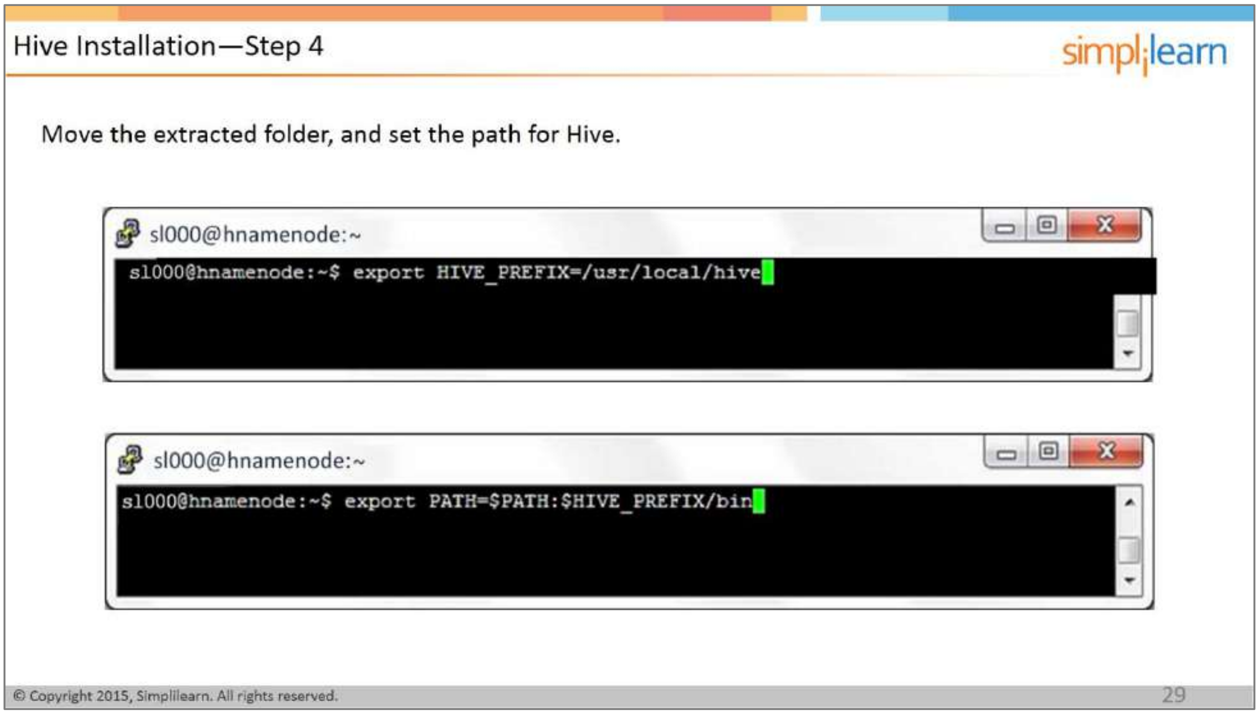
Hive Installation—Step 2

Use the wget command to download the Hive tar file in Ubuntu system. Once the download is complete, you need to extract the Hive tar file. The command for extracting the Hive tar file is shown in the image.



Hive Installation—Step 3

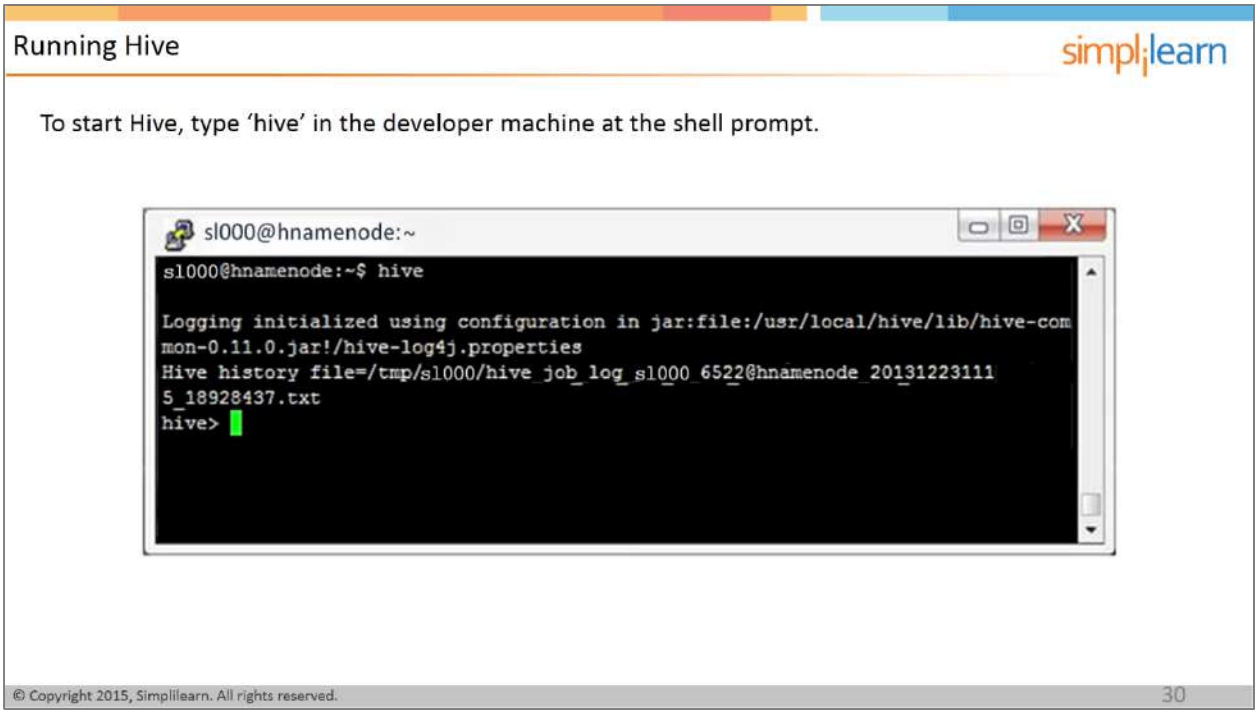
Move the extracted tar file to the location mentioned. Use the command displayed in the image to perform this step of Hive installation.



Hive Installation—Step 4

Ensure you have moved the extracted folder, and set the path for Hive.

Use the command shown in the first image to set the Hive prefix. To set the path for Hive, use the command shown in the second image.

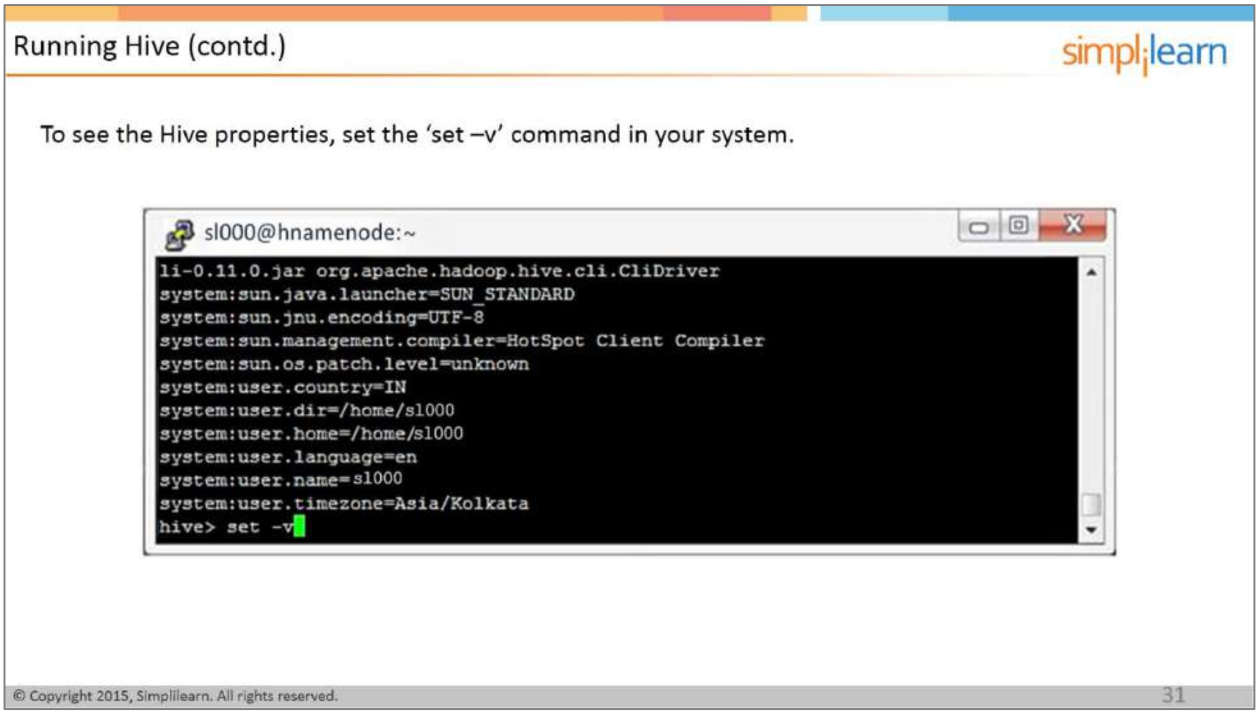


Running Hive

To start the Hive prompt, use the command 'hive' and press **Enter**. You will get a prompt similar to the one shown in the image.

There are a number of properties that are set in your Hive system.

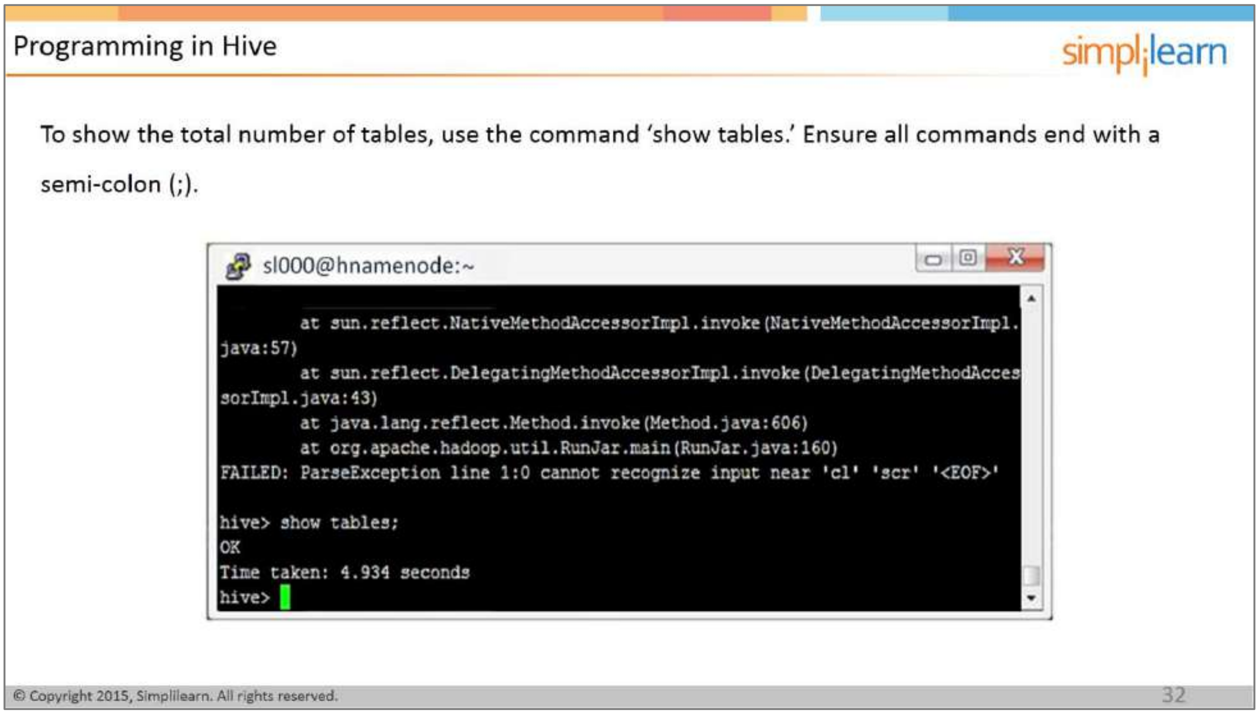
To set Hive properties in your machine, use the command 'set –v.'



Running Hive (contd.)

There are a number of properties that are set in your Hive system.

To set Hive properties in your machine, use the command 'set –v.'

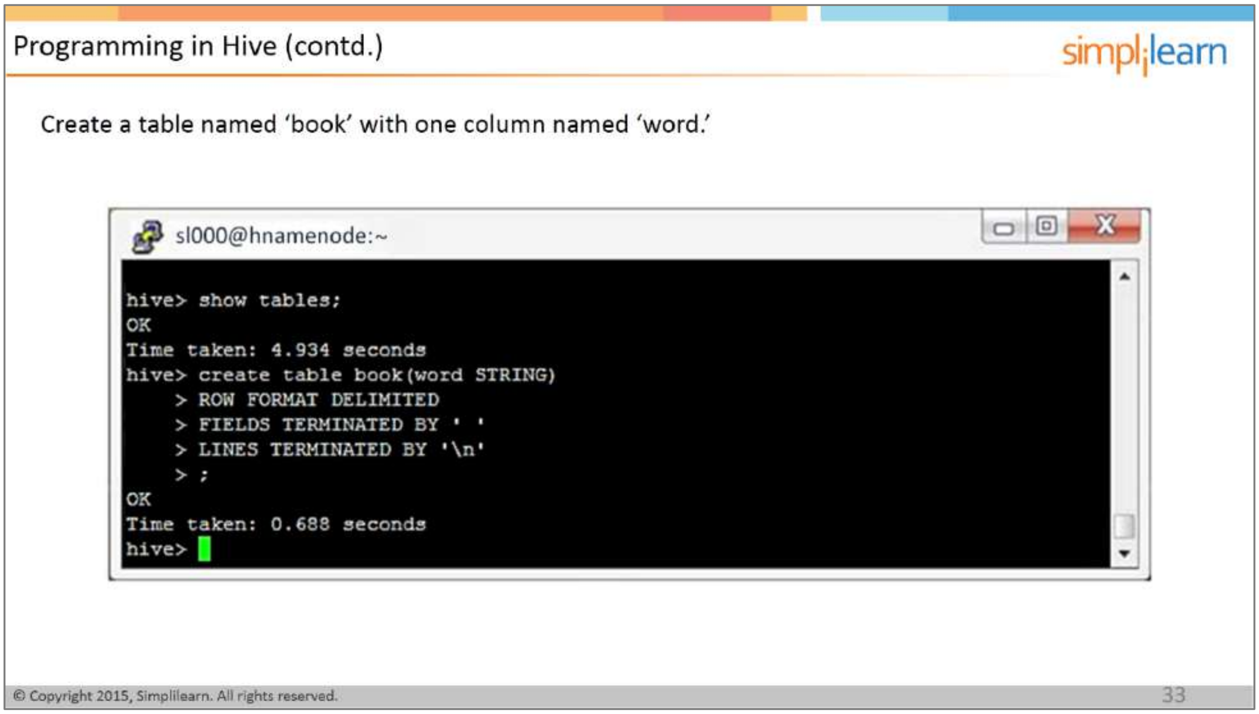


Programming in Hive

So far you have learned to install and run Hive. Let's start to program in Hive.

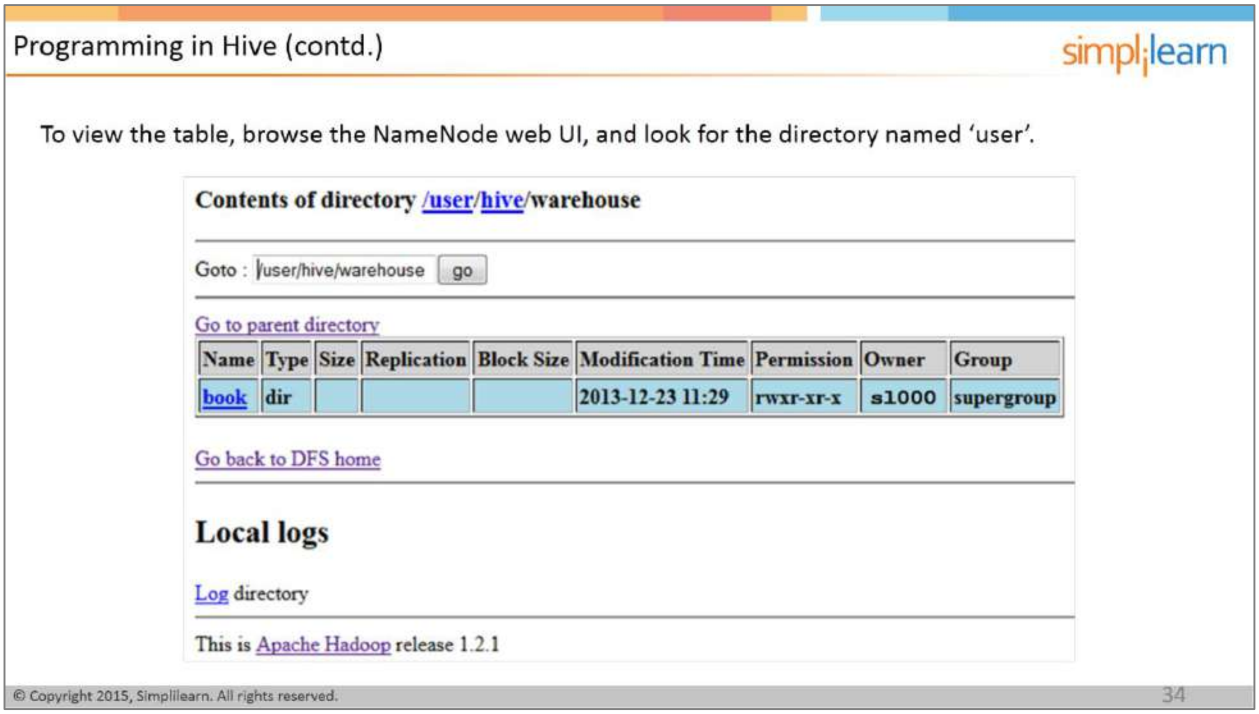
To show the total number of tables, use the command 'show tables.' Ensure all commands end with a semi-colon (;).

Initially, you will get an output similar to the one shown. This is because there are currently no tables in Hive.



Programming in Hive (contd.)

To see the table, you first need to create it. Use the command 'create table' to create them. Let the name of the table be 'book,' and it has one column named 'word.'



Programming in Hive (contd.)

Hive stores a table as a folder. To view the created table, browse the NameNode web UI, and look for the directory named 'user.'

Under the user directory, you will find a folder 'hive'. Under the folder 'hive,' there is a folder 'warehouse.' In the warehouse folder, you will find the table 'book.'

Programming in Hive (contd.)

Load the data in the 'book' table. Ensure you have the book 'War and Peace' loaded in HDFS.

Contents of directory /data/small

Goto : /data/small go

Go to parent directory

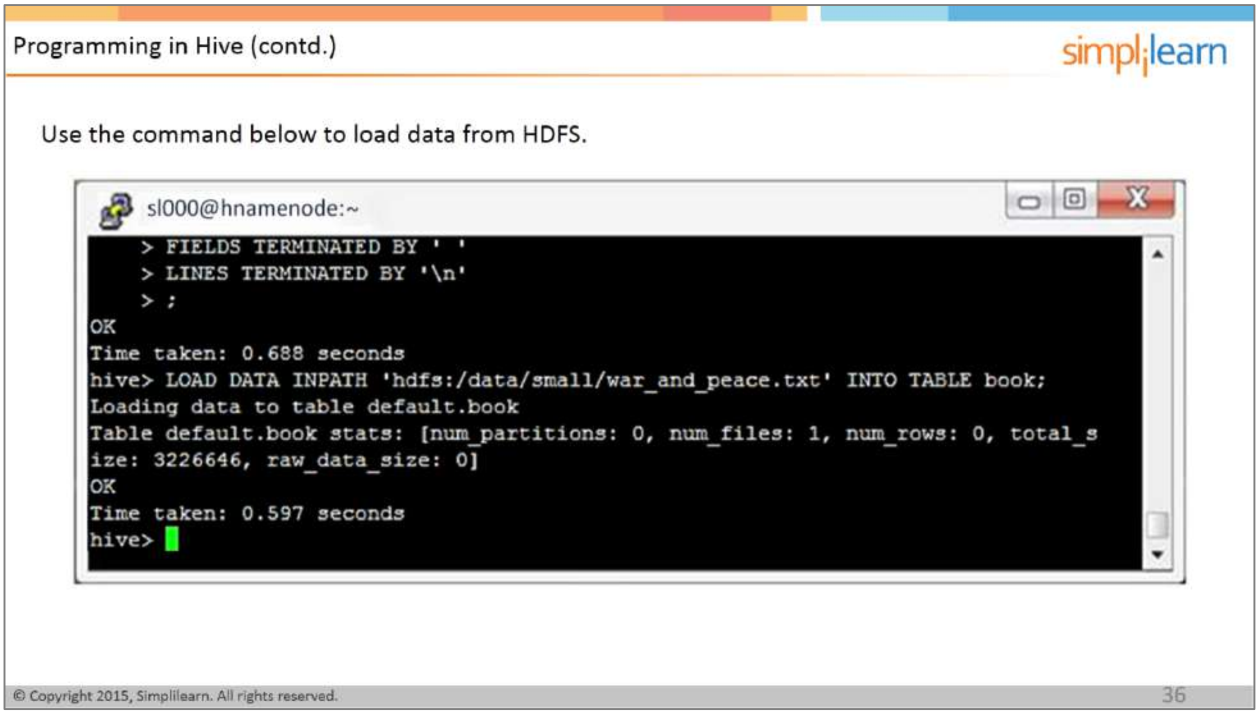
Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
pigresults1	dir				2013-12-16 11:57	rwxr-xr-x	s1000	supergroup
war_and_peace.txt	file	3.08 MB	3	64 MB	2013-12-13 09:56	rw-r--r--	s1000	supergroup

 35

Programming in Hive (contd.)

Once you have created the table, you need to add unstructured data.

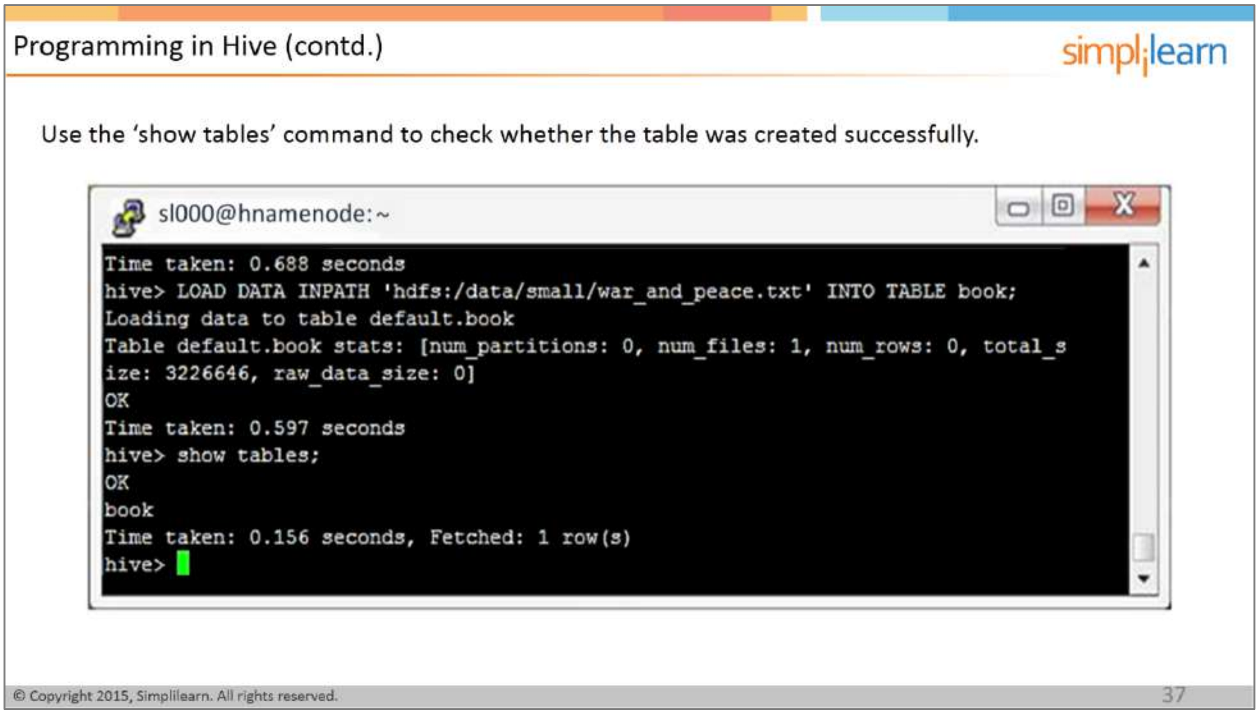
Here, add a book 'War and Peace' as a data. Ensure you have the book 'War and Peace' loaded in HDFS.



Programming in Hive (contd.)

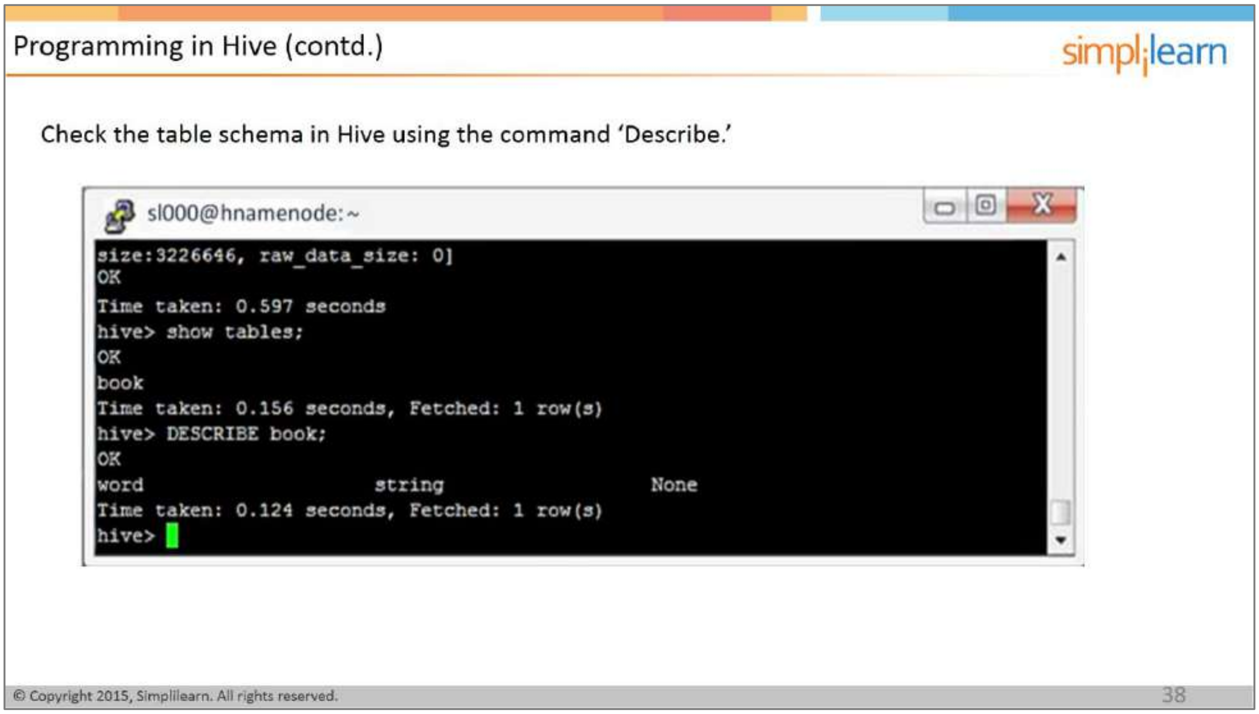
The next step is to load the data from HDFS to Hive.

Use the command 'LOAD DATA INPATH' to load the data from HDFS.



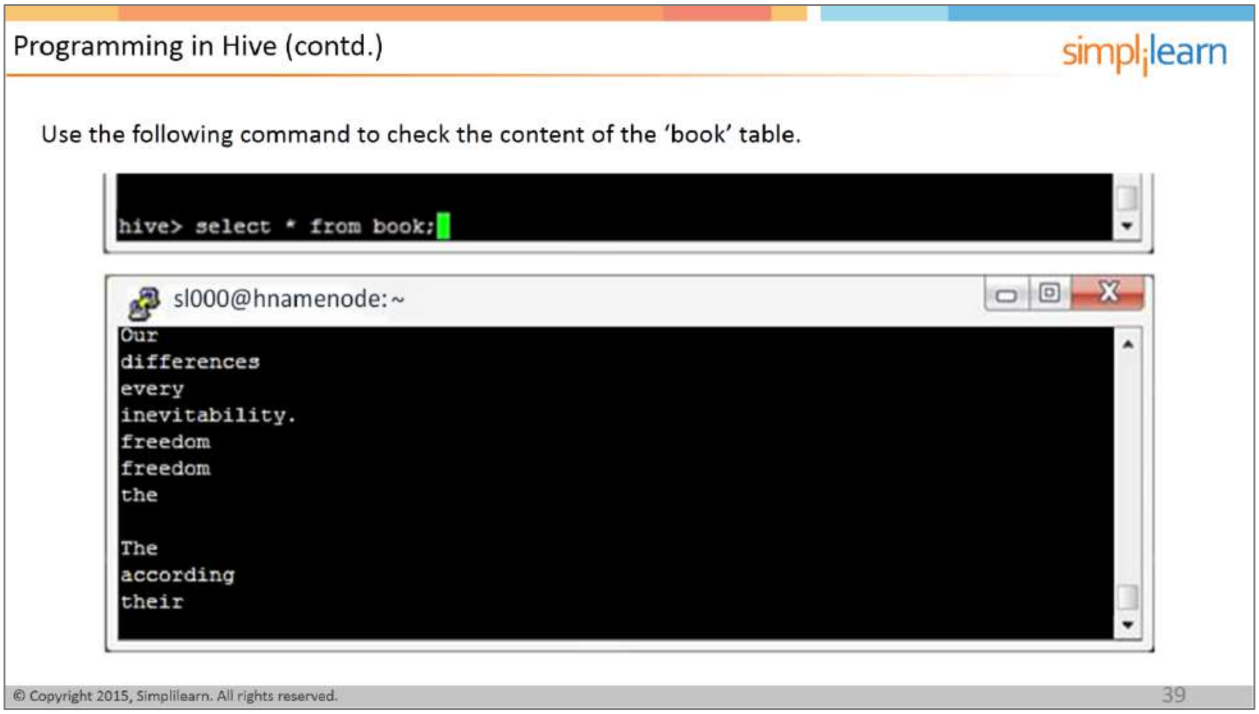
Programming in Hive (contd.)

You can now see the created table using the command 'show tables' in Hive prompt. The table is successfully created, when you find your table name in the Hive prompt list.



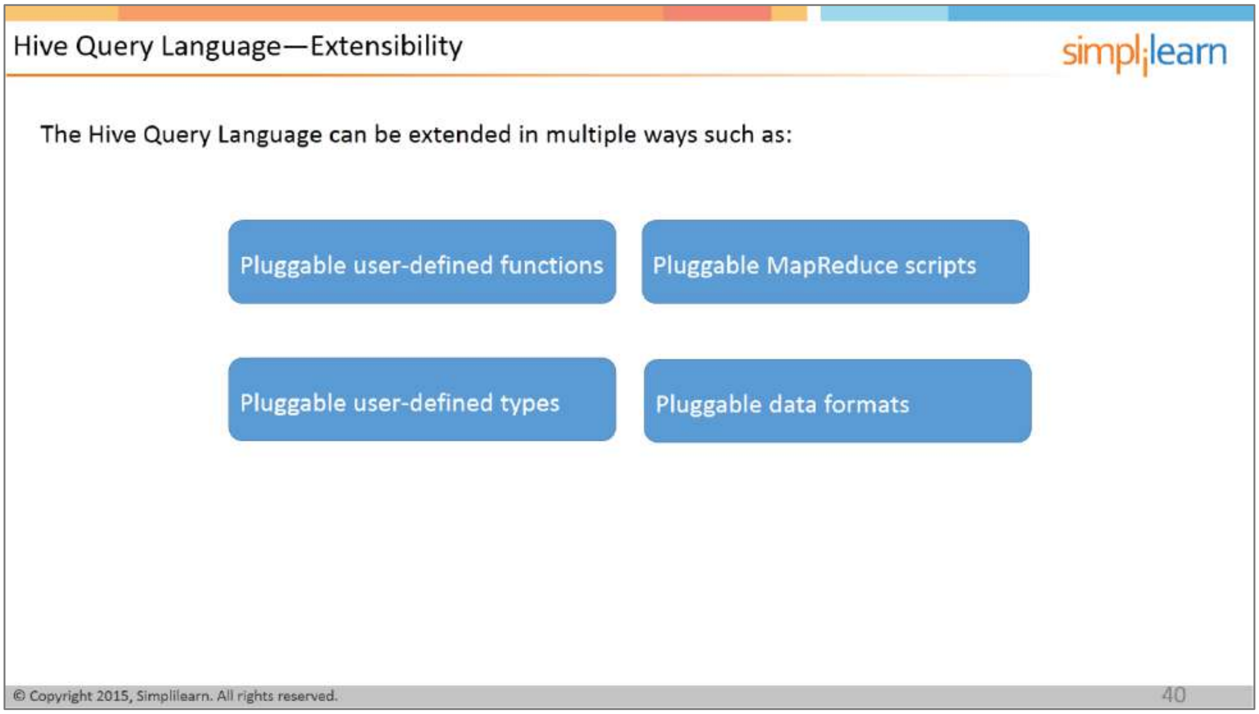
Programming in Hive (contd.)

You can also check the table schema in Hive using the command 'Describe.'



Programming in Hive (contd.)

Use the command given to check the content present in the table. You will obtain the output as words.



Hive Query Language—Extensibility

The Hive Query Language can be extended in multiple ways. Some of the common ways include:

- Pluggable user-defined functions
- Pluggable MapReduce scripts
- Pluggable user-defined types
- Pluggable data formats

User-Defined Function

Following is the code used to extend the user-defined function:

```
package com.example.hive.udf;

import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.Text;

public final class Lower extends UDF {
 public Text evaluate(final Text s) {
  if (s == null) { return null; }
  return new Text(s.toString().toLowerCase());
 }
}
```

41

User-Defined Function

Hive has the ability to define a function. Any new User-Defined Function or UDF class needs to inherit from the existing UDF class. All UDF classes need to implement one or more methods named ‘evaluate.’ Evaluate should never be a void method. However, it can return null if required.

After compiling UDF, you must include it in the Hive class path. Once Hive gets started, you can use the new defined function in a query statement after registering them.

You can use the given code to register the function.

User-Defined Function (contd.)

Following is the code used to register the class:

```
CREATE FUNCTION my_lower AS 'com.example.hive.udf.Lower';
```

Following is the code to use the function in a Hive query statement:

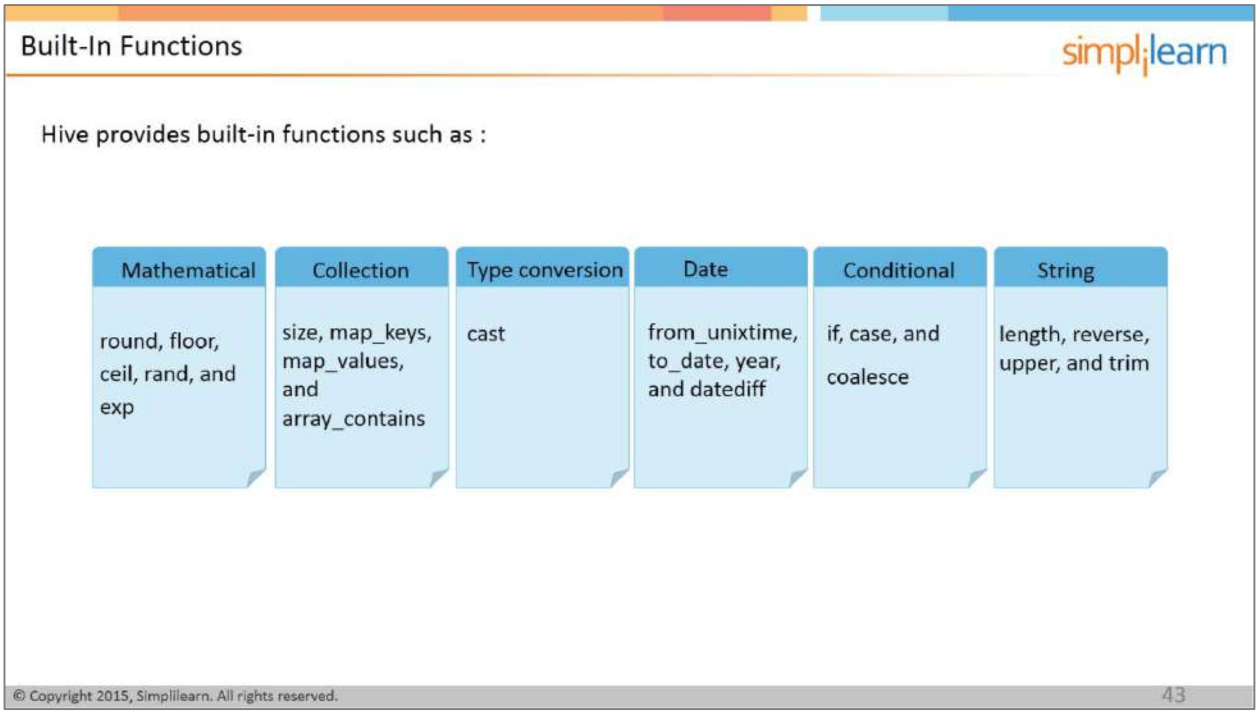
```
SELECT my_lower(title), sum(freq) FROM titles GROUP BY my_lower(title);
```

 42

User-Defined Function (contd.)

After compiling UDF, you must include it in the Hive class path. Once Hive gets started, you can use the new defined function in a query statement after registering them.

You can use the given code to register the function.

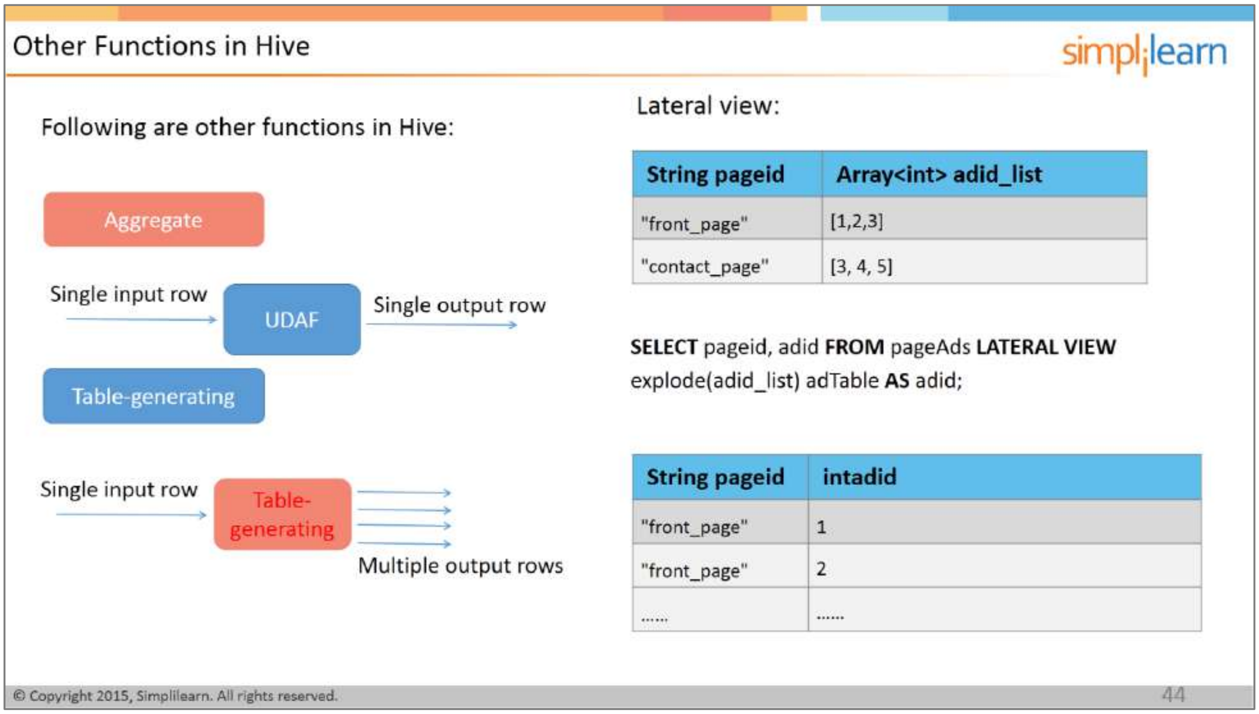


Built-In Functions

Hive provides many built-in functions such as:

- Mathematical,
- Collection,
- Type conversion,
- Date,
- Conditional, and
- String.

Take a look at the examples provided for each Built-in functions.



Other Functions in Hive

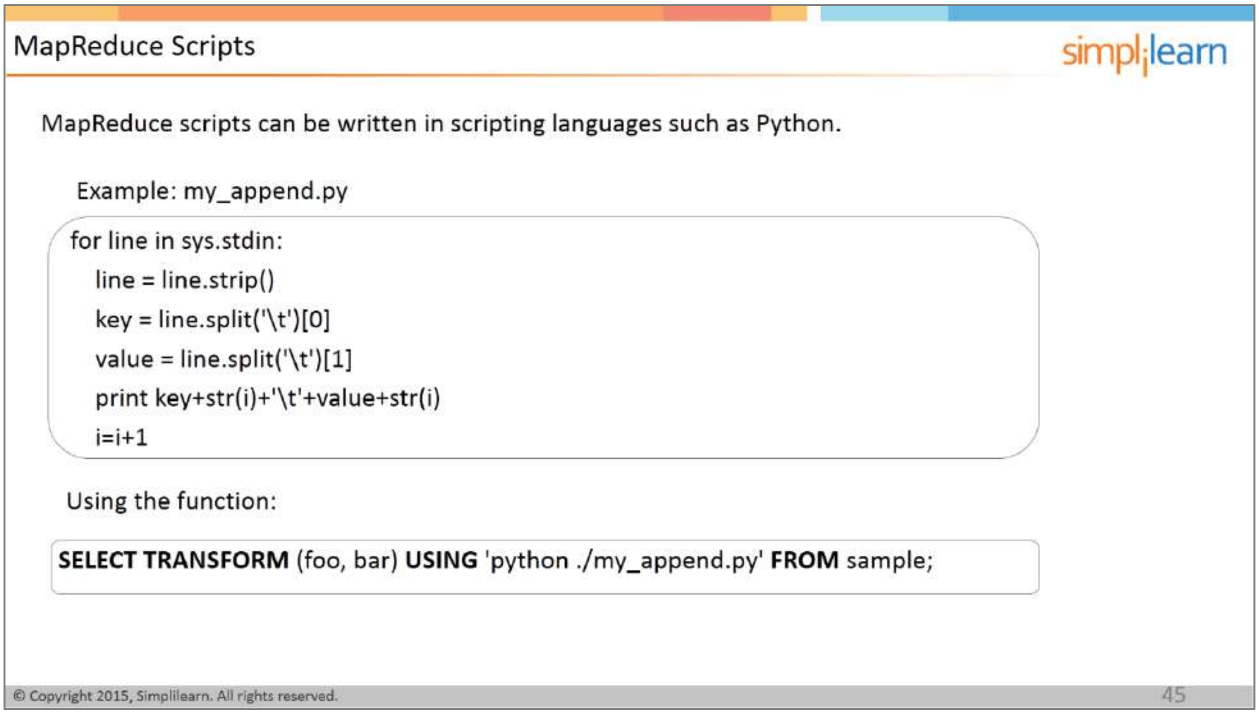
There are other functions in Hive, such as the aggregate function and the table-generating function.

Aggregate functions create the output if the full set of data is given. The implementation of these functions is complex compared with that of the UDF. The user should implement a few more methods, however, the format is similar. Therefore, Hive provides many built-in User-Defined Aggregate Functions or UDAF.

Normal user-defined functions, namely concat(), take in a single input row and give out a single output row. In contrast, table-generating functions transform a single input row to multiple output rows.

Lateral view is used in conjunction with table generating functions. An SQL script in lateral view is displayed.

Consider the base table named pageAds. It contains two columns: pageid, which is the name of the page and adid_list, which is an array of ads appearing on the page. A lateral view with explode() can be used to convert the adid_list into separate rows using the given query.



MapReduce Scripts

MapReduce scripts is written in scripting languages such as Python.

Users can plug in their own custom mappers and reducers in the data stream. To run a custom mapper script and reducer script, the user can issue a command which uses the TRANSFORM clause to embed the mapper and the reducer scripts.

Look at the script in the image. Key-value pairs will be transformed to STRING and delimited by TAB before feeding to the user script by default.

The method strip returns a copy of all of the words in which whitespace characters have been stripped from the beginning and the end of the word.

The method split returns a list of all of the words using TAB as the separator.

UDF/UDAF vs. MapReduce Scripts

The table shows the comparison of User-defined and User-defined aggregate functions with MapReduce scripts.

Attribute	UDF/UDAF	MapReduce scripts
Language	Java	Any language
1/1 input/output	Supported via UDF	Supported
n/1 input/output	Supported via UDAF	Supported
1/n input/output	Supported via UDTF	Supported
Speed	Faster (in same process)	Slower (spawns new process)

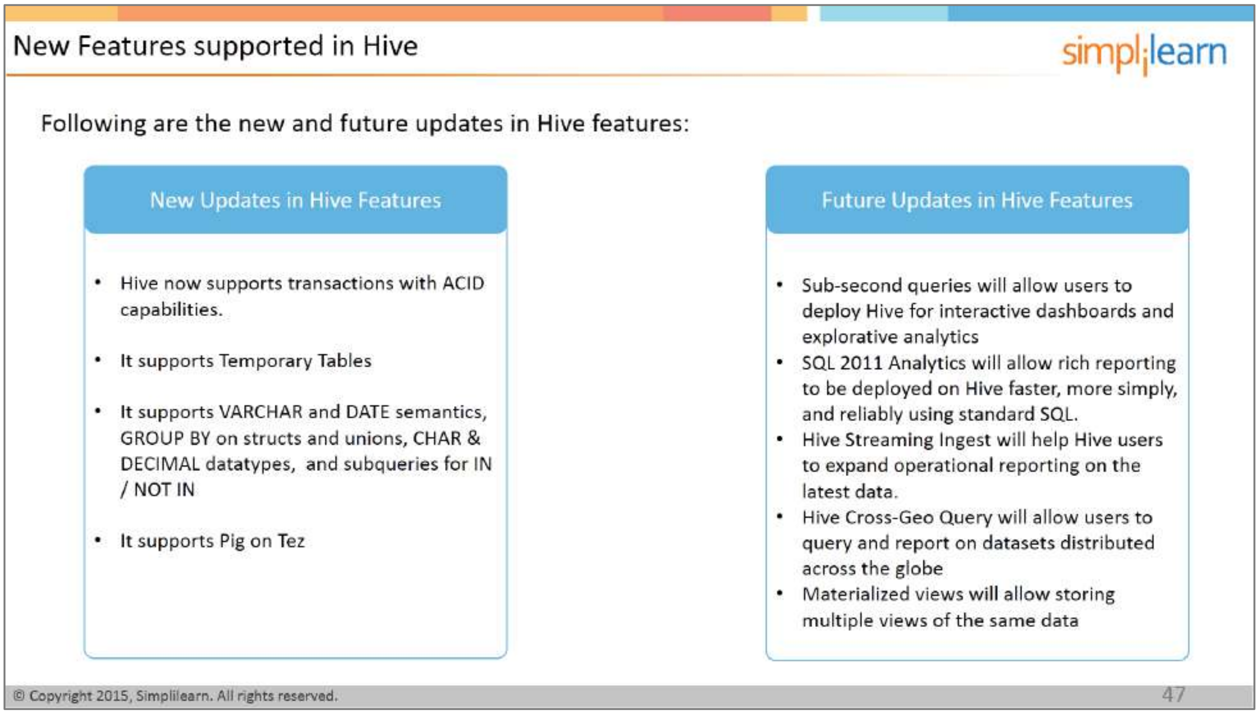
 46

UDF/UDAF vs. MapReduce Scripts

User-defined functions are written in Java while MapReduce scripts can be written in any language.

Both user-defined functions and MapReduce scripts support 1 to 1, n to 1, and 1 to ‘n’ input to output.

However, user-defined functions are much faster than MapReduce scripts since the latter spawns new processes for different operations.



New Feature supported in Hive

Let's look at the new features and future updates in Hive.

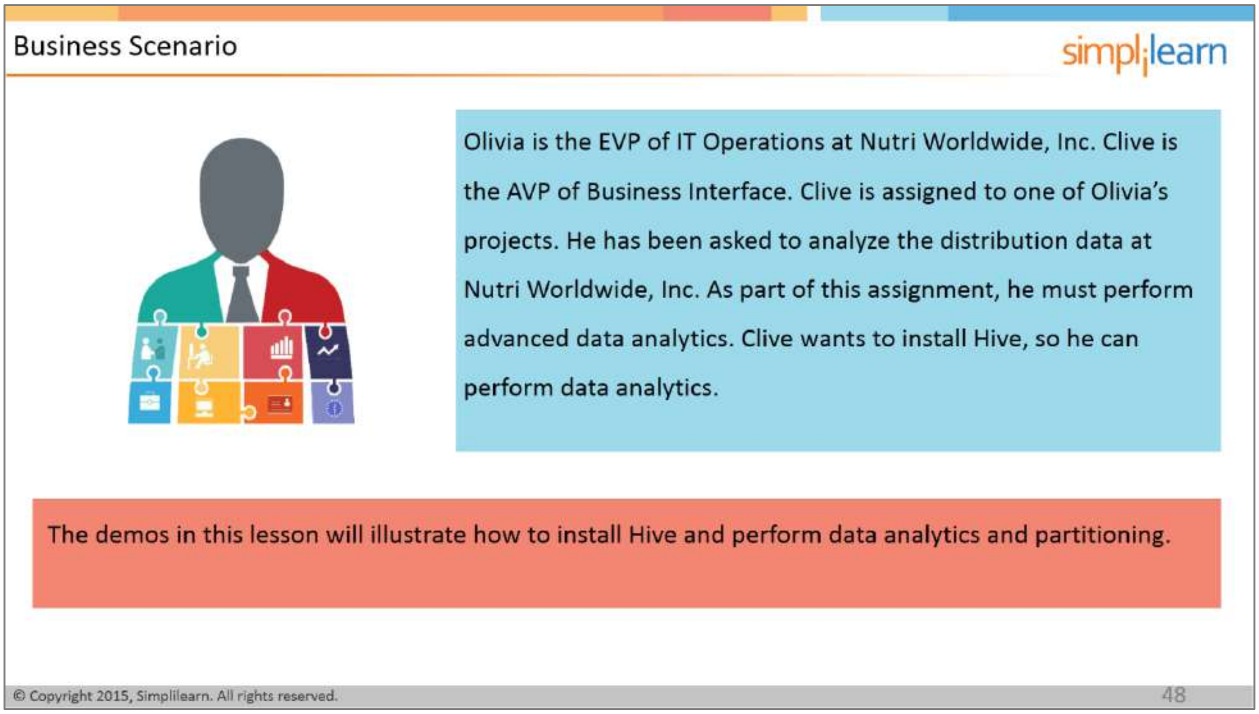
New Updates:

- Hive now supports transactions with ACID capabilities. This helps in streaming and baseline update for Hive such as modifying dimension tables or other fact tables.
- This Cost Based Optimizer will ensure Apache Hive stays as the de facto standard for SQL on Hadoop.
- It supports Temporary Tables.
- It supports VARCHAR and DATE semantics, GROUP BY on structs and unions, CHAR & DECIMAL datatypes, and subqueries for IN / NOT IN.
- It supports Pig on Tez.

Future Updates:

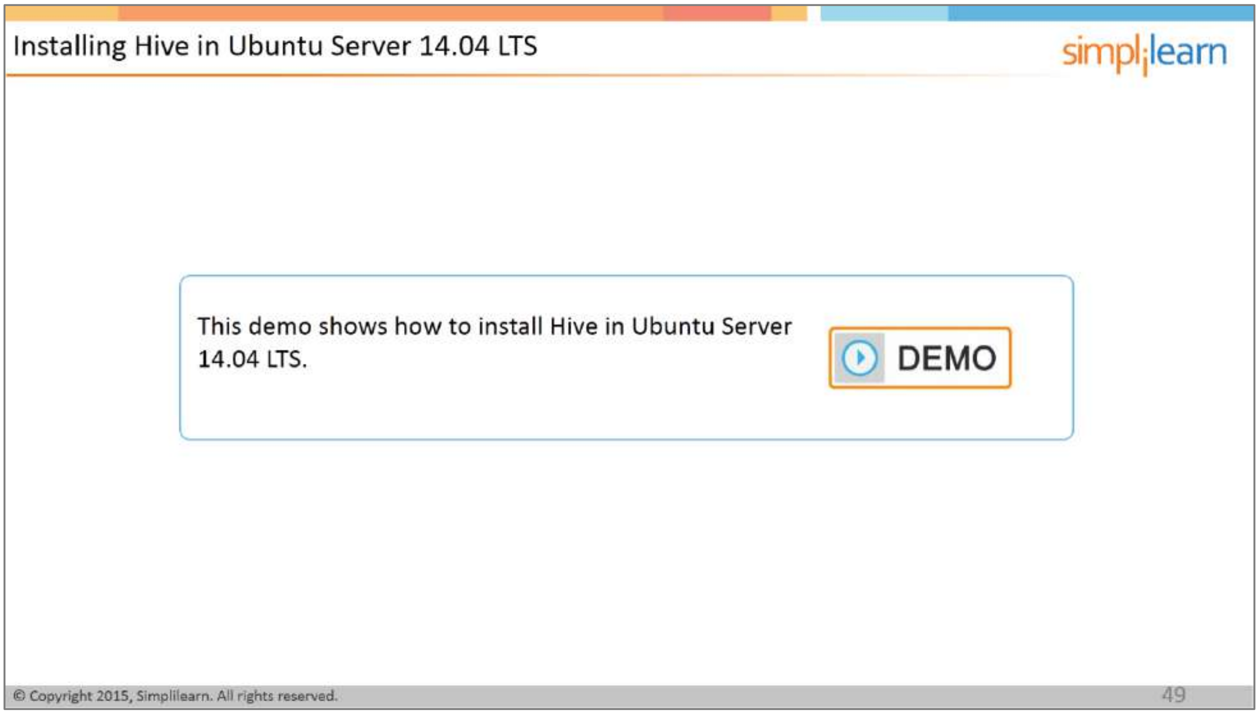
- Sub-second queries will allow users to deploy Hive for interactive dashboards and explorative analytics that have more demanding response-time requirements.
- SQL 2011 Analytics will allow rich reporting to be deployed on Hive faster, more simply and reliably using standard SQL.

- Hive Streaming Ingest will help Hive users to expand operational reporting on the latest data.
- Hive Cross-Geo Query will allow users to query and report on datasets distributed across the globe due to legal or efficiency constraints.
- Materialized views will allow storing multiple views of the same data allowing faster analyses.



Business Scenario

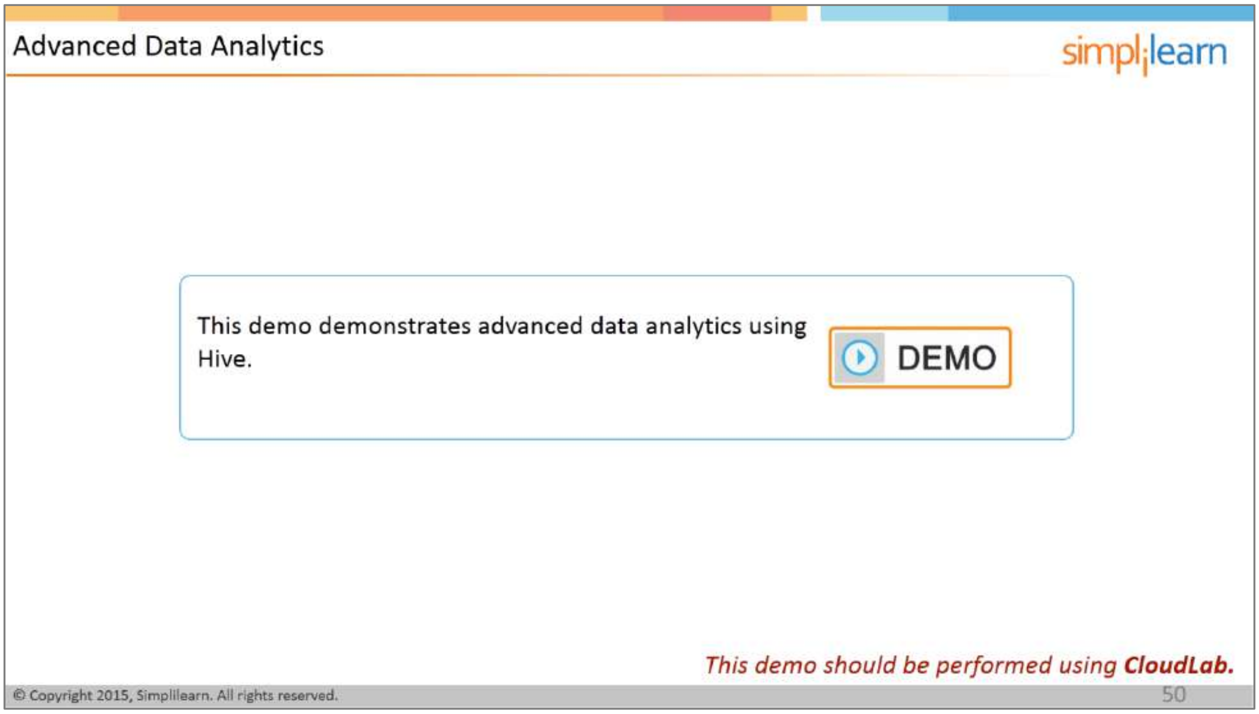
Olivia is the EVP of IT Operations at Nutri Worldwide, Inc. Clive is the AVP of Business Interface. Clive is assigned to one of Olivia's projects. He has been asked to analyze the distribution data at Nutri Worldwide, Inc. As part of this assignment, he must perform advanced data analytics. Clive wants to install Hive, so he can perform data analytics.



Installing Hive in Ubuntu

1. To download Hive in your browser, go to the specified URL.
2. Click Download a release now!
3. Select stable release under the Parent Directory.
4. Right click the bin tar file and copy the link location.
5. Let's download Hive in the Ubuntu machine.
6. Use wget command and enter the link location of Hive download.
7. The download time may vary on network and other infrastructure conditions.
8. Use tar xvzf command to unpack the file.
9. Verify unpacking by using ls-l command.
10. You will notice a folder by the name apache-hive followed.
11. You may clear the screen by using clear command.
12. Now create a soft link for the hive directory.
13. Use sudo ln –s command to create a soft link at /usr/local/hive.
14. List the location typed here to check whether a soft link has been created or not.
15. Now, create the Hive's warehouse directory in HDFS. Create it using the command typed here at the location shown.
16. Change the permission of HDFS tmp directory.
17. Use hadoop fs-chmod command.
18. Change permissions of the mentioned directory.

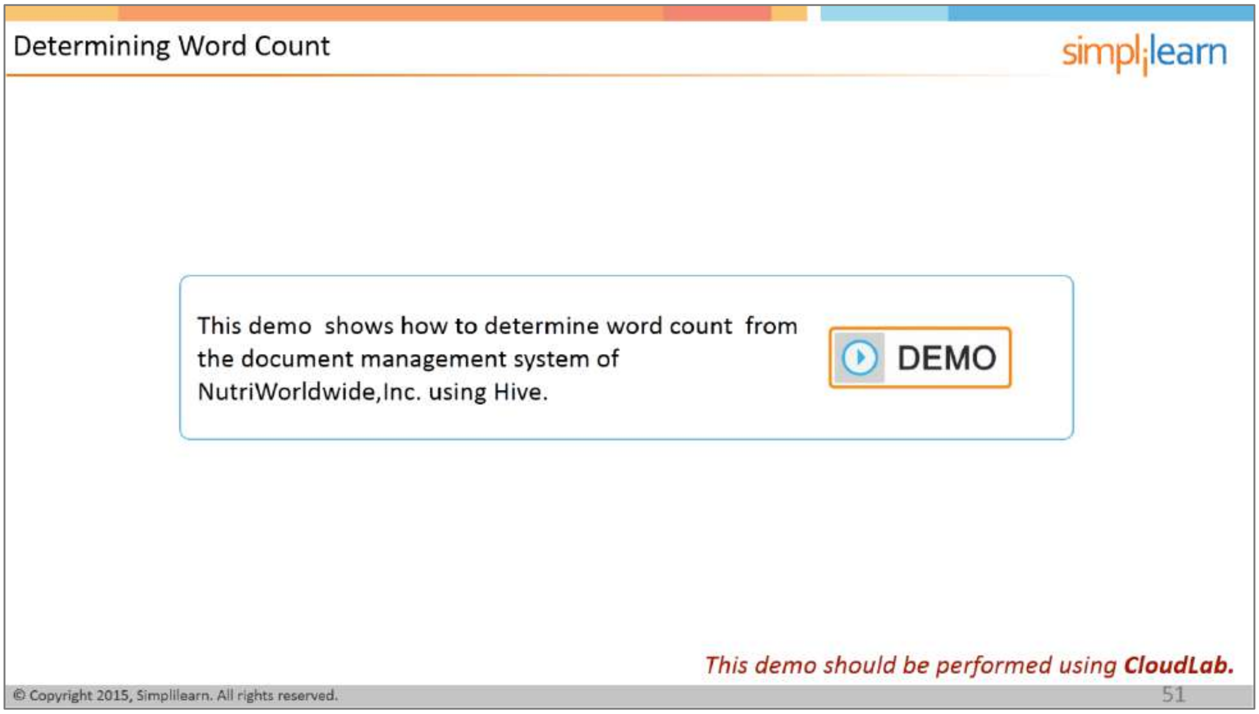
19. Let's make path and other environment variables setting in the file mentioned here.
20. Enter HIVE_HOME as home directory of Hive.
21. Add Hive's bin directory to the path.
22. Save the file using Control and O key if you are using nano editor in Ubuntu.
23. Close the file.
24. Implement file changes using source command
25. Launch Hive by typing hive on command prompt.
26. You will notice a hive prompt. Enter show tables command on the prompt.
27. Enter quit to close the hive command line interface.
28. Enter hiveserver 2 now on command prompt to launch JDBC and ODBC server.
29. We will connect now to hive using Beeline client.
30. You need to mention host and port in JDBC URL.
31. In our instance host is 'localhost' and port is '1000'
32. Enter show tables command now in beeline.



Advanced Data Analytics

This demo shows advanced data analytics using Hive.

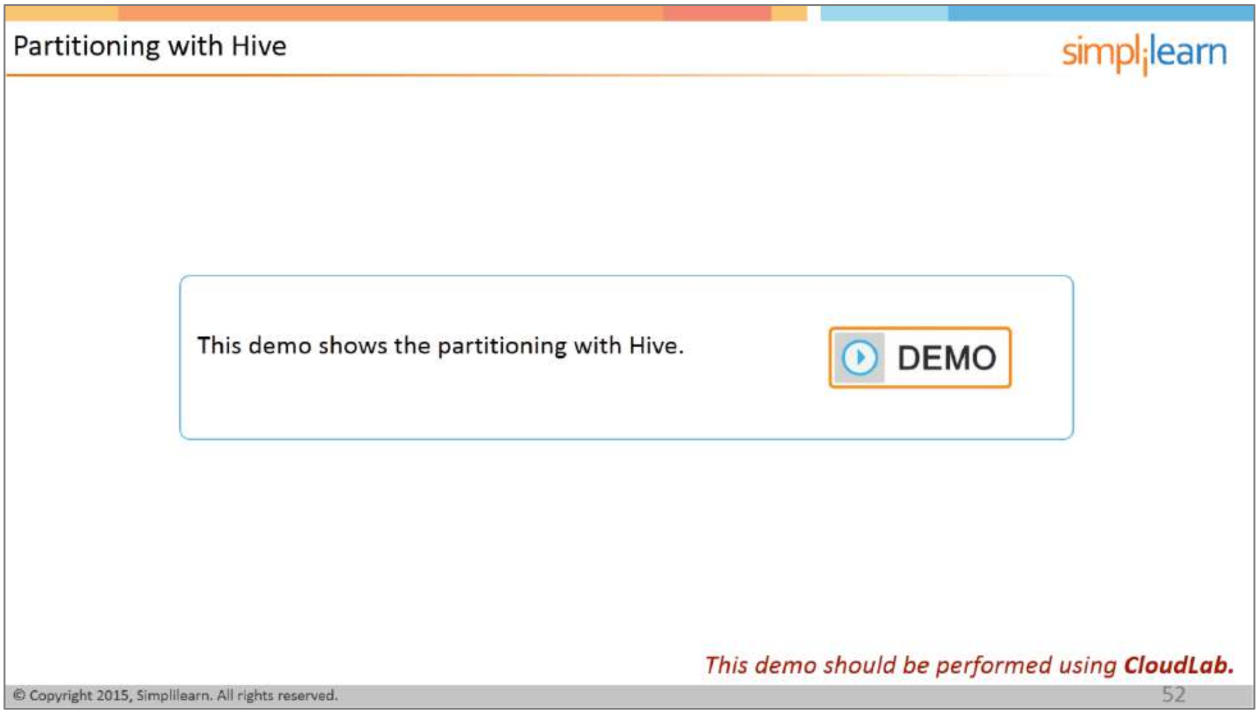
1. Create a table called ‘book’ with the datatype STRING.
2. Observe ROW FORMAT DELIMITED
3. Observe FIELDS TERMINATED BY ' '
4. Observe LINES TERMINATED BY '\n'
5. Now let us load the data from the file into the table.
6. The command to load data is shown.
7. Type show tables; to verify the table
8. The highlighted part shows the table book.
9. Let us verify the schema of the table book by typing describe book;.
10. The output describes the table book.
11. Type select lower (word), count(*) as count.
12. Type from book.
13. Observe Where lower(substring(word,1,3))='was'
14. Observe Group by word and press Enter key.
15. Observe Having count > 50
16. Observe Sort by count desc;
17. Observe the job execution.



Determining Word Count

This demo shows how to determine word count from the document management system of NutriWorldwide, Inc. using Hive.

1. To begin with, launch Hive on your system.
2. Create a table named ‘Book’ with a single column, textword, to store the contents.
3. The file is saved in the text file format.
4. The table, ‘Book’ is created once you press enter after semi colon on Hive prompt.
5. Load the book, Gutenberg.txt, from the HDFS path.
6. The contents get uploaded in the table book in Hive warehouse.
7. Query the table for determining the word count by using the Select statement.
8. Use the Select query with count operator.
9. Use lateral View and user-defined functions to split the contents into space-separated words.
10. A lateral view first applies the user-defined function to each row of the base table and then joins the resulting output rows to the input rows to form a virtual table having the supplied table alias.
11. To put it simply, this line will put each word on a separate record of lateral view.
12. Now, group by word so that the count operator can give the result by each distinct word.
13. Execute the query and observe the result with each word showing the count of times it occurs in the document.



Partitioning with Hive

This demo demonstrates partitioning with Hive.

1. To begin with, launch Hive on your system.
2. Create a table 'president'.
3. Mention the two columns: s-n-o which means serial number and 'line' which includes the president name per line.
4. Partition by the parameter 'country' which is of type string.
5. Use the usual Hive syntax 'Row format delimited'.
6. The fields are terminated by a comma since the data is comma-separated.
7. Execute the query and use the describe command to verify if the table created matches the requirement.
8. You will observe that the table has been created successfully.
9. Let us now load data in the table.
10. Load data in the table 'president' using the LOAD DATA INPATH command since the data is already present in HDFS.
11. You will observe an error if the partition is not specified.
12. The reason for the error is that we did not specify the partition in which data is to be loaded.
13. Let us try to rectify the error by using the clause partition 'country = USA' to ensure that data gets loaded in the right partition of the table.
14. You will observe that query has been successfully executed.

15. Let us check the data in HDFS to see how Hive tables and partitions are created.
16. From the Hive prompt itself, use the command shown to view the table directory.
17. Please note that the directory location may vary with Hive and Hadoop distribution version.
18. List the directory for the partition ‘country=USA’ partition’.
19. Note that a directory is created for the table ‘president’. A directory is also created for the partition ‘country equals USA’.
20. Let us now load the data for some other countries also.
21. Let us load the data for the partition ‘country equals India’.
22. Now load the data for the partition ‘country equals Russia’.
23. Check how the various partitions have been created in HDFS.
24. Use the d-f-s minus l-s command again from the Hive prompt.
25. You will observe that a directory for each partition has been created. You can list the content of each directory to view the data files present in every partition directory.

Spot the Error

Susan, a Sales Manager of a finance company, had to segregate the USA sales record from the 'salesrecord' table. She loaded the table successfully. However, she was confused with the partitioning syntax. Can you assist Susan by identifying the partitioning syntax?

Spot the correct syntax.

```
hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord ;

hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord partition (country='USA');

hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord (country='USA');
```


Spot the Error

The incorrect syntax are: `hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord ;`

And

`hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord (country='USA');`

The correct syntax is: `hive> LOAD DATA INPATH '/sales/salesrecord .txt' OVERWRITE INTO TABLE salesrecord partition (country='USA');`

simplilearn

QUIZ 1

Which command is used for the creation of tables in Hive?

a. CREATE TABLE

b. SELECT

c. DROP

d. LOAD

 57

simplilearn

QUIZ 2

Which command is used for the deletion of tables in Hive?

a. CREATE TABLE

b. SELECT

c. DROP TABLE

d. LOAD

 59

simplilearn

QUIZ 3

Which command is used for listing data in a table in Hive?

a. CREATE TABLE

b. SELECT

c. DROP TABLE

d. LOAD

 61

simplilearn

QUIZ 4

Which component stores the system catalog and metadata about tables, columns, and partitions in Hive?

a. CLI

b. Metastore

c. Driver

d. Thrift Server

 63

QUIZ 5

Which component manages the lifecycle of a HiveQL statement?

a. CLI

b. Metastore

c. Driver

d. Thrift Server

QUIZ 6

Which component compiles HiveQL into a directed acyclic graph of MapReduce tasks?

a. CLI

b. Query compiler

c. Driver

d. Thrift Server

simplilearn

QUIZ 7

Which component in Hive executes the tasks produced by a compiler in the proper dependency order?

a. CLI

b. Query compiler

c. Execution engine

d. Thrift Server

simplilearn

QUIZ 8

Which component in Hive provides a thrift interface, a JDBC/ODBC server, and a way of integrating Hive with other applications?

a. CLI

b. Query compiler

c. Execution engine

d. Thrift Server

QUIZ 9	Which of the following components in Hive contains the compiler, optimizer, and executor?

a. Driver

b. Query compiler

c. Execution engine

d. Thrift Server

QUIZ 10	Which of the following components can be used to accept command inputs from the user?

a. CLI

b. Query compiler

c. Execution engine

d. Thrift Server

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which command is used for the creation of tables in Hive?	a. The CREATE TABLE command is used to create a table in Hive.
2	Which command is used for the deletion of tables in Hive?	c. The DROP TABLE command is used to delete a table in Hive.
3	Which command is used for listing data in a table in Hive?	b. The SELECT command is used to list data in a table in Hive.
4	Which component stores the system catalog and metadata about tables, columns, and partitions in Hive?	b. The Metastore component stores the system catalog and metadata.
5	Which component manages the lifecycle of a HiveQL statement?	c. The Driver component in Hive manages the lifecycle of a HiveQL statement.

ANSWERS:

S.No.	Question	Answer & Explanation
6	Which component compiles HiveQL into a directed acyclic graph of MapReduce tasks?	b. The Query compiler is used to compile HiveQL into a directed acyclic graph of MapReduce tasks.
7	Which component in Hive executes the tasks produced by a compiler in the proper dependency order?	c. The execution engine component in Hive executes the tasks produced by the compiler in the proper dependency order.
8	Which component in Hive provides a thrift interface, a JDBC/ODBC server, and a way of integrating Hive with other applications?	d. The Thrift Server component in Hive provides a thrift interface, a JDBC/ODBC server, and a way of integrating Hive with other applications.
9	Which of the following components in Hive contains the compiler, optimizer, and executor?	a. The driver component in Hive contains the compiler, optimizer, and executor.
10	Which of the following components can be used to accept command inputs from the user?	a. CLI is used as an input medium to accept command inputs from the user.

Case Study

simplilearn

Scenario | Analysis | Solution

XY Networks provides network security support to lot of organizations. It has system generated log files critical for security analysis and monitoring. The company has chosen to implement Hadoop to store the log files. But XY's data scientists are finding it difficult to use Hadoop MapReduce as they are not Java programmers. For any data analysis, they have to run to the Java developers to code the MapReduce programs. In case there are any mistakes, they have to send the programs back to developers. Data Scientists then find that Hadoop provides a tool called Hive, which provides an SQL interface to structured data in HDFS. Data scientists are well versed in using SQL.

hadoop

 78

Case Study

simplilearn

Scenario | Analysis | Solution

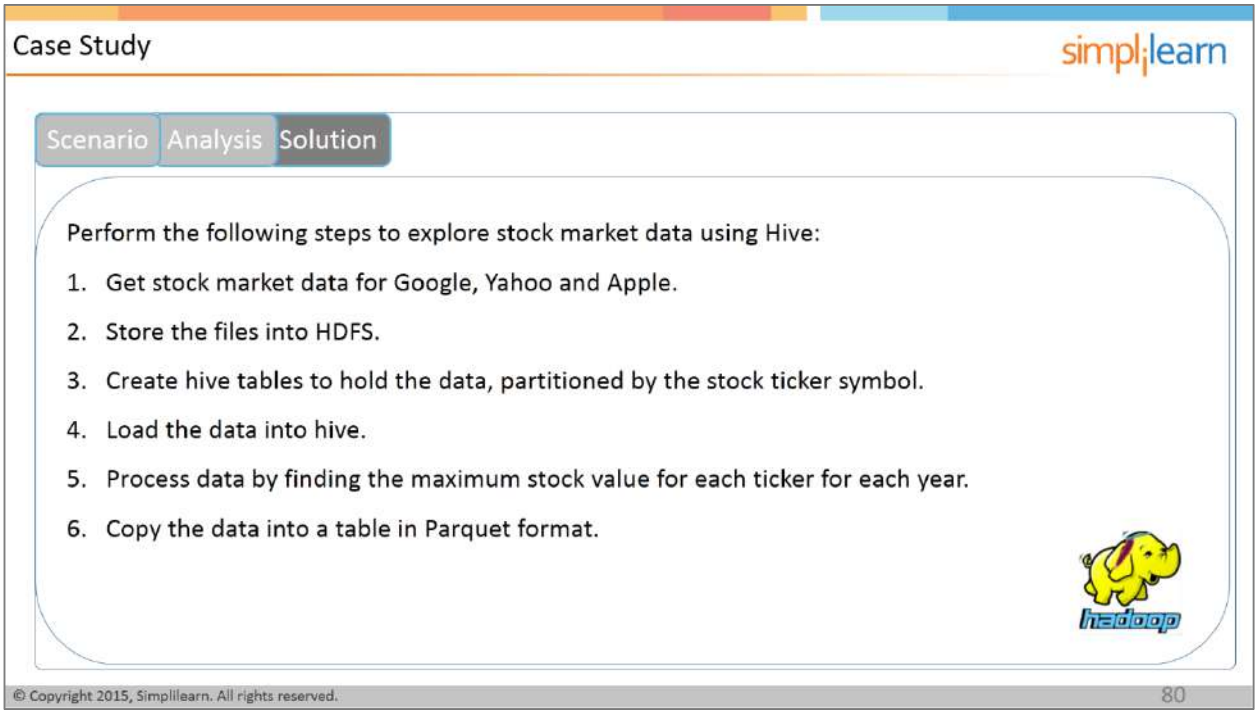
The team does research on Apache Hive and find that it is popular among data scientists for analyzing structured big data.

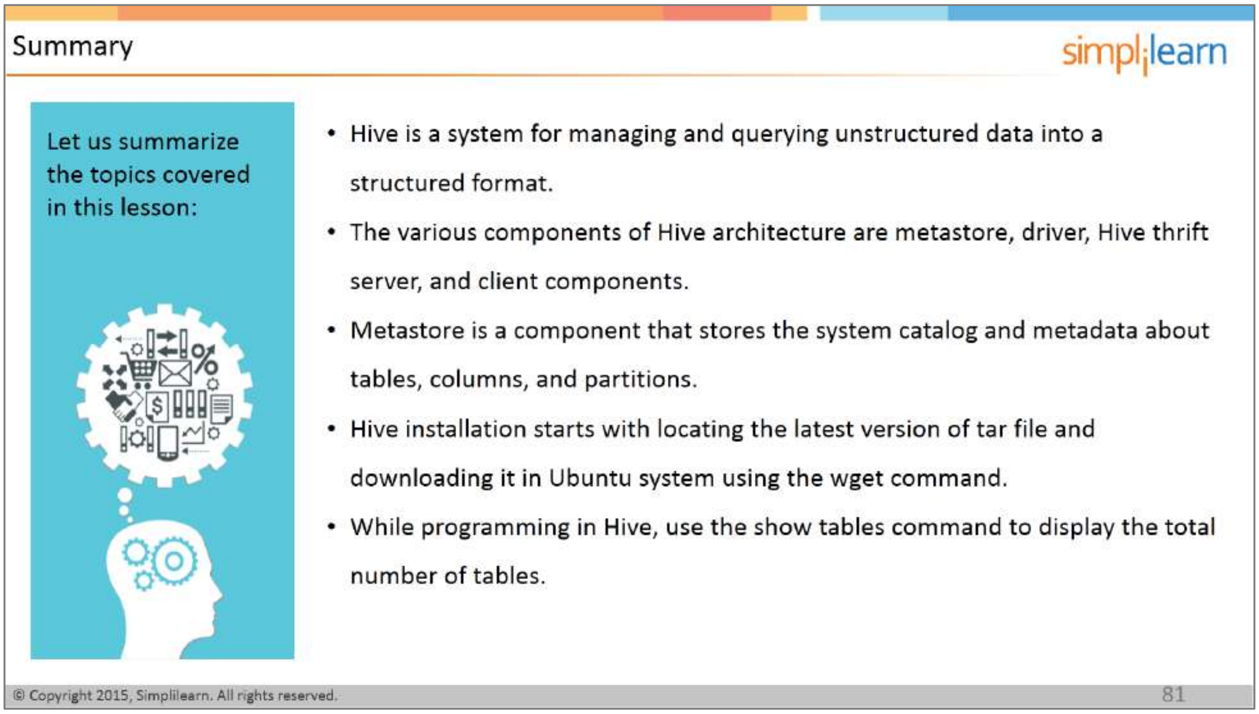
Advantages of Hive over using MapReduce:

1. No need for Java programming
2. Supports most of ANSI standard SQL
3. Provides GROUP BY and statistical functions for aggregating data
4. Provides joins of multiple sources of data which is difficult to do in MapReduce
5. Also supports Parquet format which is a columnar storage format that has become very popular

hadoop

 79

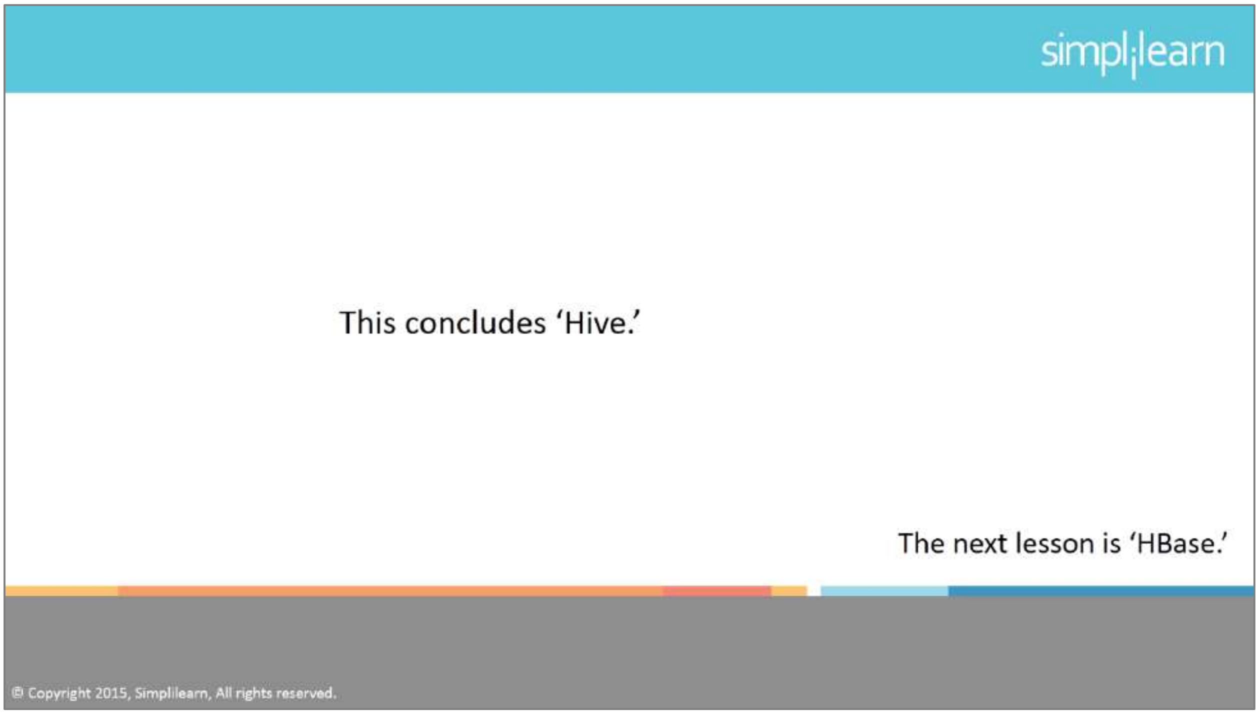

Case Study
simplilearn
Scenario
Analysis
Solution
Perform the following steps to explore stock market data using Hive:
1. Get stock market data for Google, Yahoo and Apple.
2. Store the files into HDFS.
3. Create hive tables to hold the data, partitioned by the stock ticker symbol.
4. Load the data into hive.
5. Process data by finding the maximum stock value for each ticker for each year.
6. Copy the data into a table in Parquet format.
hadoop
© Copyright 2015, Simplilearn. All rights reserved.
80



Summary:

Let us summarize the topics covered in this lesson.

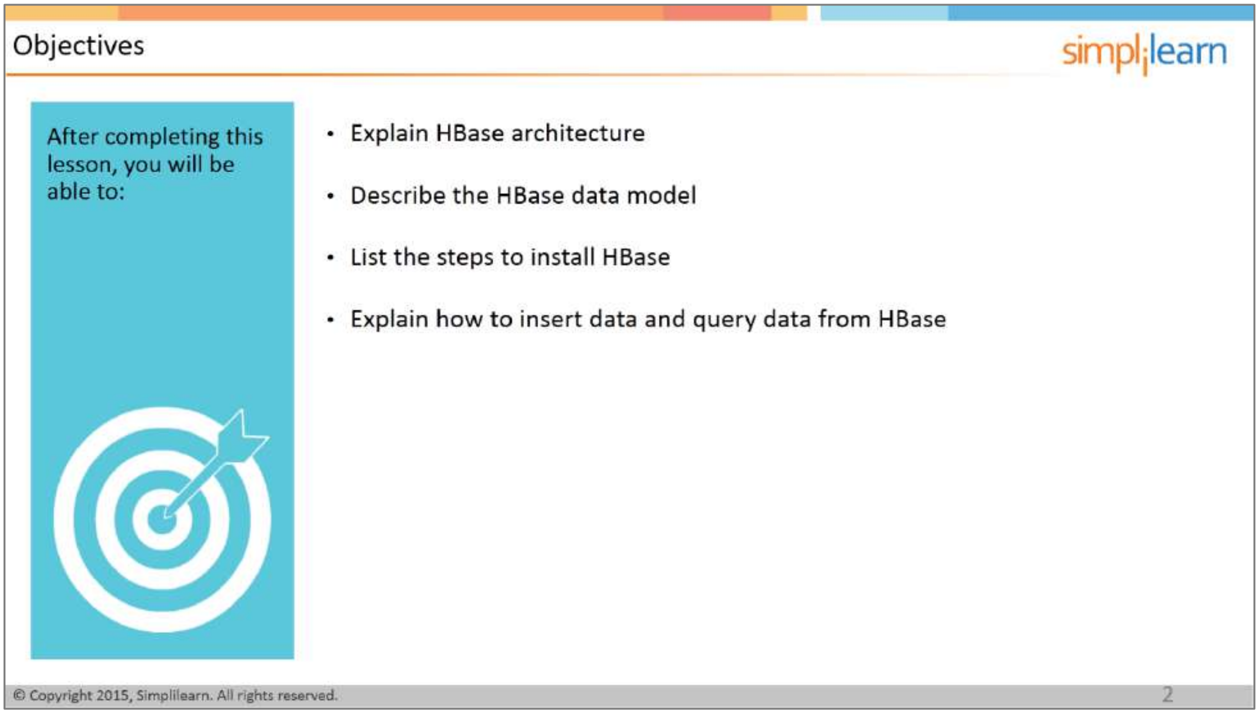
- Hive is a system for managing and querying unstructured data into a structured format.
- The various components of Hive architecture are Metastore, driver, Hive Thrift Server, and Client Components.
- Metastore is a component that stores the system catalog and metadata about tables, columns, and partitions.
- Hive installation starts with locating the latest version of tar file and downloading it in Ubuntu system using the wget command.
- While programming in Hive, use the show tables command to display the total number of tables.



Conclusion

This concludes 'Hive.' In the next lesson we will focus on 'HBase.'

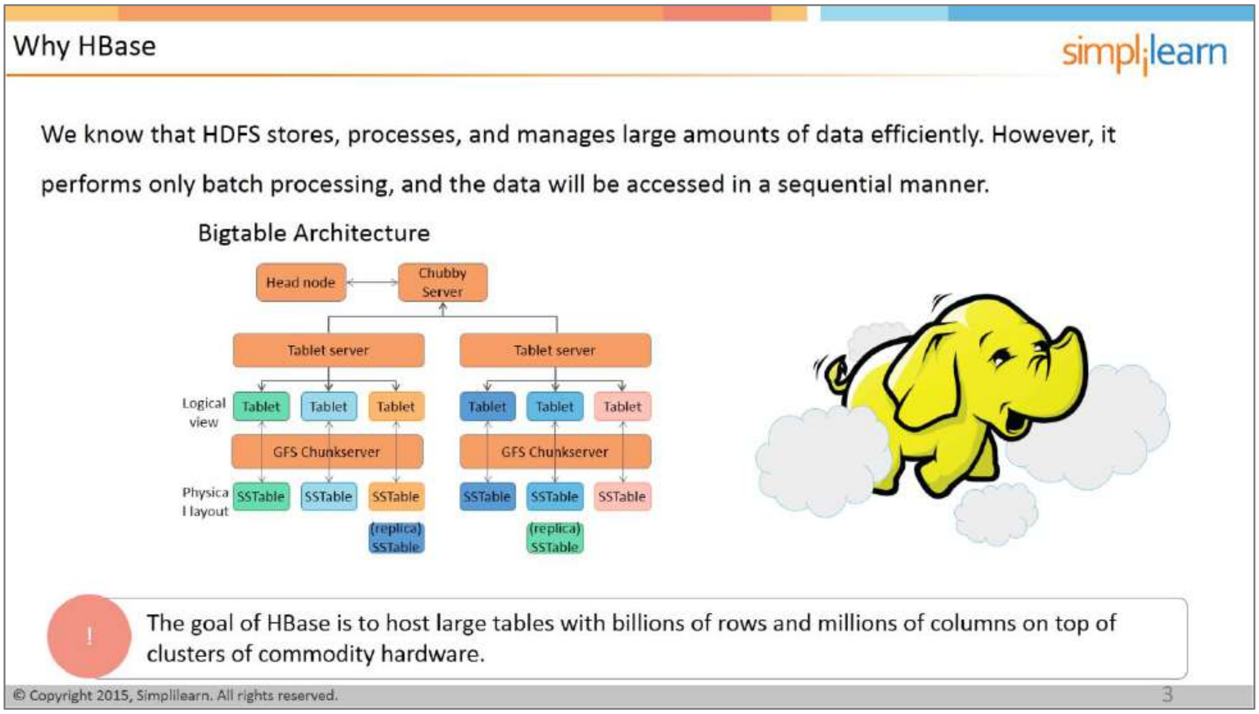
Lesson 8—HBase



Objectives

After completing this lesson, you will be able to:

- Explain the HBase architecture,
- Describe the HBase data model,
- List the steps to install HBase, and
- Explain how to insert data and query data from HBase.



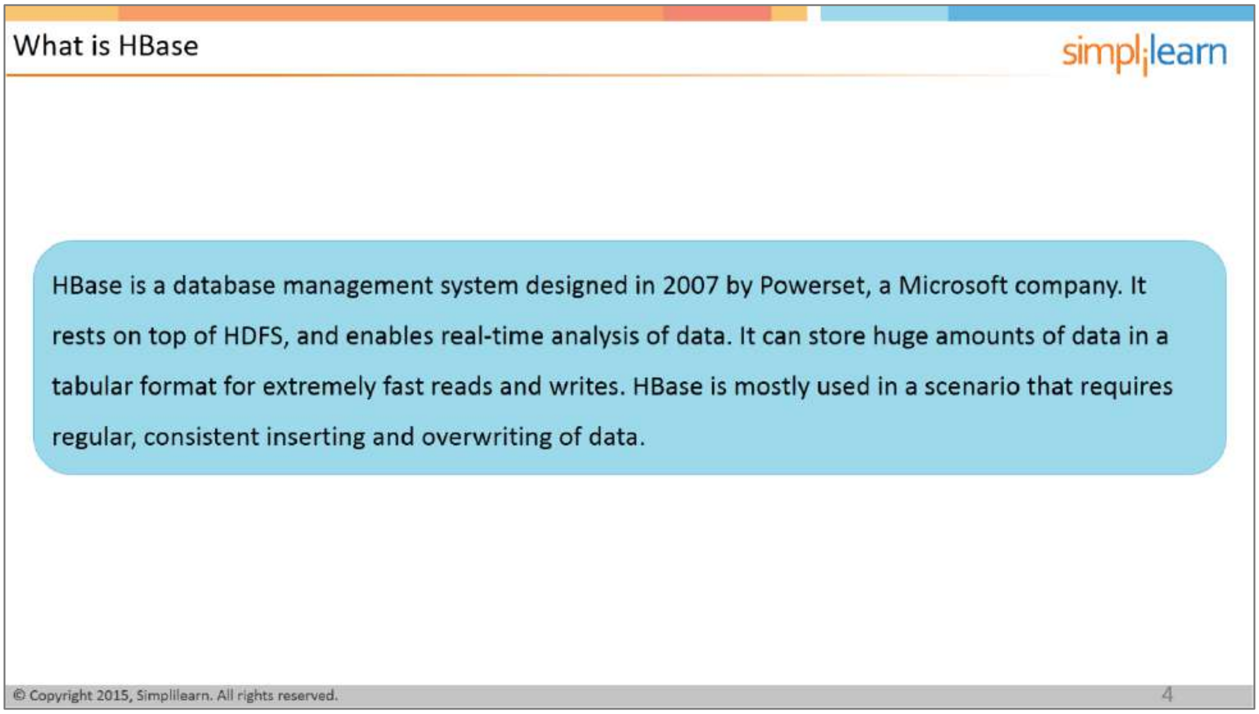
Why HBase

We know that HDFS stores, processes, and manages large amounts of data efficiently. However, it performs only batch processing, and the data will be accessed in a sequential manner. This means, one has to search the entire dataset for even the simplest of jobs.

Hence, a new solution was required to access, read, or write data any time regardless of its sequence in clusters of data. HBase is modeled after Google's Bigtable—a distributed storage system for structured data.

Just as Bigtable leverages the distributed data storage provided by the Google File System, Apache HBase provides Bigtable-like capabilities on top of Hadoop and HDFS.

The goal of HBase is to host large tables with billions of rows and millions of columns on top of clusters of commodity hardware.



What is HBase

HBase is a database management system designed in 2007 by Powerset, a Microsoft company. It rests on top of HDFS, and enables real-time analysis of data. It can store huge amounts of data in a tabular format for extremely fast reads and writes. HBase is mostly used in a scenario that requires regular, consistent inserting and overwriting of data.

HBase—Real-life Connect

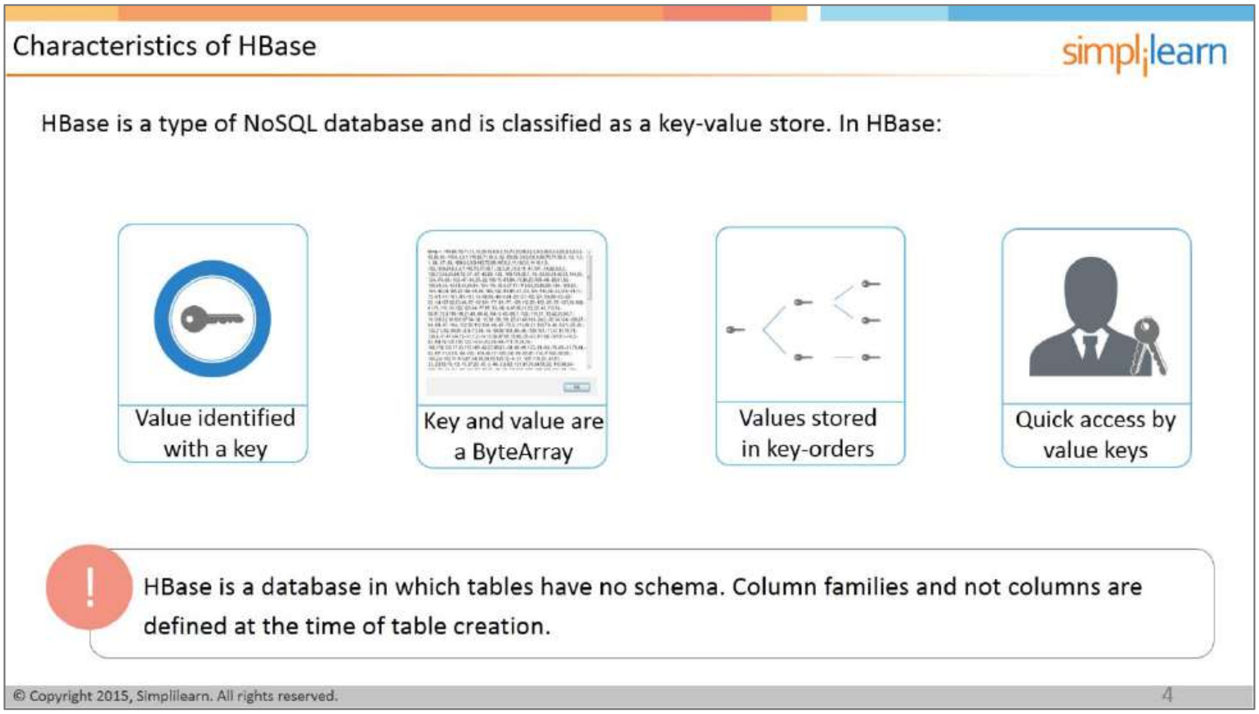
You would have read that Facebook has introduced a new Social Inbox integrating email, IM, SMS, text messages, and on-site Facebook messages. They need to store over 135 billion messages a month. Where can they store such data? The answer is HBase.

Facebook chose HBase because they needed a system that could handle two types of data patterns:

An ever-growing data set that is rarely accessed, and

Highly volatile

You read what's in your Inbox, and then you rarely look at it again. Makes sense?

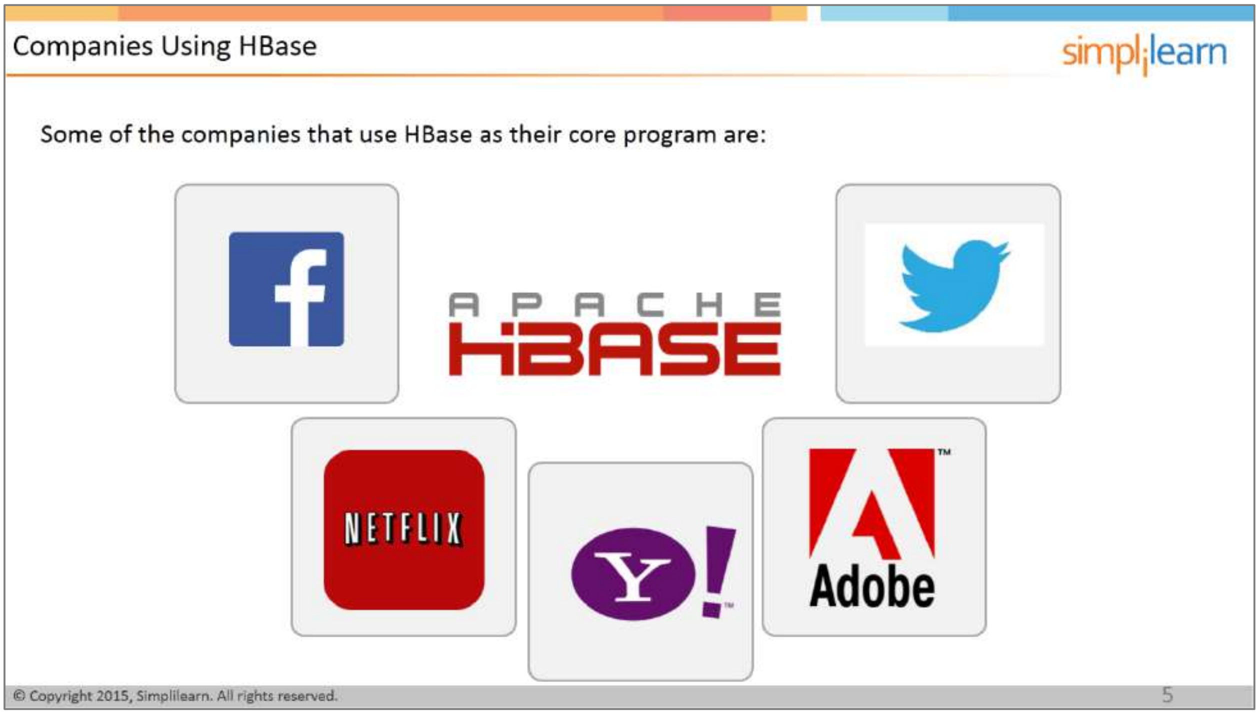


Characteristics of HBase

HBase is a type of NoSQL database and is classified as a key-value store.

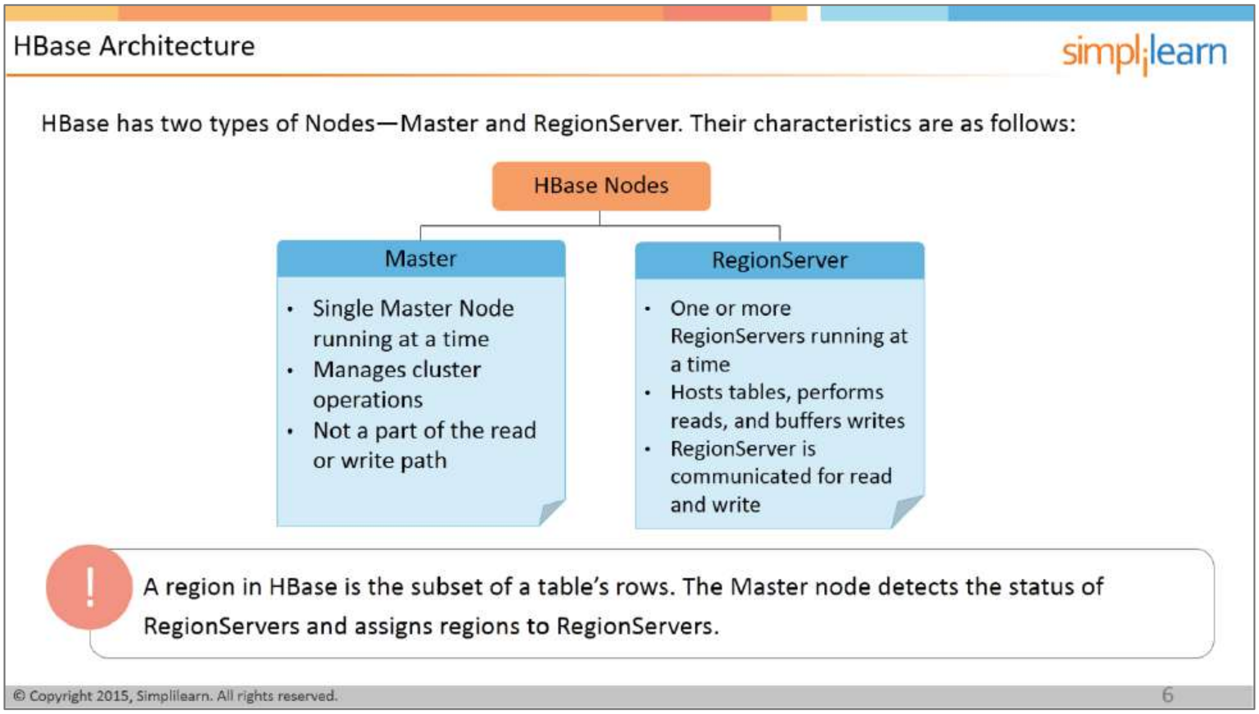
In HBase, value is identified with a key. Both key and value are ByteArray, which means binary formats can be stored easily. Values are stored in key-orders and can be quickly accessed by their keys.

HBase is a database in which tables have no schema. Column families and not columns are defined at the time of table creation.



Companies Using HBase

Some of the companies that use HBase as their core program are Facebook, Netflix, Yahoo, Adobe, and Twitter.



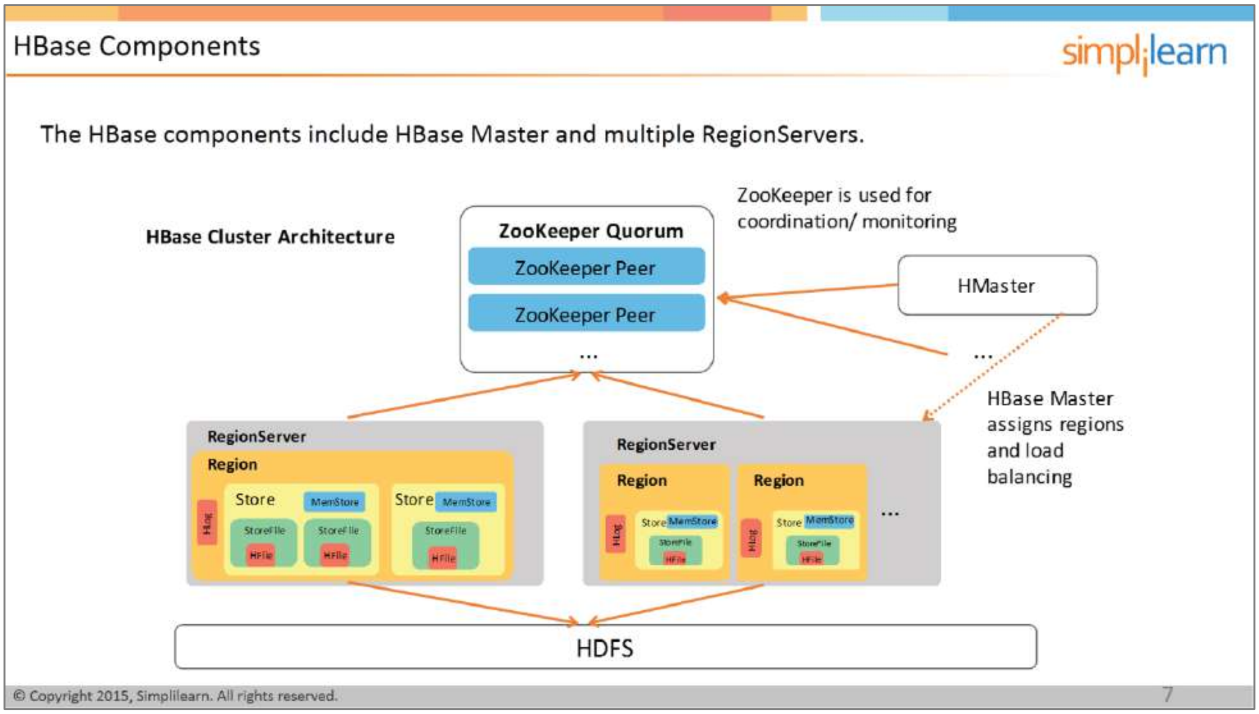
HBase Architecture

HBase has two types of Nodes: Master and RegionServer.

There is only one Master node running at a time whereas there can be one or more RegionServers. The high availability of the Master node is maintained by ZooKeeper. The Master node manages cluster operations like assignment, load balancing, and splitting. It is not a part of the read or write path.

The RegionServer hosts tables, performs reads, and buffers writes. Client communicates with the RegionServer for read and write.

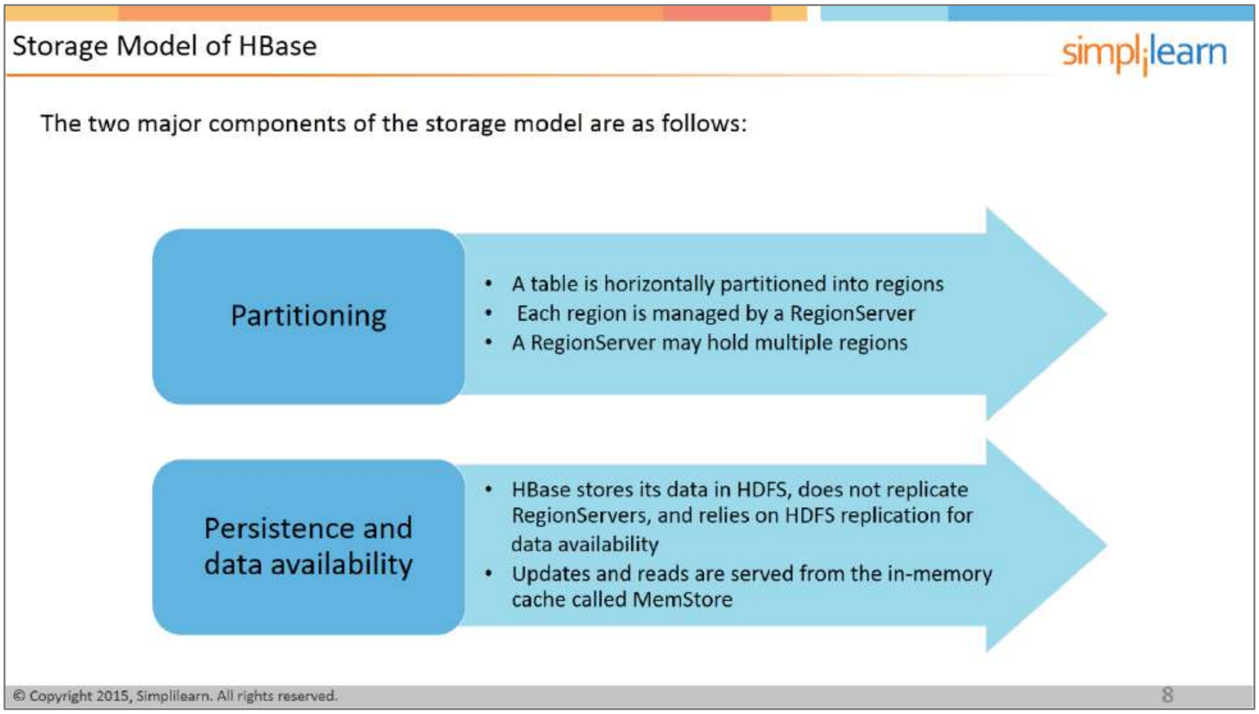
A region in HBase is the subset of a table's rows. The Master node detects the status of RegionServers and assigns regions to RegionServers.



HBase Components

The image represents the HBase components which include HBase Master and multiple RegionServers. The HBase Master is responsible for management of the schema stored in HDFS. RegionServers act like availability servers that enable the maintenance of a part of the complete data stored in HDFS based on the requirement of the user.

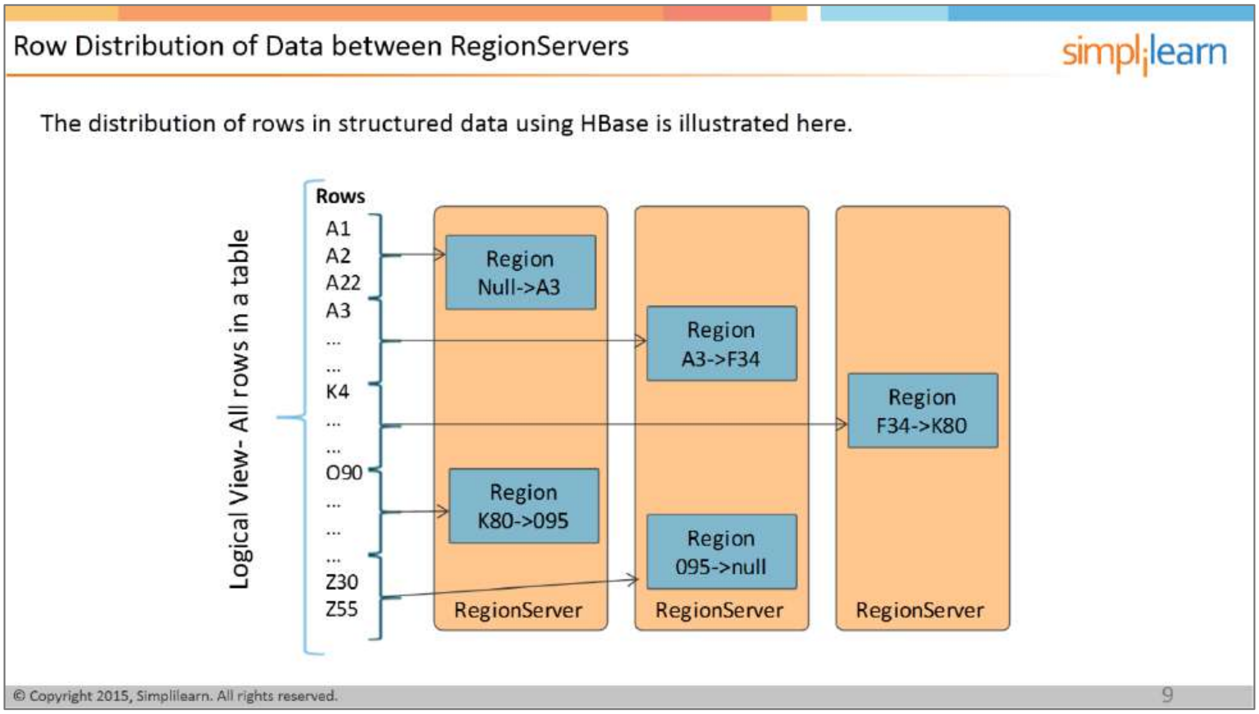
The RegionServers perform this task by using the HFile and Write Ahead Log or WAL service. The RegionServers always stay in sync with the HBase Master. It is the ZooKeeper that makes the RegionServers perform a stable sync with the HBase Master.



Storage Model of HBase

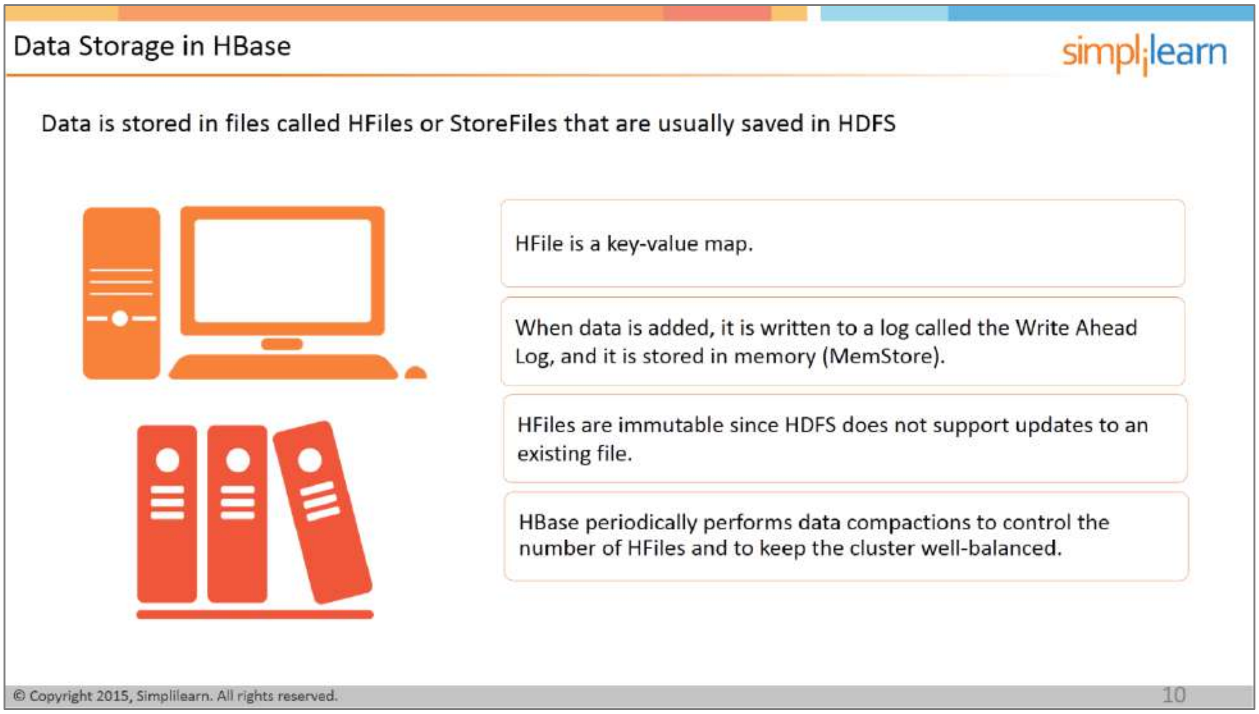
Partitioning is one part of the storage model of HBase, where a table is horizontally partitioned into regions. Each region is composed of a sequential range of keys. Each region is managed by a RegionServer. A RegionServer may hold multiple regions.

Persistence and data availability are also important components of the storage model. HBase stores its data in HDFS, does not replicate RegionServers, and relies on HDFS replication for data availability. The region data is first cached in memory. Updates and reads are served from the in-memory cache called MemStore. Periodically, MemStore is flushed to HDFS. WAL stored in HDFS, is used for the durability of updates.



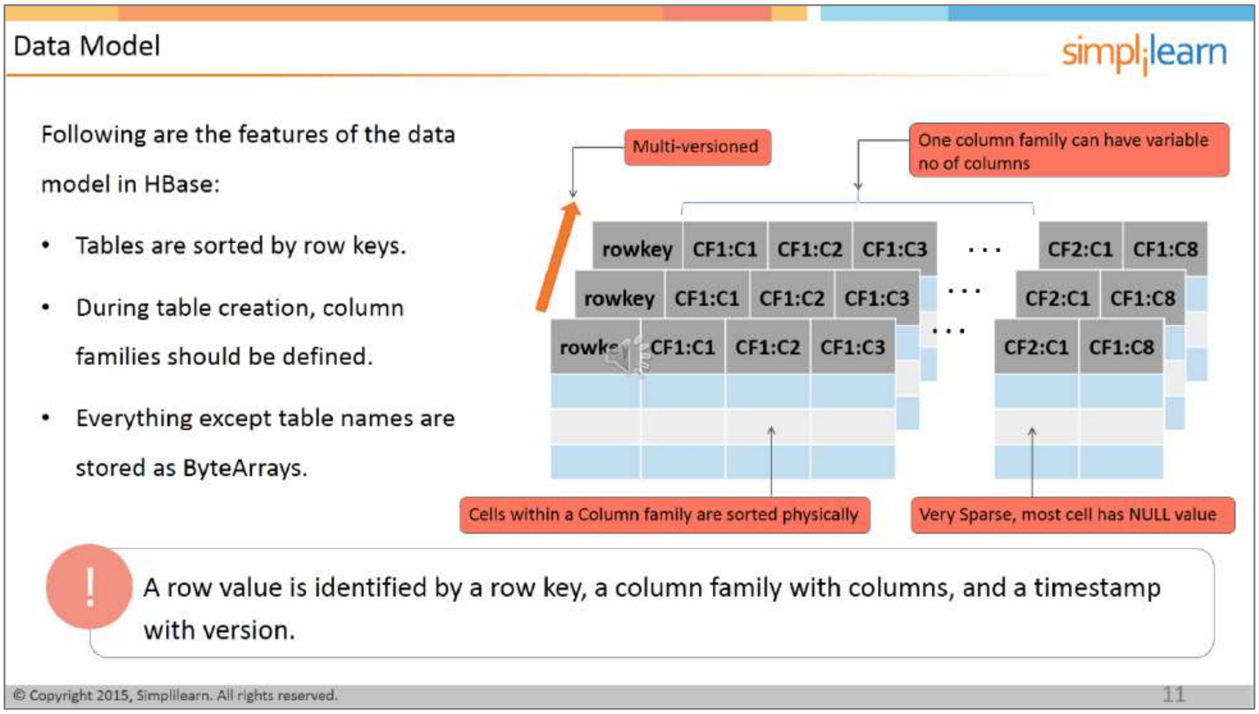
Row Distribution of Data between RegionServers

The image describes the distribution of rows in structured data using HBase. It shows how the data is sliced and maintained in individual RegionServers, depending on the requirement of the user. This type of distribution ensures availability of data to a specific user.



Data Storage in HBase

Data is stored in files called HFiles or StoreFiles that are usually saved in HDFS. An HFile is a key-value map generated due to the MapReduce operations performed by Hadoop. When data is added, it is written to WAL and stored in memory. This in-memory data store is called MemStore. HFiles are immutable since HDFS does not support updates to an existing file. To control the number of HFiles and keep the cluster well-balanced, HBase periodically performs data compactions.

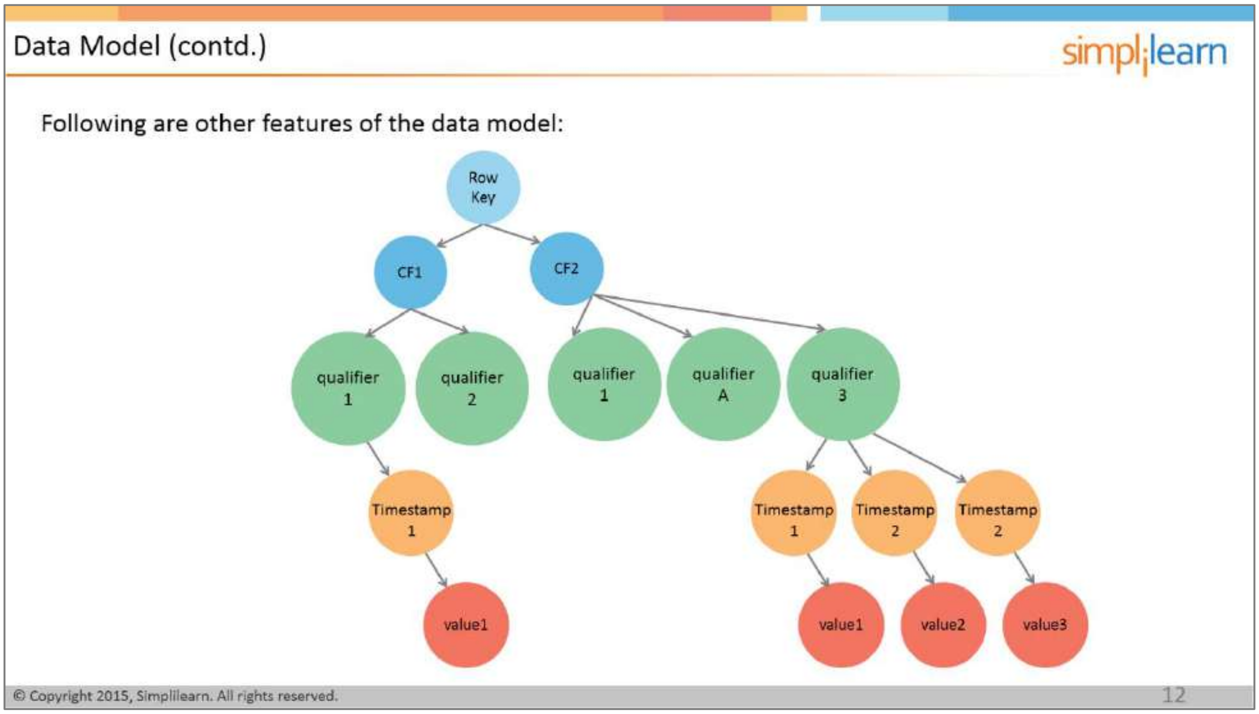


Data Model

Features of the data model in HBase are the following:

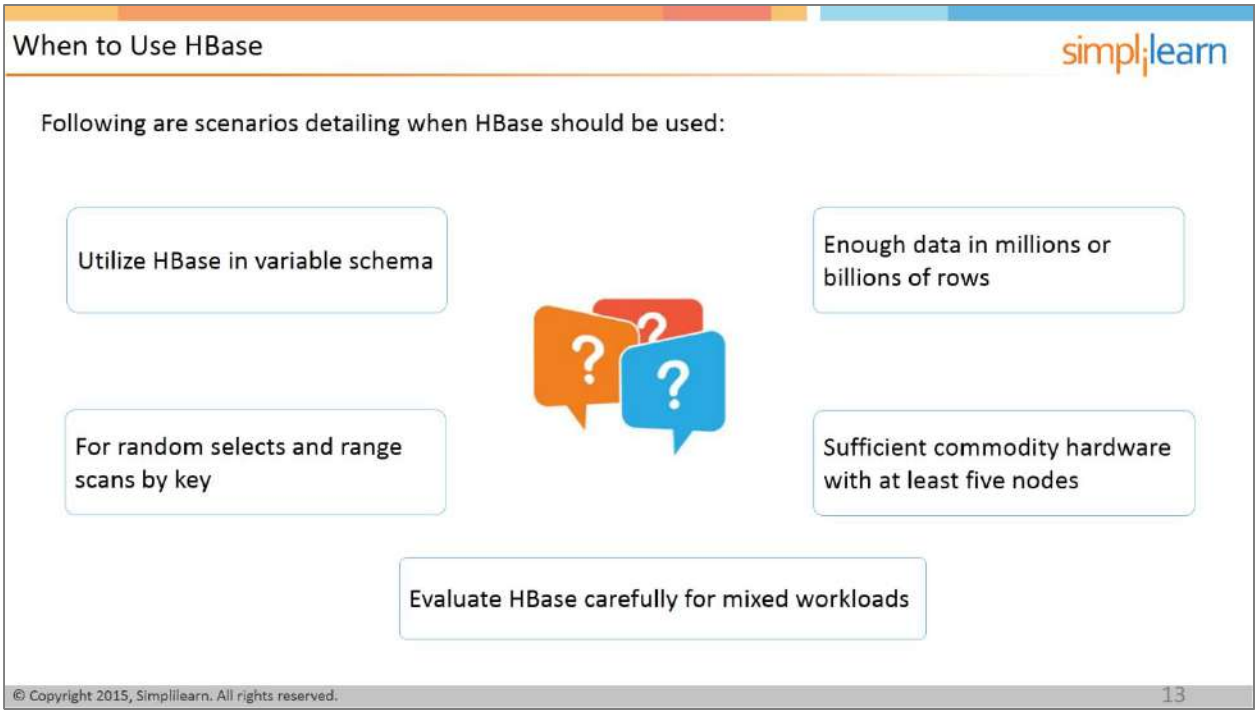
In HBase, all tables are sorted by row keys. At the time of table creation, you need to define only its column families. Each family may consist of any number of columns and each column consists of any number of versions. Columns only exist when inserted; however, NULLs are free. Columns within a family are sorted and stored together. Everything, except table names, is stored as a ByteArray.

A row key, a column family with columns, and a timestamp with version identify a row value.



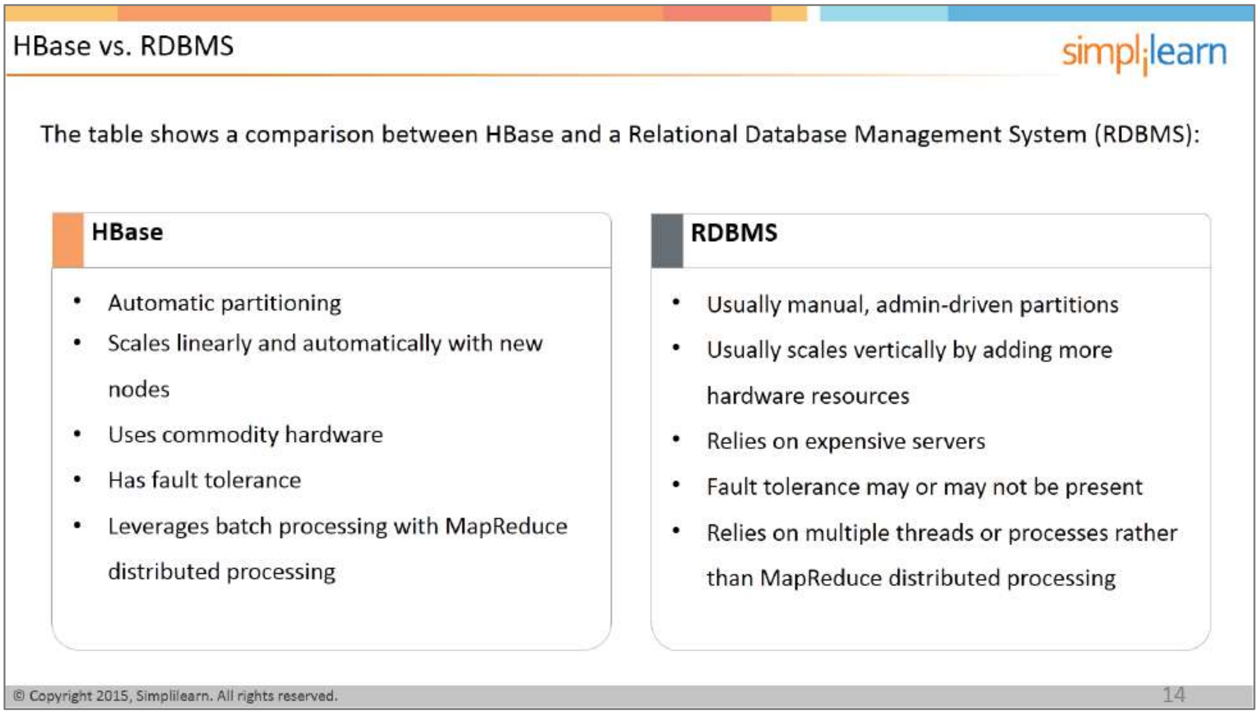
Data Model (contd.)

The Hive Thrift Server enables the integration of Hive with other applications and is responsible for providing a thrift interface to the user. It also maintains connectivity in modules. It provides a Java Database Connectivity or JDBC, or Open Database Connectivity or ODBC server.



When to Use HBase

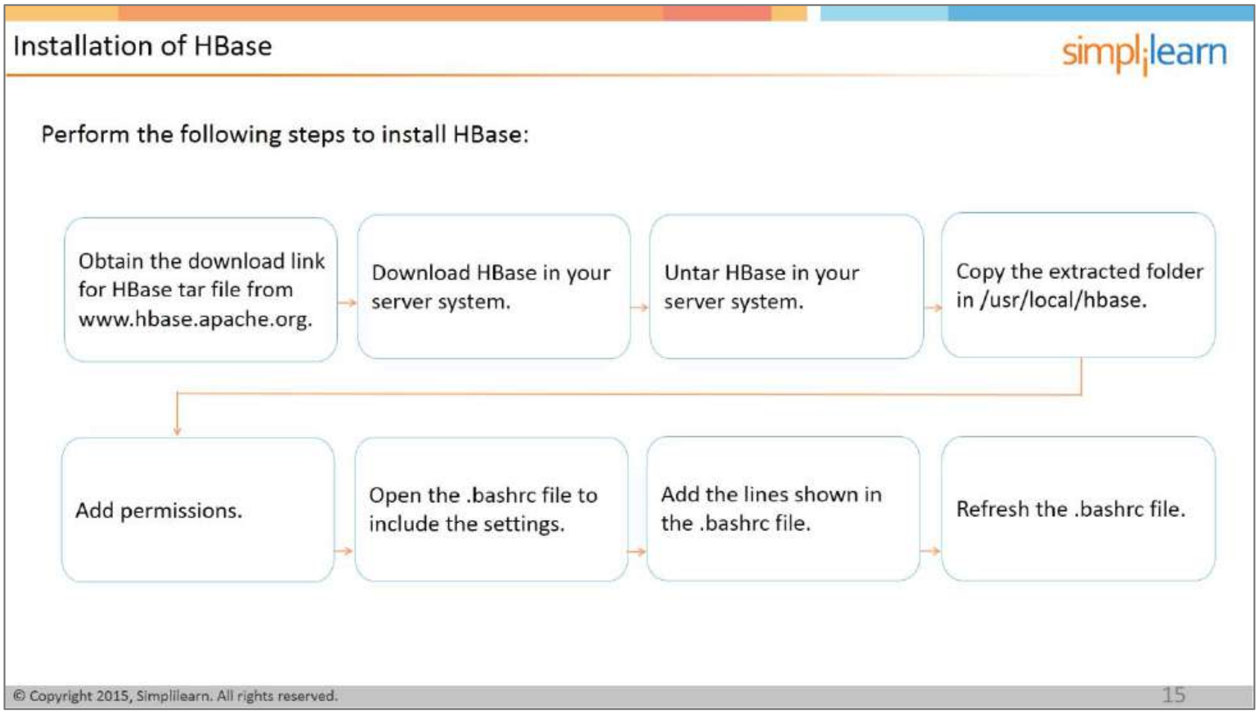
HBase is not suitable for every problem. HBase is used when you have enough data in millions or billions of rows. It can be used when you have sufficient commodity hardware with at least five nodes. It is the developer's responsibility to evaluate HBase carefully for mixed workloads. Developers can use HBase for random selects and range scans by key. They can also utilize HBase in variable schema.



HBase vs. RDBMS

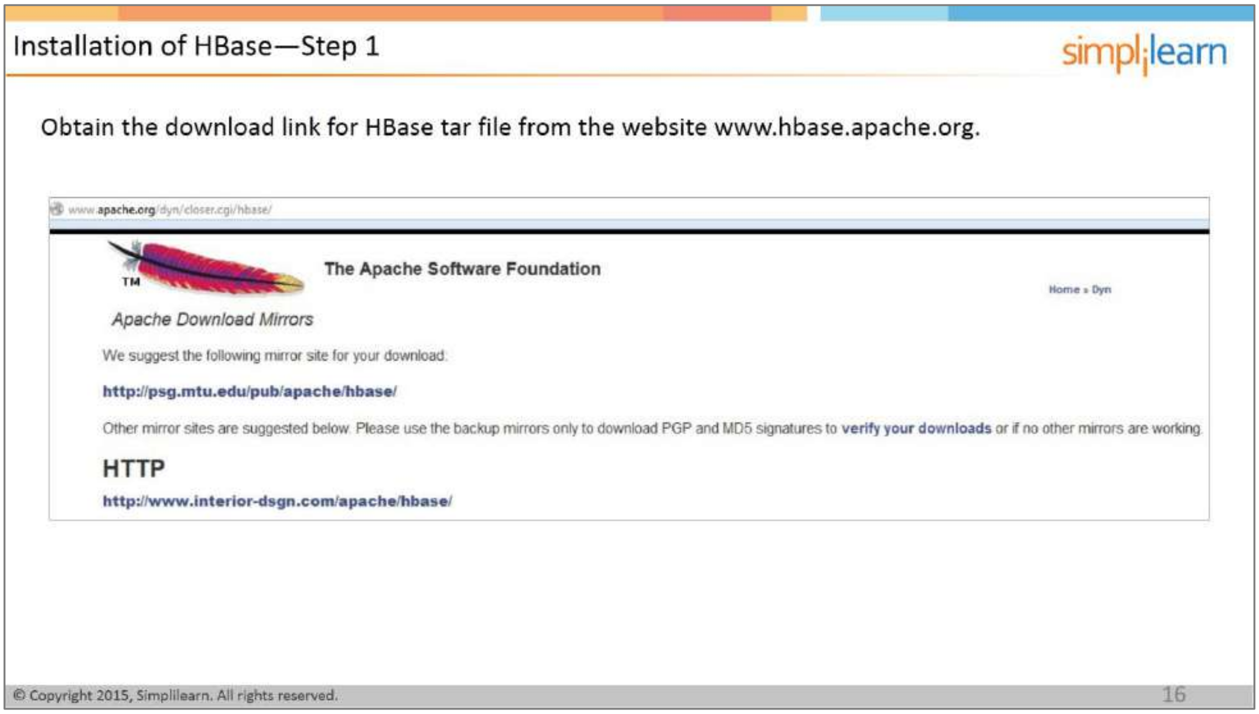
HBase provides advantages in comparison to the Relational Database Management System or RDMS. HBase allows automatic partitioning as compared to manual partitioning in RDBMS. HBase can scale linearly and automatically with new nodes. However, RDBMS primarily scales vertically by adding more hardware resources.

Furthermore, as part of the Hadoop ecosystem, HBase uses commodity hardware while RDBMS relies on expensive servers. HBase has mechanisms for fault tolerance that RDBMS may or may not have. HBase leverages batch processing with MapReduce distributed processing. RDBMS relies on multiple threads or processes rather than MapReduce distributed processing.



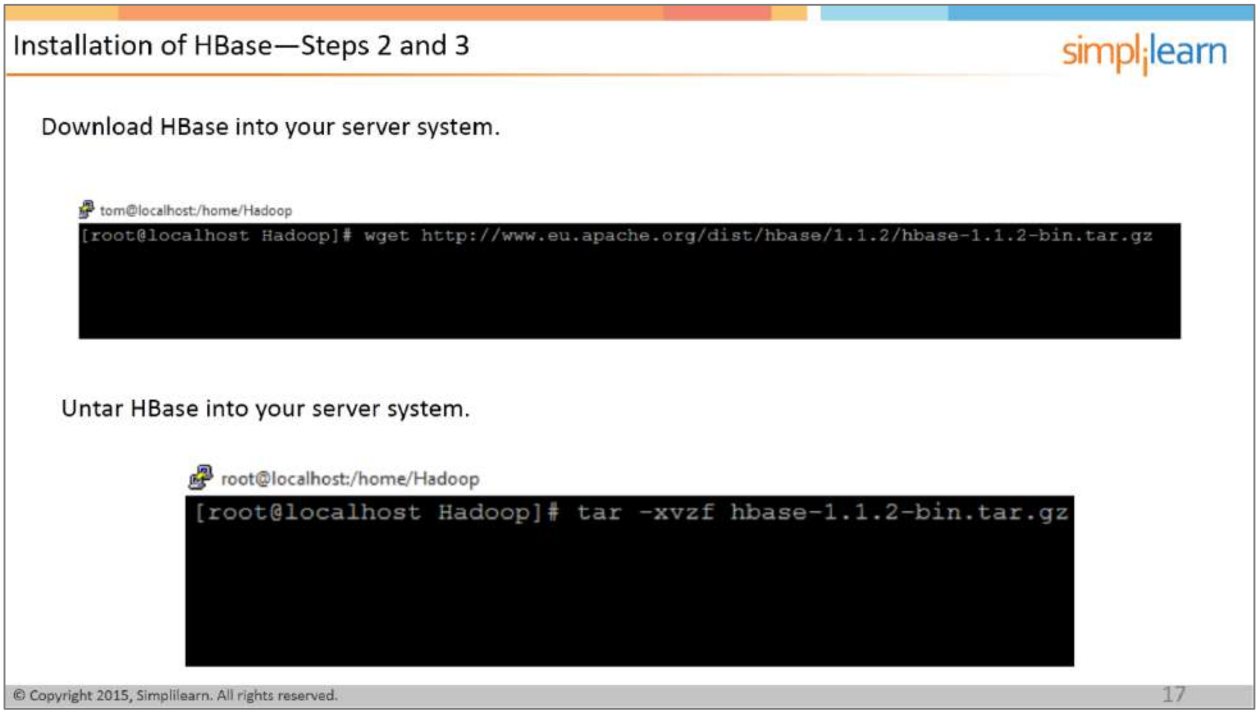
Installation of HBase

The following steps must be performed for the installation of HBase. Obtain the download link for the HBase tar file from the URL mentioned. Download and untar HBase in your server system. Copy the extracted folder in the location mentioned on the image and add permissions. Open the .bashrc file to include the settings. Add the lines shown in the .bashrc file and refresh it.



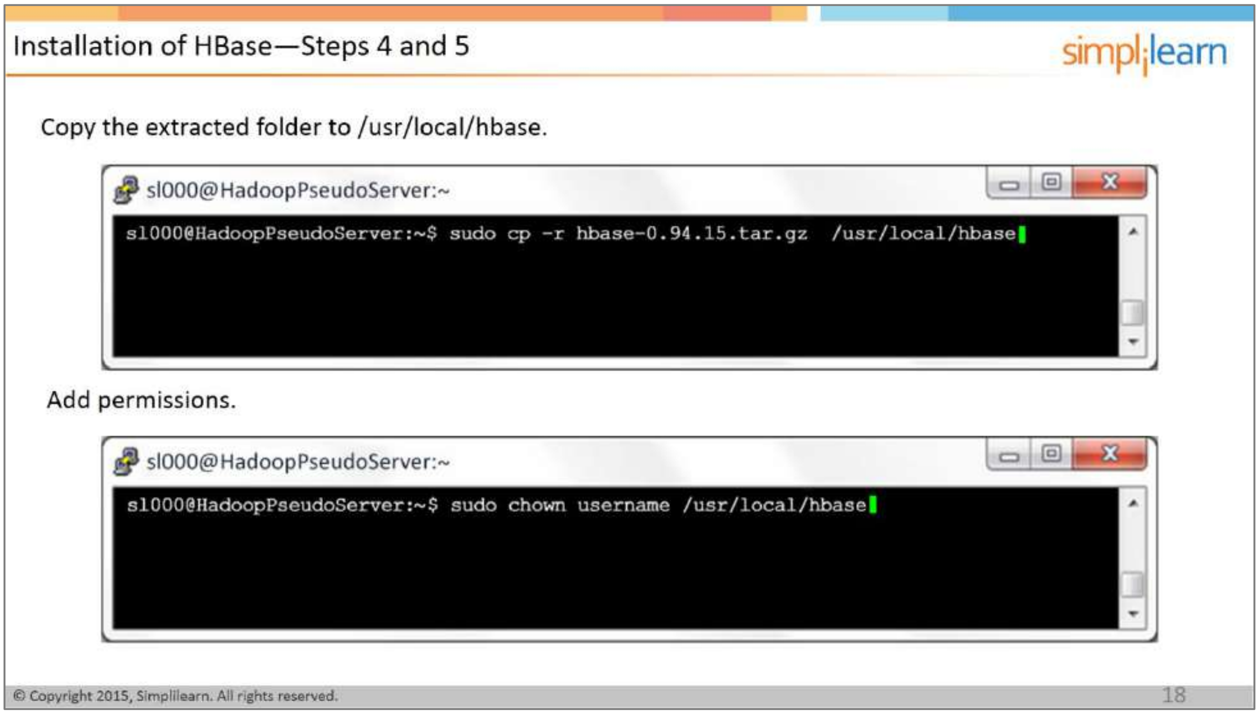
Installation of HBase—Step 1

You can obtain the download link for HBase from the website mentioned on the image.



Installation of HBase—Steps 2 and 3

Once you acquire the download link, you can use the wget command to download the related files of HBase in Ubuntu as shown on the image. Once the files are downloaded, untar the file using the command displayed on the image.

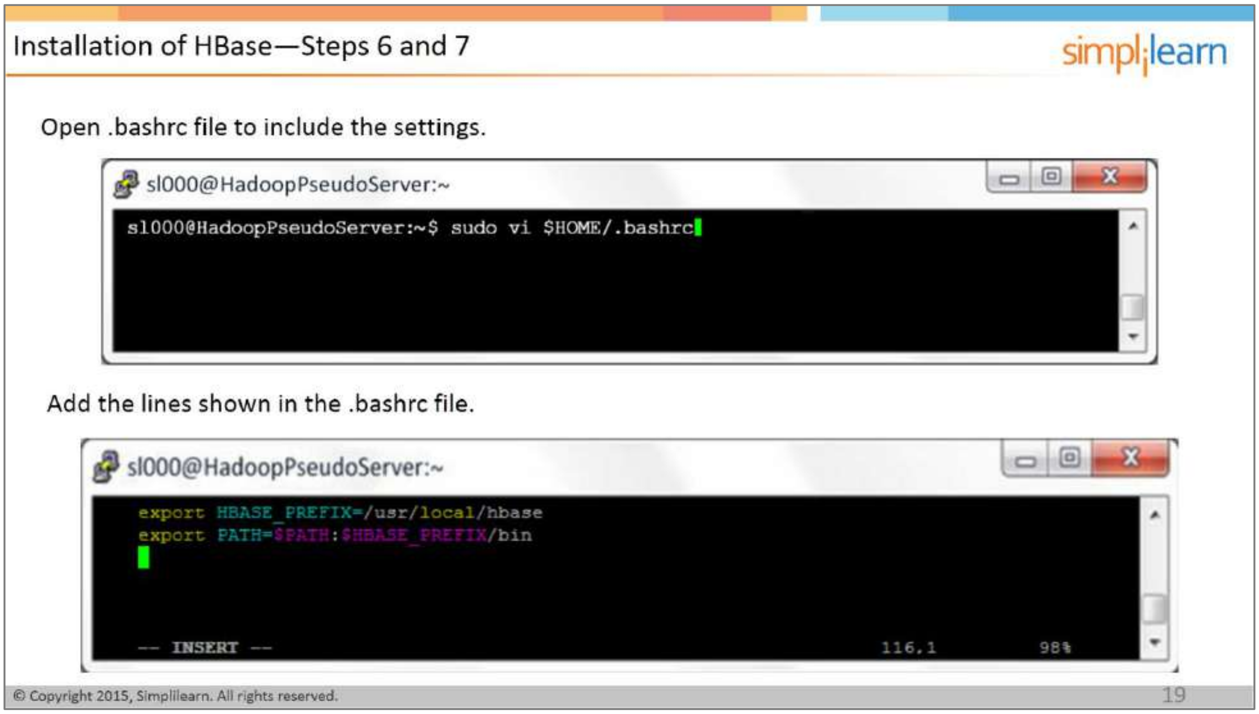


Installation of HBase—Steps 4 and 5

Partitions are similar to dense indexes on columns.

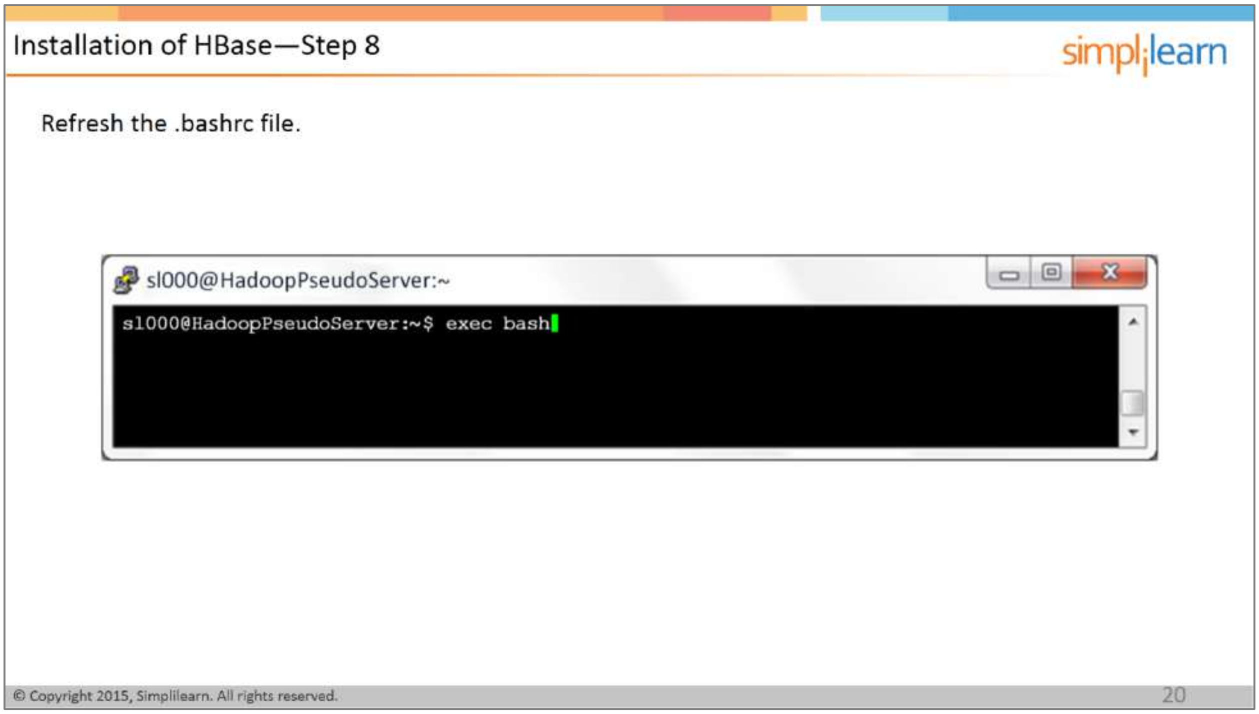
Following are the features of partitions:

- They contain nested sub-directories in HDFS for each combination of partition column values.
- They allow users to retrieve rows efficiently.



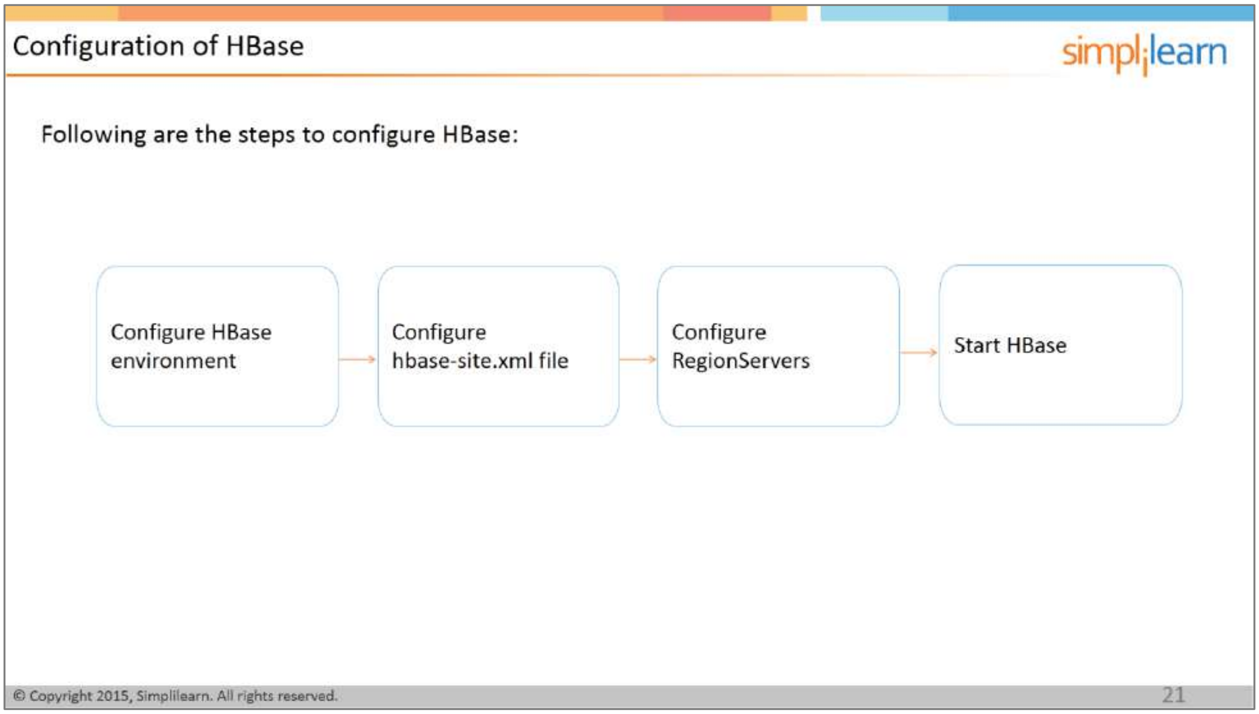
Installation of HBase—Steps 6 and 7

Open the .bashrc file and include the settings by altering the .bashrc file and adding two lines as shown on the image.



Installation of HBase—Step 8

Once the data is saved, refresh the .bashrc file using the command shown on the image.

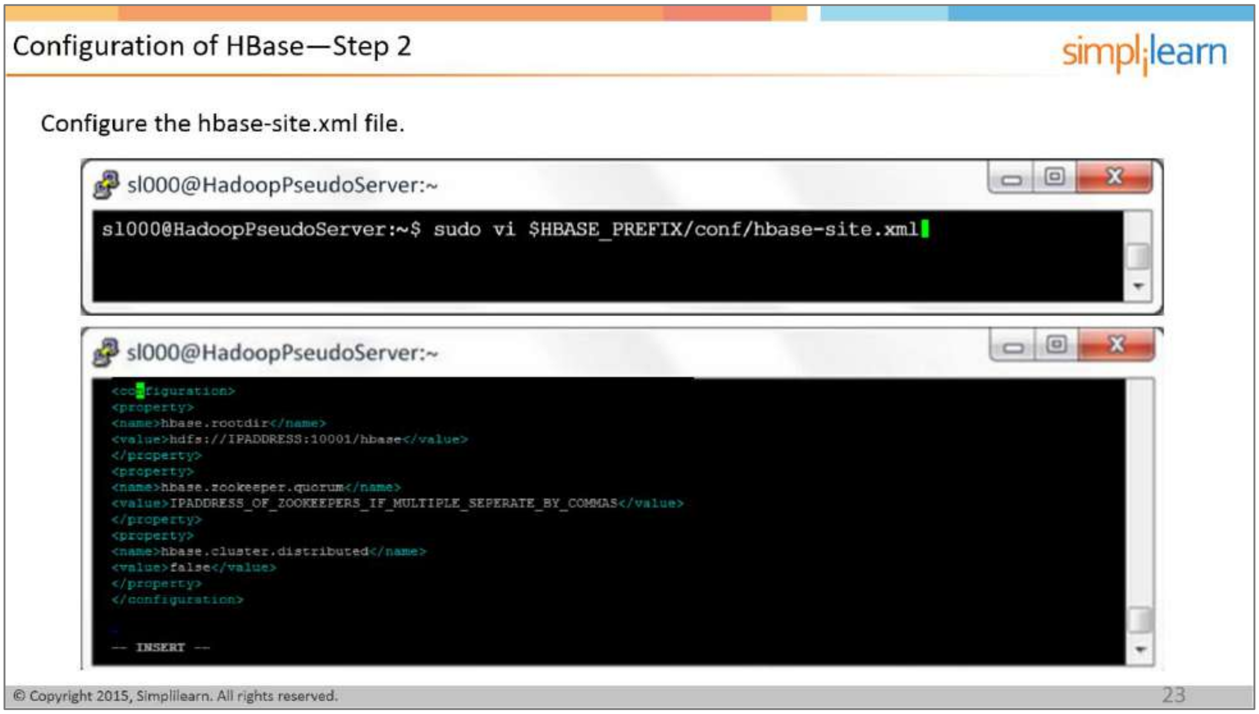


Configuration of HBase

Perform the steps as shown to configure HBase.

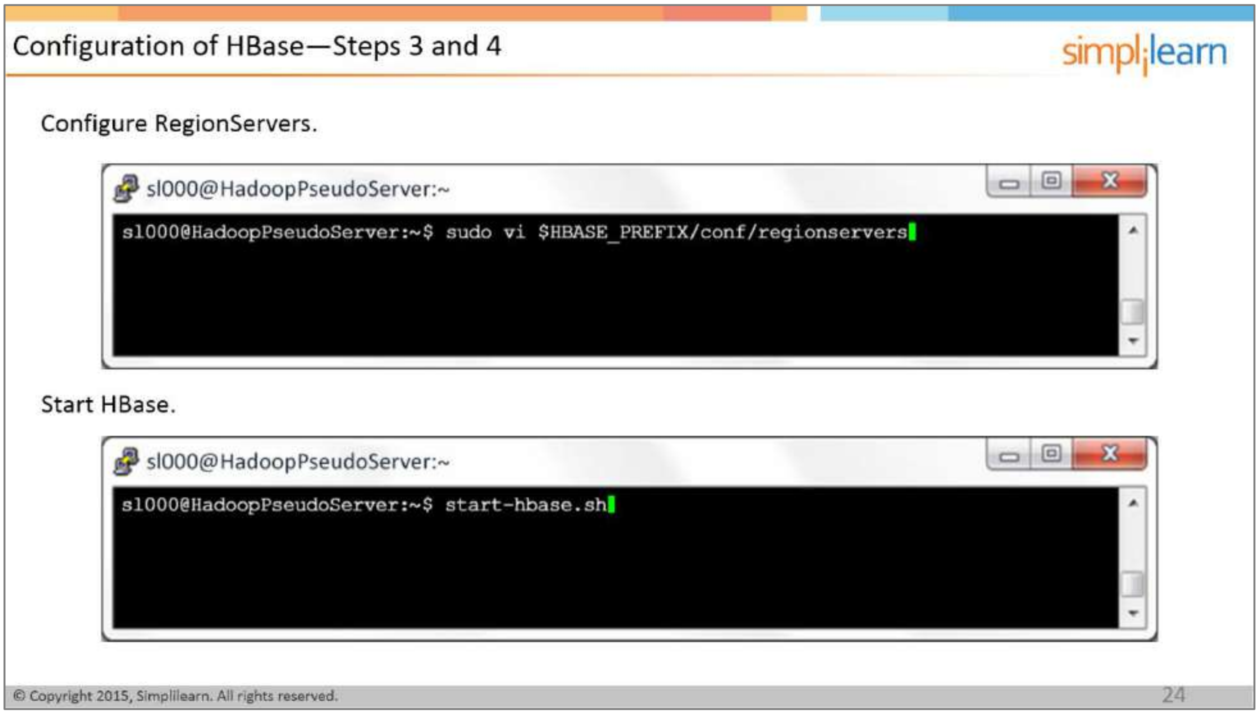
Configuration of HBase—Step 1

Configure the HBase environment and set the JAVA path to ensure that HBase has access to Java Virtual Machine or JVM as shown.



Configuration of HBase—Step 2

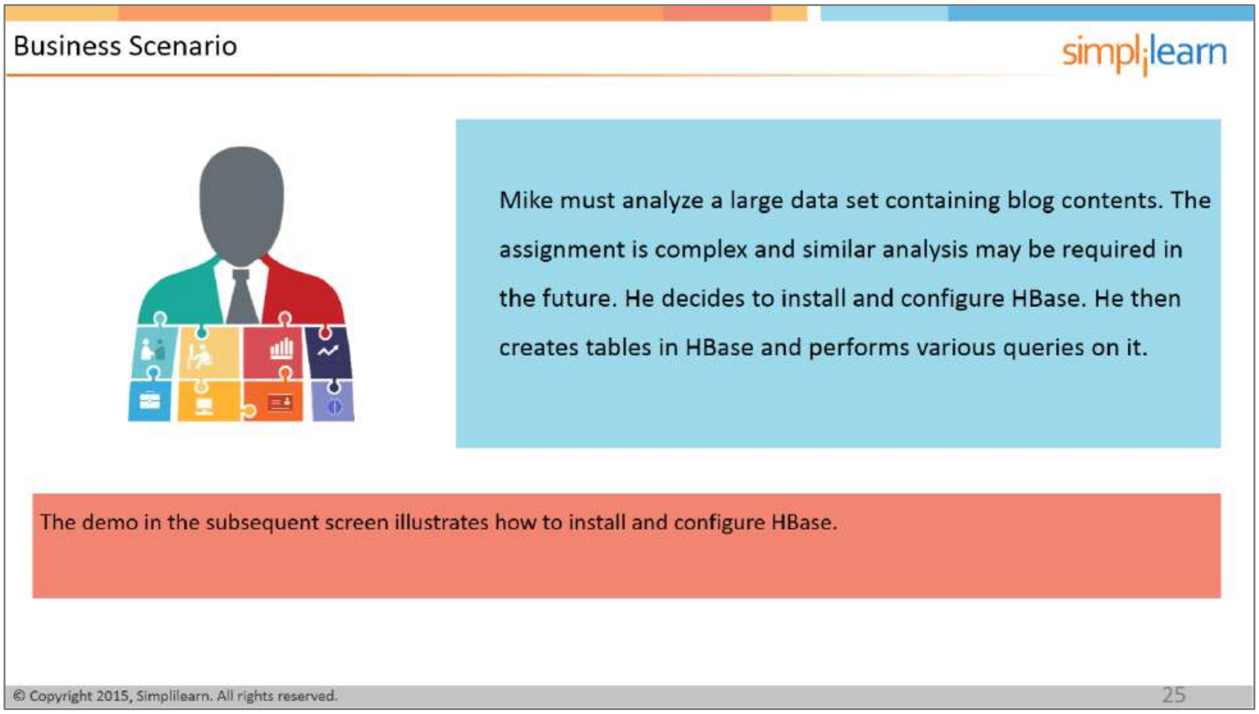
Open the xml file, and enter the code shown. This will set the root directory, timer, and distributed file system interaction.



Configuration of HBase—Steps 3 and 4

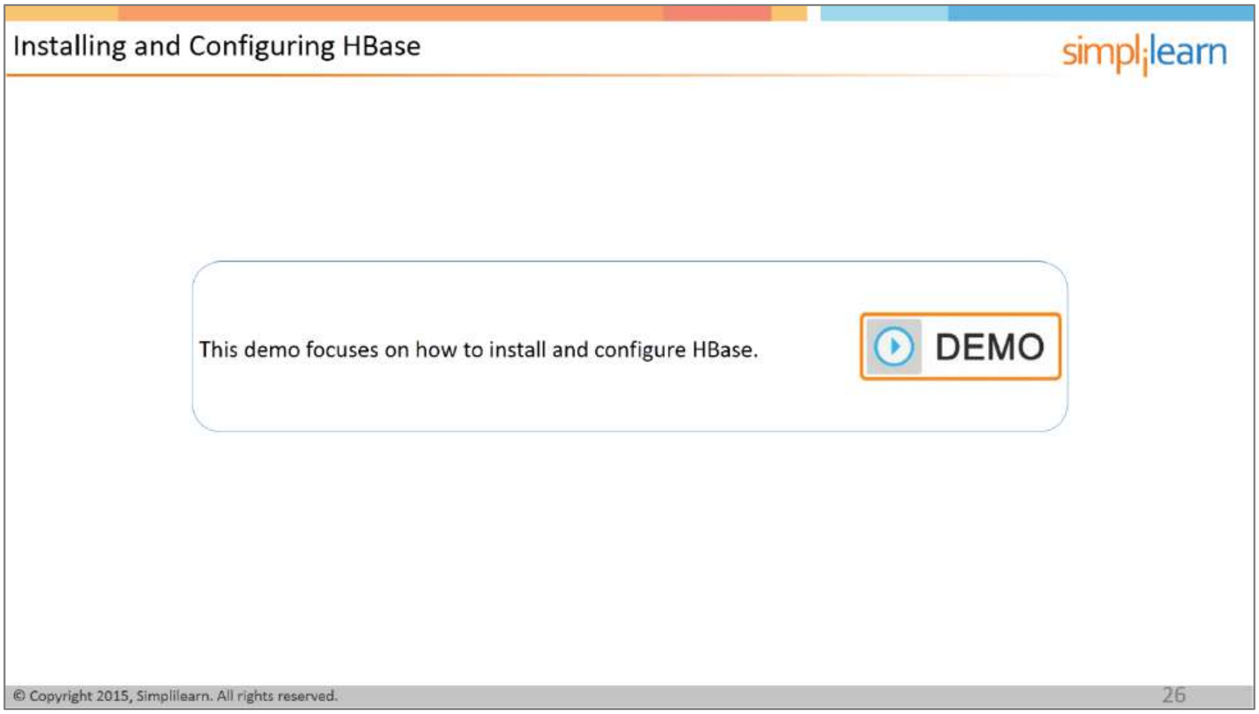
Open RegionServer files, and add the IP address of the system, which will act as RegionServers.

Use the command shown to start the HBase service that can be verified by typing the JPs Command.



Business Scenario

Mike must analyze a large data set containing blog contents. The assignment is complex and similar analysis may be required in the future. He decides to install and configure HBase. He then creates tables in HBase and performs various queries on it.

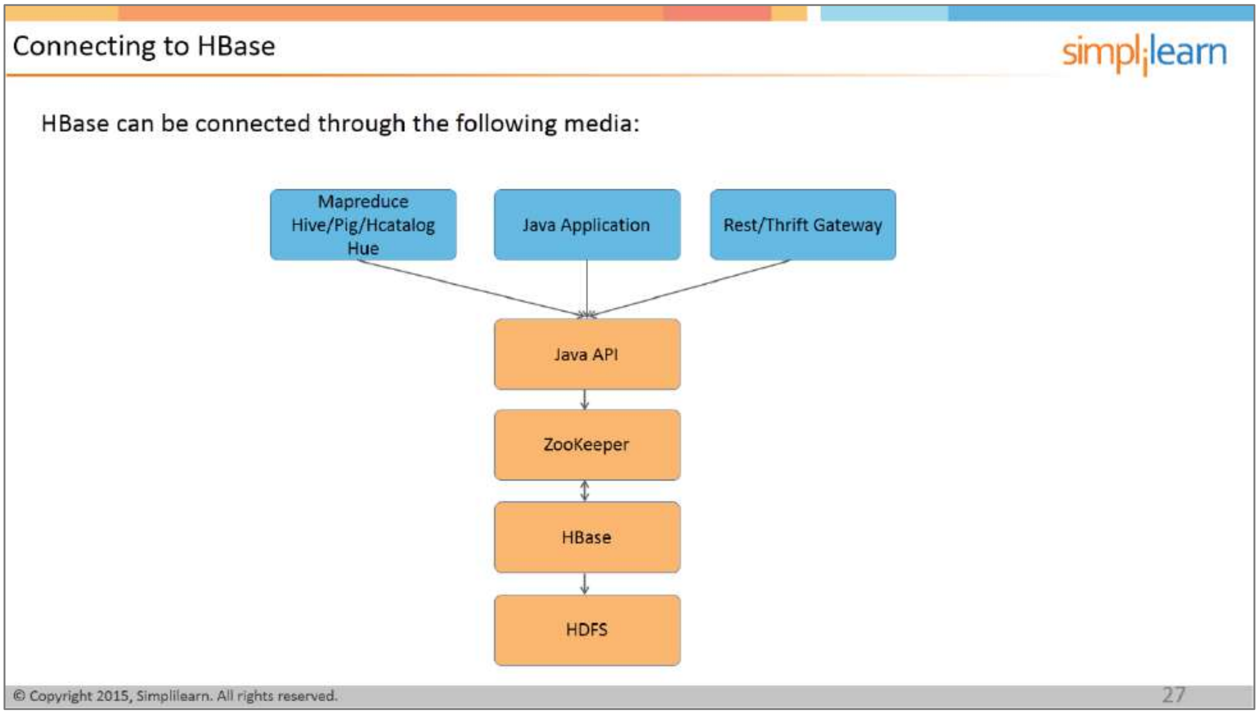


Install and configure HBase

This demo focuses on how to install and configure HBase.

1. Click nearest mirror site.
2. Click ‘Stable’ under Parent Directory.
3. Right click HBase binary file’s path and copy the link to the clipboard.
4. Navigate to the Ubuntu Server shell screen. Type wget command and paste the link.
5. Press Enter to start the downloading process.
6. Verify the download.
7. You will notice the file by the name ‘hbase’ followed by version number.
8. unpack the file once the download is complete.
9. Verify the unpacked directory.
10. You will notice a directory by the name ‘hbase’ followed by version number.
11. Create a soft link for the HBase directory.
12. Set environment variables for HBase in hbase-env.sh file.
13. Enter Java Home path and close the file after saving them.
14. Edit the hbase-site.xml file.
15. Enter HBase and Zookeeper configuration.
16. Ensure this configuration is in pseudo-distributed mode of HBase.
17. Among other configurations, enter root directory of HBase and close the file after saving them.

18. Open .bashrc file for editing.
19. Enter HBase _HOME as directory of HBase. Add HBase bin directory in path. Close the file after saving them.
20. Implement .bashrc changes using source command.
21. Start HBase daemons now.
22. Start the HBase.
23. Launch HBase command line interface.
24. You can use the command 'list' to see various tables.



Connecting to HBase

You can connect to HBase using any of the following media:

- HBase offers a Java Application Programming Interface or API which can be used to conduct usual operations such as get, scan, put, and delete.
- Non-Java clients can connect to HBase using Thrift or REST services.
- HBase also offers a convenient shell built in JRuby where the majority of operations, including admin functions, can be performed from the command line.
- HBase can also be accessed via Hive, Pig, HCatalog, or Hue.

HBase Shell Commands

Common commands include, but are not limited to, the following:

Command	Description
create	Create table; pass table name, a dictionary of specifications per column family, and optionally a dictionary of table configuration hbase> create 't1', {NAME => 'f1'}, {NAME => 'f2'}, {NAME => 'f3'} hbase> # The above in shorthand would be the following: hbase> create 't1', 'f1', 'f2', 'f3'
describe	Describe the named table: hbase> describe 't1'
disable	Start disable of named table: hbase> disable 't1'
drop	Drop the named table. Table must first be disabled: hbase> drop 't1'
list	List all tables in HBase. Optional regular expression parameter can be used to filter the output: hbase> list

 28

HBase Shell Commands

Some of the commands that can be used from HBase shell include:

- Create for the creation of a table,
- Describe for describing the named table,
- Disable for disabling the table,
- Drop for dropping the table, and
- List for listing the tables in HBase.

HBase Shell Commands (contd.)

Common commands include, but are not limited to, the following:

Command	Description
count	Count the number of rows in a table. Return value is the number of rows: hbase> count ‘t1’ hbase> count ‘t1’, INTERVAL => 100000
delete	Put a delete cell value at the specified table/row/column and optionally timestamp coordinates: hbase> delete ‘t1’, ‘r1’, ‘c1’, ts1
get	Get row or cell contents; pass table name, row, and optionally a dictionary of column(s), timestamp, timerange and versions. Examples: hbase> get ‘t1’, ‘r1’ hbase> get ‘t1’, ‘r1’, {COLUMN => ‘c1’, TIMERANGE => [ts1, ts2], VERSIONS => 4}
put	Put a cell ‘value’ at specified table/row/column and optionally timestamp coordinates: hbase> put ‘t1’, ‘r1’, ‘c1’, ‘value’, ts1
scan	Scan a table; pass table name and optionally a dictionary of scanner specifications: hbase> scan ‘t1’, {COLUMNS => ['c1', 'c2'], LIMIT => 10, STARTROW => ‘xyz’}

 29

HBase Shell Commands (contd.)

Other commands that can be used from HBase shell are:

- Count for counting the number of rows in a table,
- Delete for deleting a cell value,
- Get for getting the contents of a row or cell,
- Put for putting a cell value, and
- Scan for scanning a table’s values.

Spot the Error

Linda, a Business Analyst had to deal with billions of rows and columns to strategize the business standards. She tried to install and configure HBase in her system. However, she was prompted with an error message while installing HBase in her system. Can you assist Linda by identifying the error in the code?

Spot the incorrect code.

```
# wget http://www.us.apache.org/dist/hbase/stable/hbase-1.1.2-bin.tar.gz
# untar xvzf hbase-1.1.2-bin.tar.gz
# sudo ln -s /home/hadoop/hbase-1.1.2 /usr/local/hbase
# nano /usr/local/hbase/conf/hbase-env.sh
# nano /usr/local/hbase/conf/hbase-site.xml
```

 33

Spot the Error

The incorrect code is:

```
# untar xvzf hbase-1.1.2-bin.tar.gz
```

The correct syntax is:

```
# tar xvzf hbase-1.1.2-bin.tar.gz
```

 34

QUIZ 1

Which of the following category of database does HBase belong to?

a. Column oriented

b. Relational database

c. Object oriented

d. File system

QUIZ 2

Which scalability factor can be achieved with HBase?

a. Horizontal scalability

b. Vertical scalability

c. Lateral scalability

d. None of these

QUIZ 3

Which of the following settings is to be checked to ensure multiple sites in distributed mode?

a. Java

b. RegionServer

c. Hbase-env.sh

d. Hbase-site.xml

 40

QUIZ 4

Which of the following settings is to be checked to ensure all Java dependencies are enabled in HBase?

a. Java

b. RegionServer

c. Hbase-env.sh

d. Hbase-site.xml

 42

QUIZ 5

What is the main responsibility of HBase?

a. To perform query optimization

b. To perform read/write access to Big Data

c. To store the Big Data

d. To break the Big Data into chunks

QUIZ 6

Why does HBase use ZooKeeper?

a. To perform region assignment

b. To perform error recovery

c. To perform error detection

d. To speed up the process

simplilearn

QUIZ 7

Which of the following files in HBase is responsible for setting the IP address of the HBase server?

a. Hbase-master.xml

b. Hbase-site.xml

c. Hbase-slave.xml

d. RegionServer

 48

simplilearn

QUIZ 8

Which of the following files in HBase is responsible for maintaining the RegionServer information?

a. Hbase-master.xml

b. Hbase-site.xml

c. Hbase-slave.xml

d. RegionServer

 50

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which of the following category of database does HBase belong to?	a. HBase is a column oriented database.
2	Which scalability factor can be achieved with HBase?	a. Horizontal scalability factor can be achieved with HBase.
3	Which of the following settings is to be checked to ensure multiple sites in distributed mode?	d. Hbase-site.xml is used to set multiple sites in the distributed mode of HBase.
4	Which of the following settings is to be checked to ensure all Java dependencies are enabled in HBase?	c. The Hbase-env.sh file is used to ensure all Java dependencies are enabled in HBase.
5	What is the main responsibility of HBase?	b. The main responsibility of HBase is to perform read/write access to Big Data.

ANSWERS:

S.No.	Question	Answer & Explanation
6	Why does HBase use ZooKeeper?	a. HBase uses ZooKeeper to perform region assignment in distributed mode.
7	Which of the following files in HBase is responsible for setting the IP address of the HBase server?	b. Hbase-site.xml is responsible for setting the IP address of the HBase server.
8	Which of the following files in HBase is responsible for maintaining the RegionServer information?	d. The RegionServer file is responsible for maintaining the RegionServer information.

Case Study

Scenario | Analysis | Solution

XYZ is a social networking site that connects people. It support millions of customers and maintains petabytes of information. Their databases and data marts are running out of space. They want a real-time read and update of millions of records every minute. They keep upgrading their machines, but the processing power cannot keep up with customer needs. Customers are facing significant lags in data access. They heard that NoSQL solutions can help them fix their problems, and HBase is a popular database that runs on commodity hardware.

 53

Case Study

Scenario | Analysis | Solution

The IT team does research on HBase, and finds that it is a popular Big Data real-time processing database. HBase uses the same storage system as Hadoop, but provides faster processing.

Advantages of using HBase are:

1. It can process Big Data.
2. It provides random read/write access to data.
3. It provides real-time data processing.
4. It is highly scalable and fault-tolerant.
5. It runs on commodity hardware, and significantly reduces cost.
6. It uses an efficient columnar storage, and keeps sparse customer data.

 54

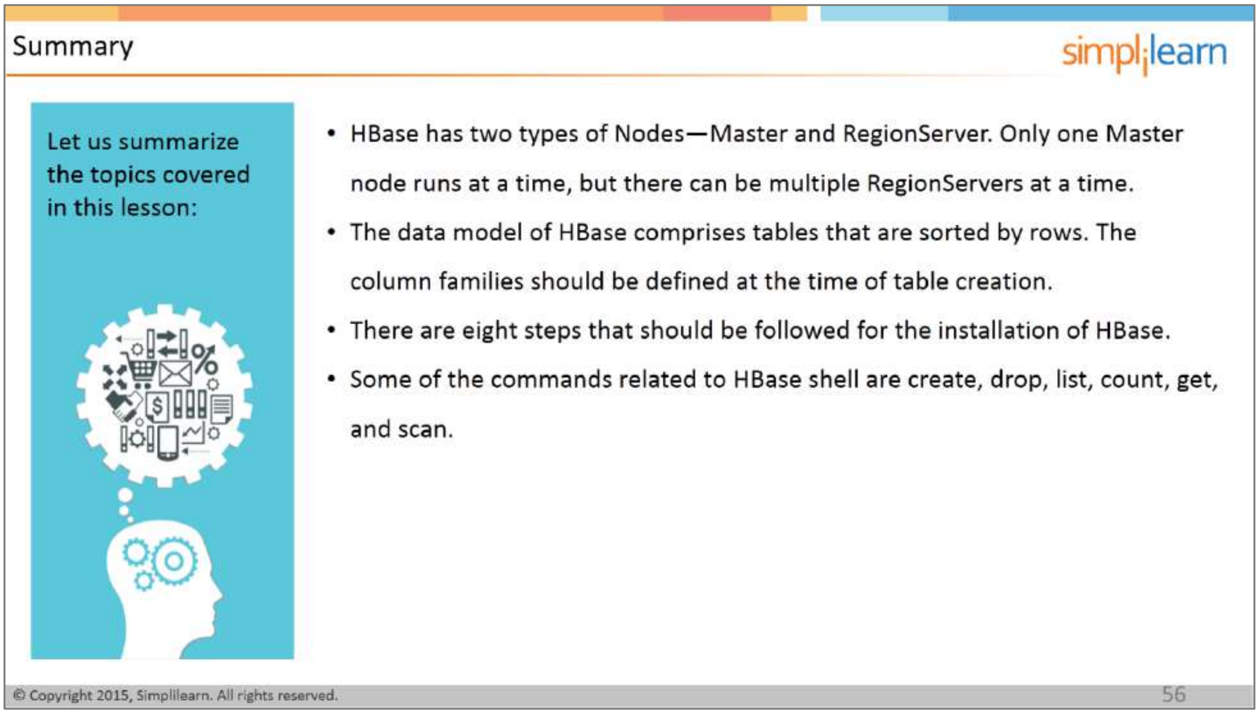
Case Study

Scenario | Analysis | **Solution**

Perform the following steps to explore HBase:

1. Create a table for customers in HBase with few column families.
2. Add versioning to the table.
3. Insert few rows into the table.
4. Check data and its multiple versions.
5. Check update/delete of data.
6. Check filtering of data.

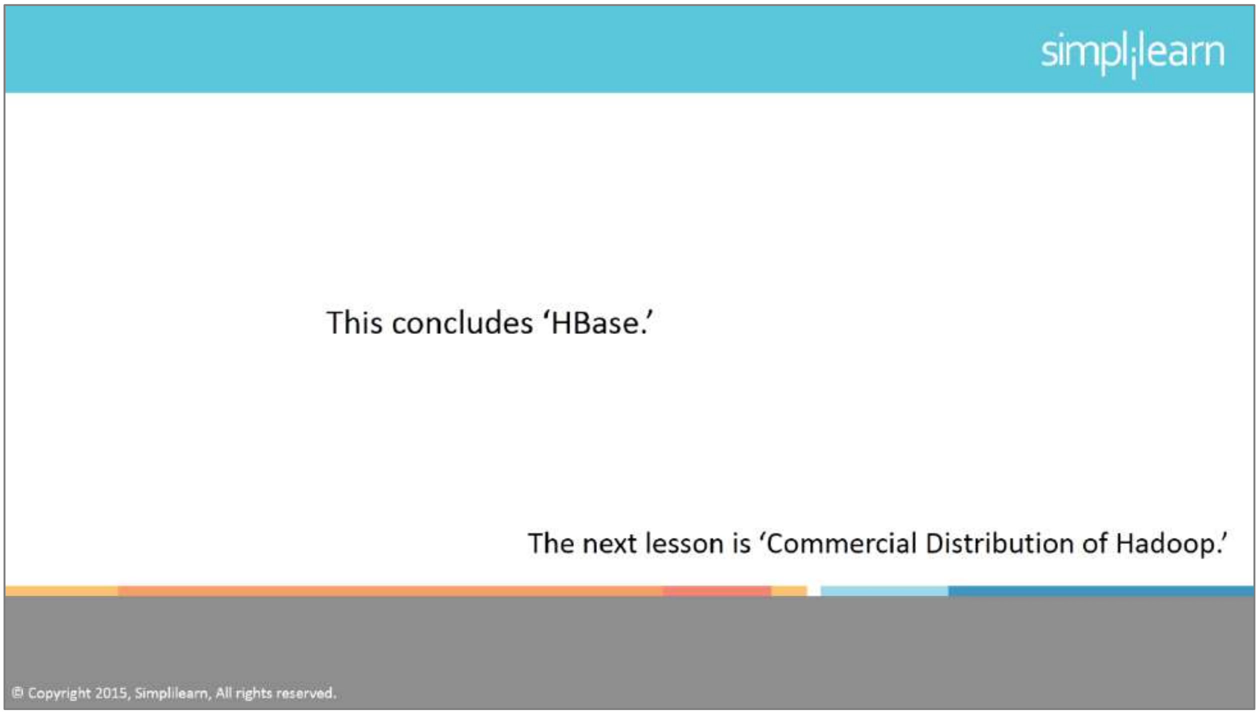
55



Summary:

Let us summarize the topics covered in this lesson:

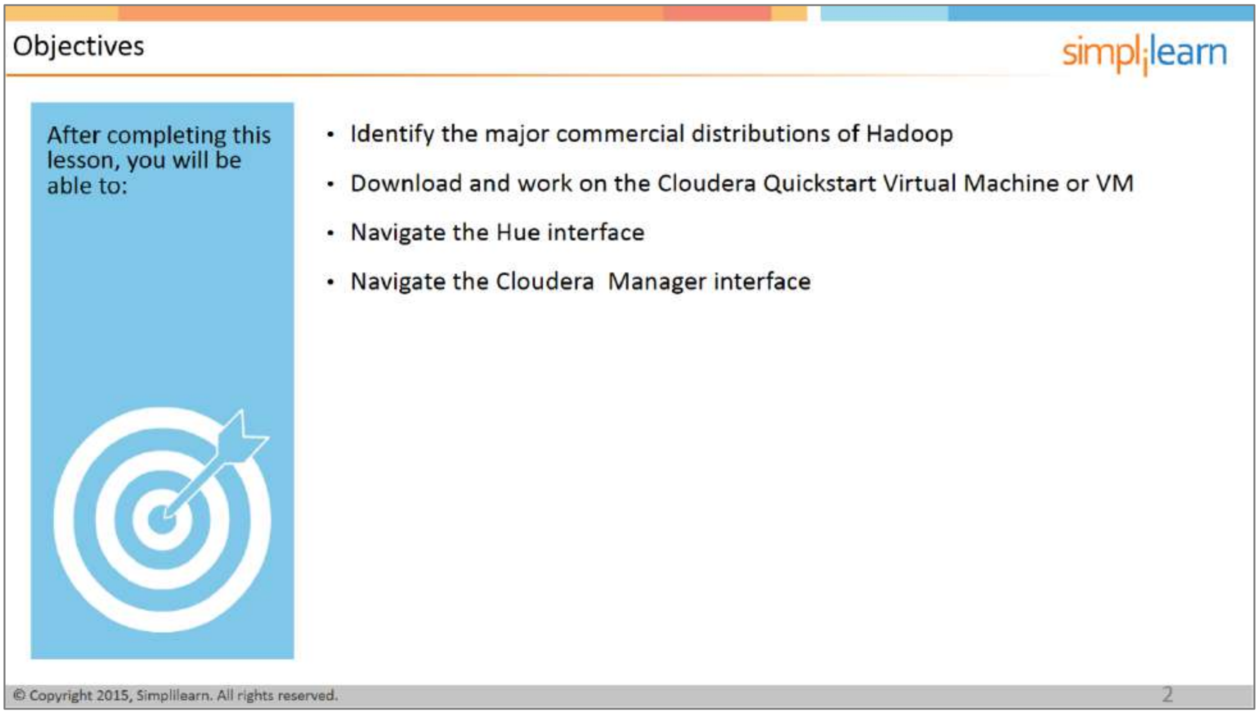
- HBase has two types of Nodes—Master and RegionServer. Only one Master node runs at a time, but there can be multiple RegionServers at a time.
- The data model of HBase comprises tables that are sorted by rows. The column families should be defined at the time of table creation.
- There are eight steps that should be followed for the installation of HBase.
- Some of the commands related to HBase shell are create, drop, list, count, get, and scan



Conclusion

This concludes 'HBase.' In the next lesson we will focus on 'Commercial Distribution of Hadoop.'

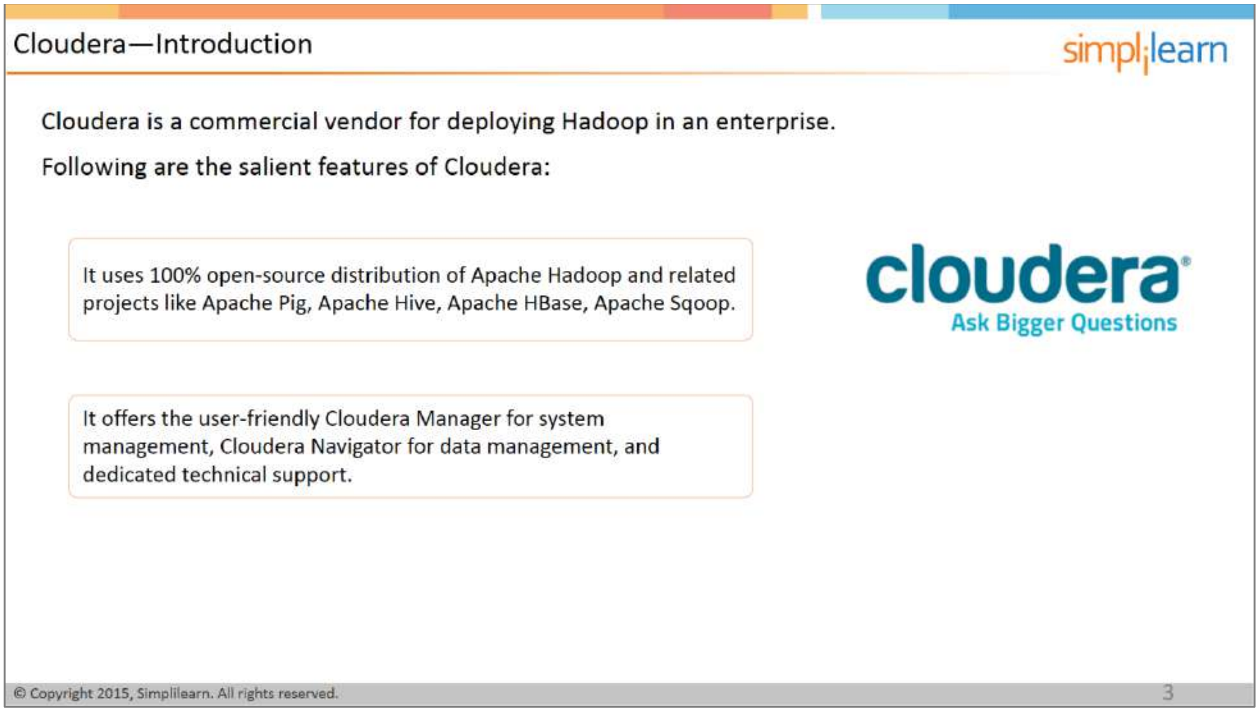
Lesson 9—Commercial Distribution of Hadoop



Objectives

By the end of this lesson, you will be able to:

- Identify the major commercial distributions of Hadoop
- Download and work on the Cloudera Quickstart Virtual Machine or VM
- Navigate the Hue interface
- Navigate the Cloudera Manager interface



Cloudera—Introduction

Cloudera is a commercial tool used to employ Hadoop in an enterprise setup. Cloudera uses 100percent open-source distribution of Apache Hadoop and related projects such as Apache Pig, Apache Hive, Apache HBase, and Apache Sqoop. Cloudera offers the user-friendly Cloudera Manager for system management, Cloudera Navigator for data management, and dedicated technical support.

Cloudera CDH

Cloudera's distribution is known as CDH (Cloudera Distribution Including Apache Hadoop), which delivers the core elements of Hadoop.

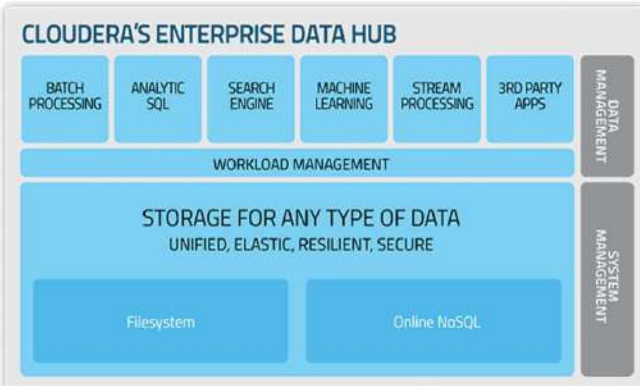
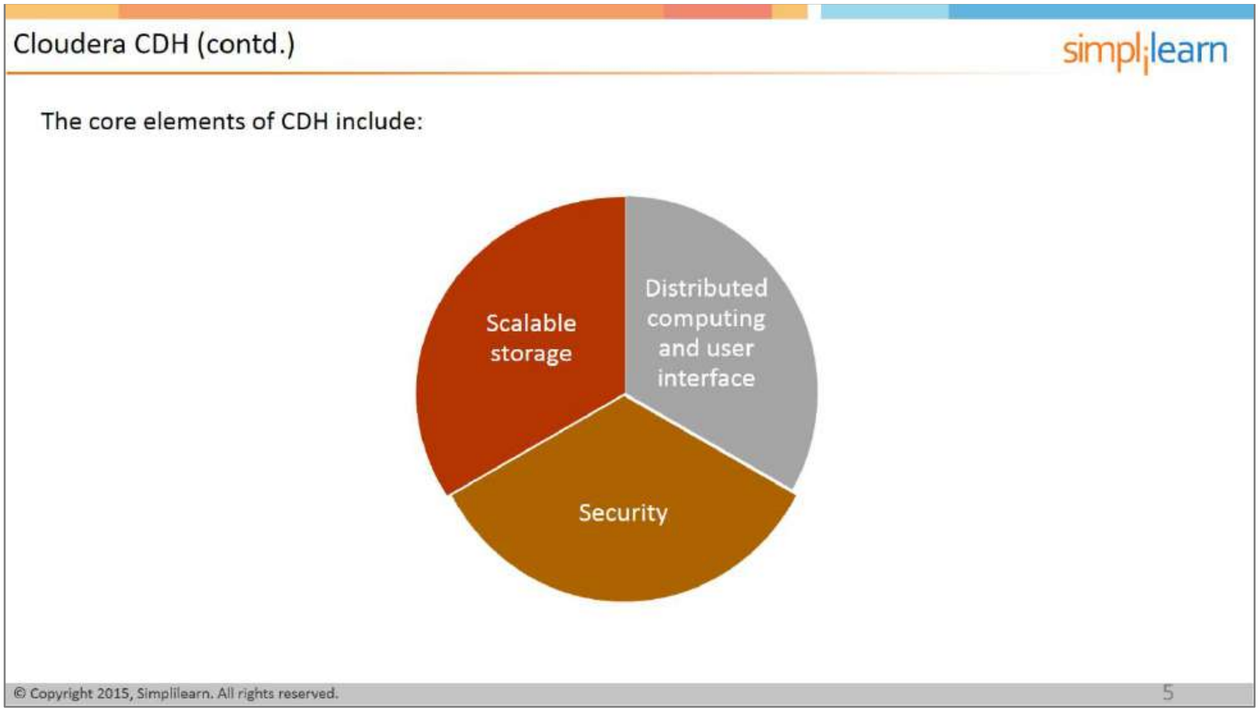


Image source: cloudera.com

Cloudera CDH

Just like Linux, Cloudera and many other vendors offer Hadoop as a commercial distribution. Cloudera's distribution is known as CDH, or Cloudera Distribution including Apache Hadoop.

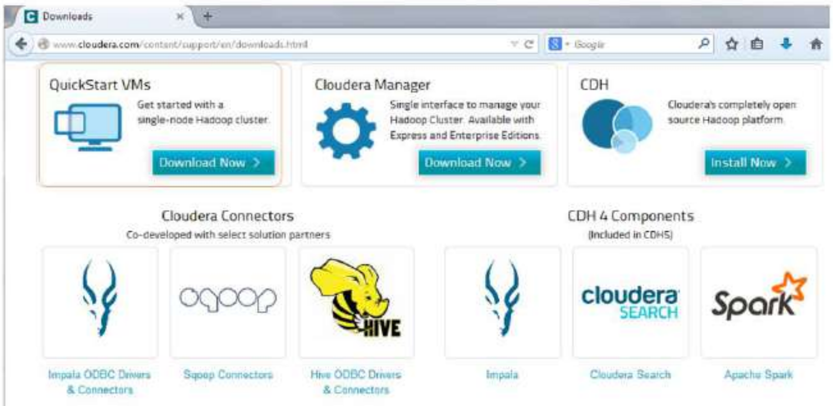


Cloudera CDH

Just like Linux, Cloudera and many other vendors offer Hadoop as a commercial distribution. Cloudera's distribution is known as CDH, or Cloudera Distribution including Apache Hadoop.

Downloading the Cloudera VM

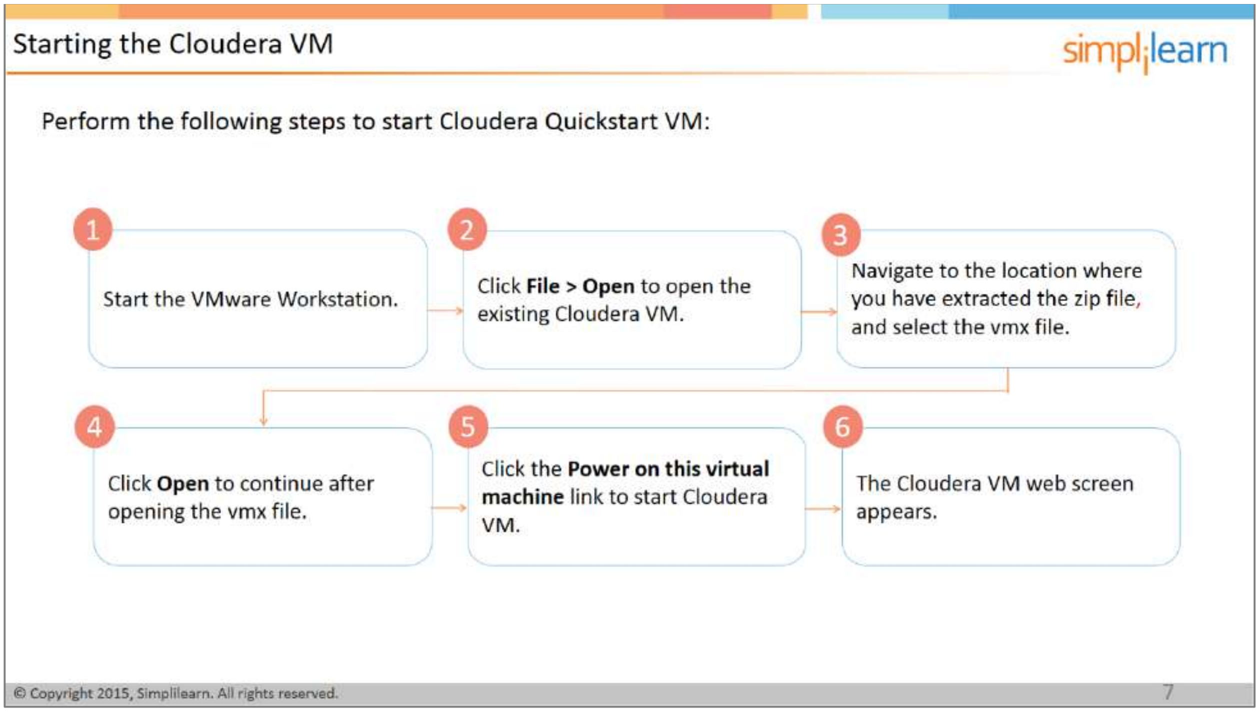
To explore the features of Cloudera, download the Cloudera QuickStart Virtual Machine (VM) from the given link:
http://www.cloudera.com/content/support/en/downloads.html



 6

Downloading the Cloudera VM

To explore the features of Cloudera, download the Cloudera QuickStart VM from the URL given here. You will need VMware Player or Workstation to start and work on the Cloudera VM.

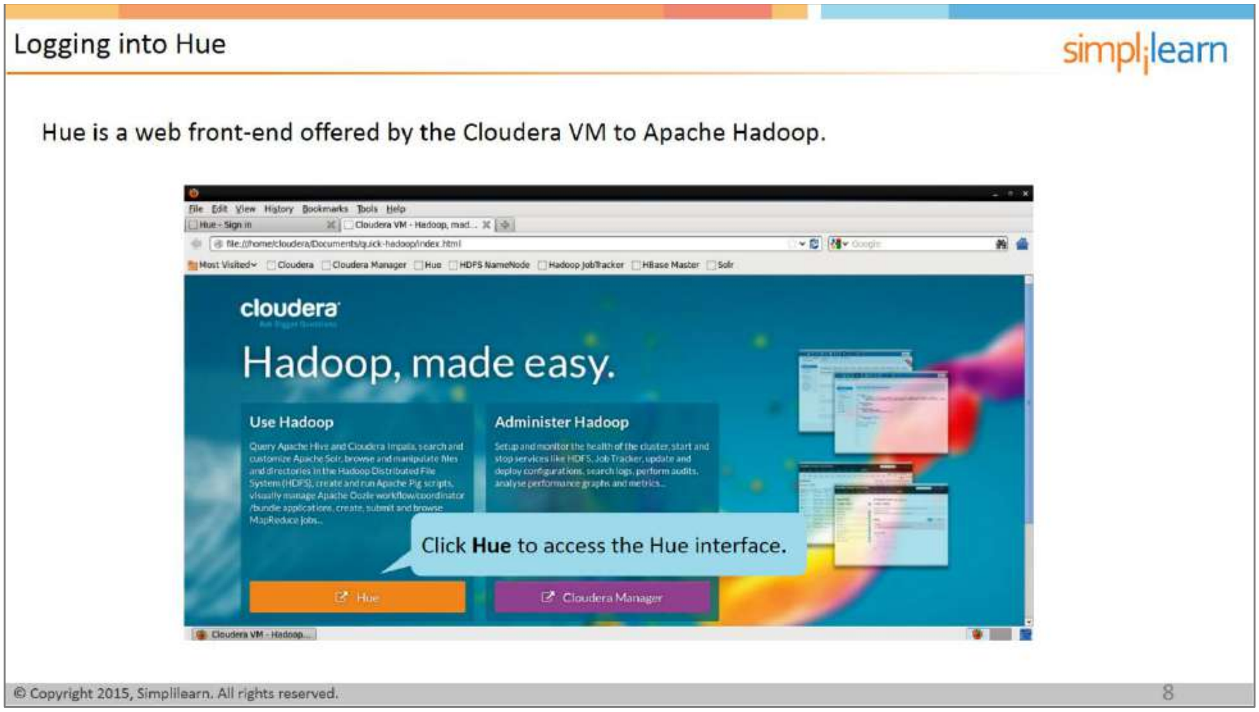


Starting the Cloudera VM

Follow the steps displayed here to start the Cloudera VM.

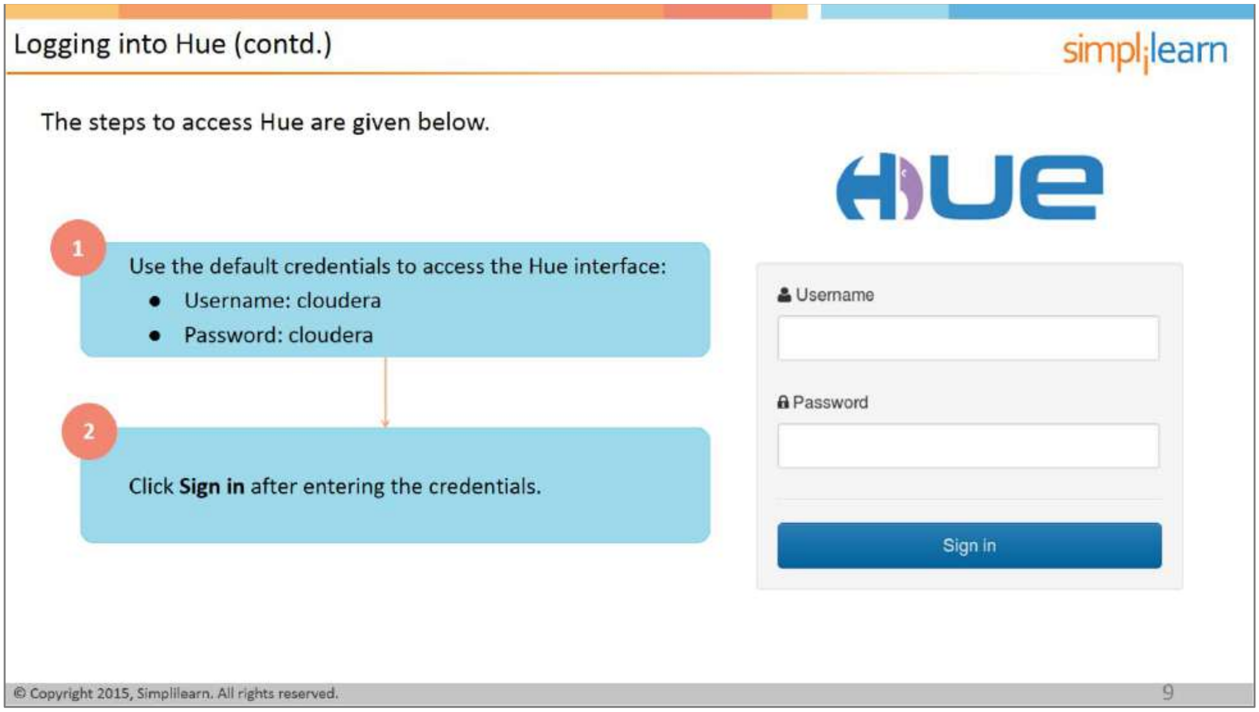
- First, start the VMware Workstation.
- Second, click the **File** menu and select the **Open** submenu to open the existing Cloudera VM.
- Third, navigate to the location where you have extracted the zip file, and select the vmx file.
- Fourth, click **Open** to continue after opening the vmx file.
- Fifth, click the **Power on this virtual machine** link to start the Cloudera VM.
- The Cloudera VM web screen appears.

This is the place from where you can begin working on the Cloudera VM.



Logging into Hue

Hue is a web front-end offered by the Cloudera VM to handle open-source Apache Hadoop. Hue provides an interface to start services such as Hive and Pig. Click the **Hue** button to access the Hue screen.

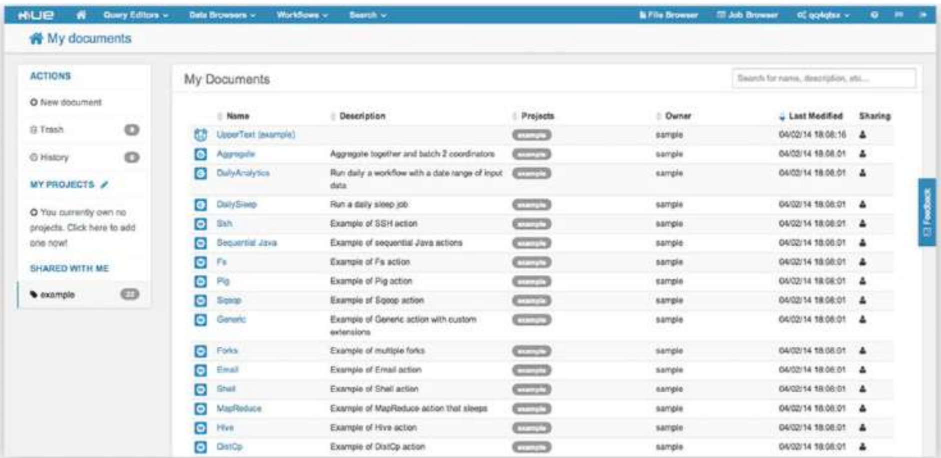


Logging into Hue (contd.)

Once you click the **Hue** button, you will be redirected to the Hue login screen. The default username and password for accessing Hue is ‘cloudera’. Enter the username and password, and sign in to continue.

Logging into Hue (contd.)

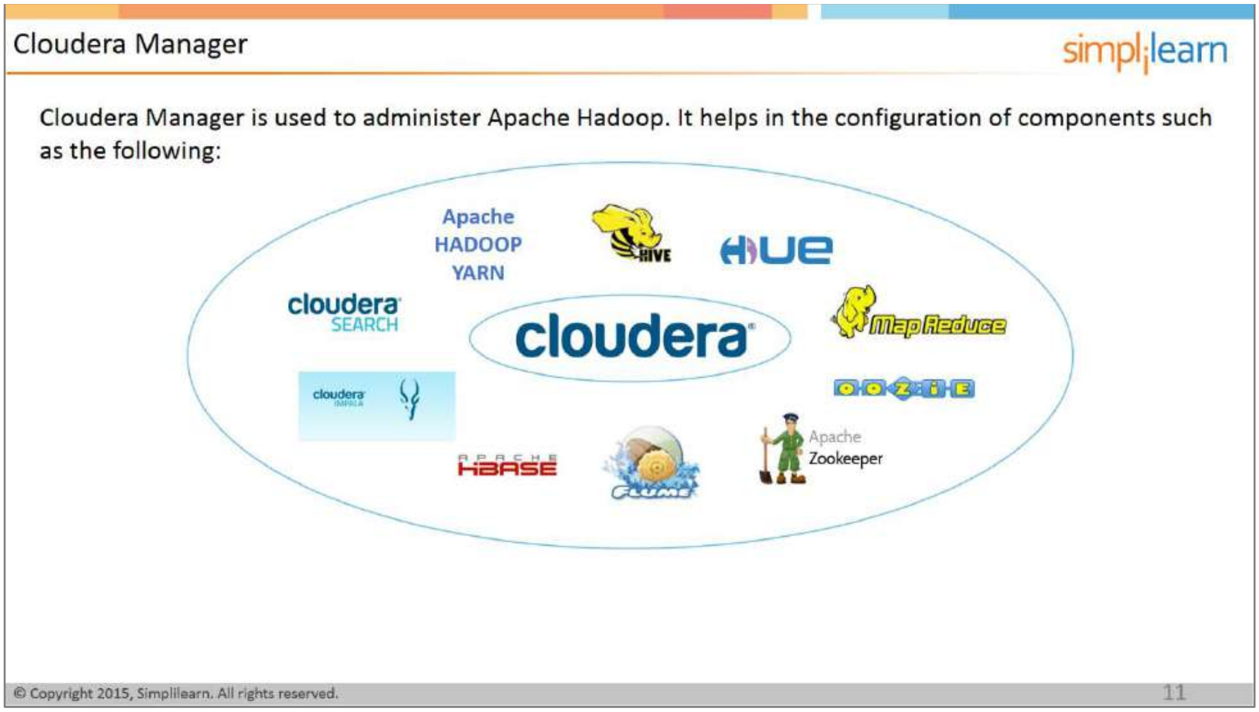
The Hue interface is displayed when you log in by entering your credentials.



 10

Logging into Hue (contd.)

The image shown displays the Hue interface that appears when you log in after entering your default user information. The developer's screen with links to various tools for performing development in Hadoop are displayed.



Cloudera Manager

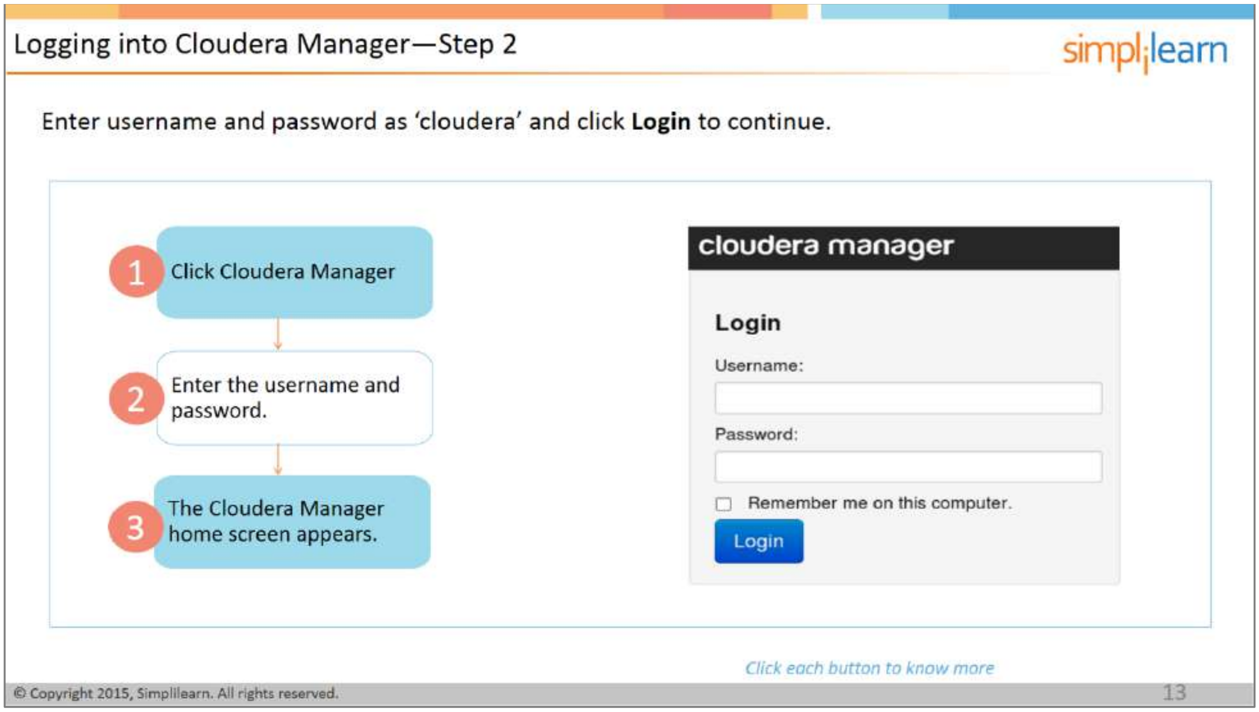
Cloudera Manager is used to administer Apache Hadoop. It is used to configure the following:

- HDFS
- Hive engine
- Hue
- MapReduce
- Oozie
- ZooKeeper
- Flume
- HBase
- Cloudera Impala
- Cloudera Search and
- YARN

Logging into Cloudera Manager

Perform the following steps to log in to Cloudera Manager:

- Click the Cloudera Manager button to access the home screen, and accept any terms and conditions if prompted.
- Enter the username and password as 'cloudera'
- The Cloudera Manager home screen appears.

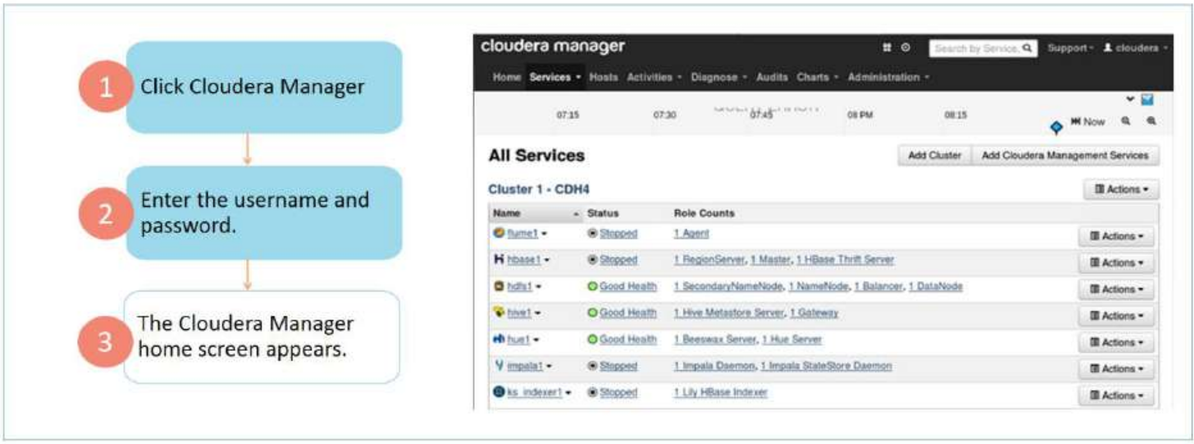


Logging into Cloudera Manager

Enter the username and password as 'cloudera'.

Logging into Cloudera Manager—Step 3

The Cloudera Manager interface is displayed.

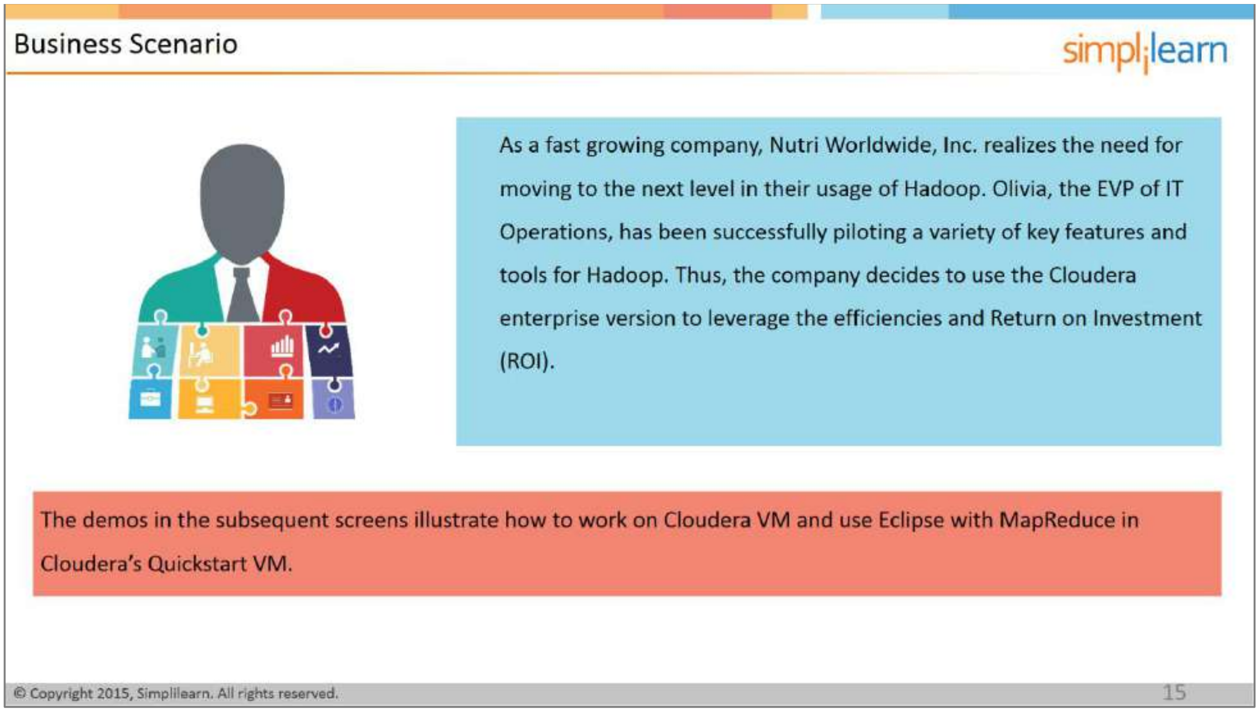


Click each button to know more

 14

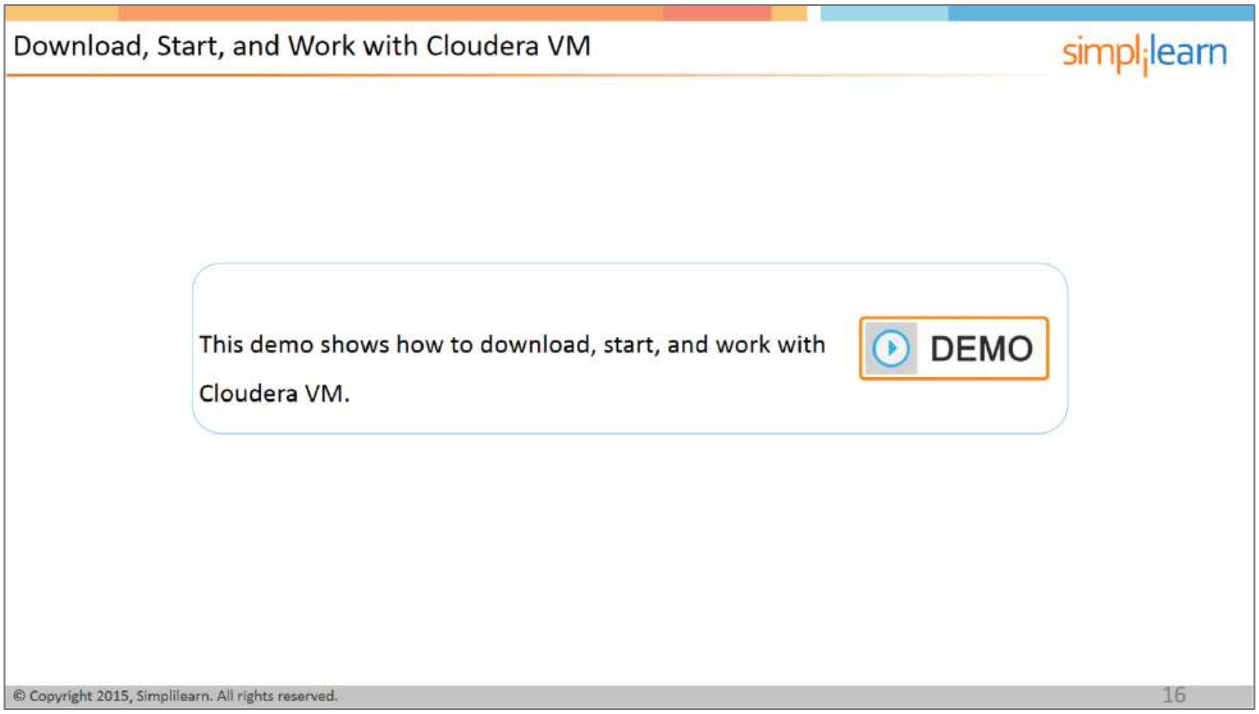
Logging into Cloudera Manager

The Cloudera Manager home screen appears.



Business Scenario

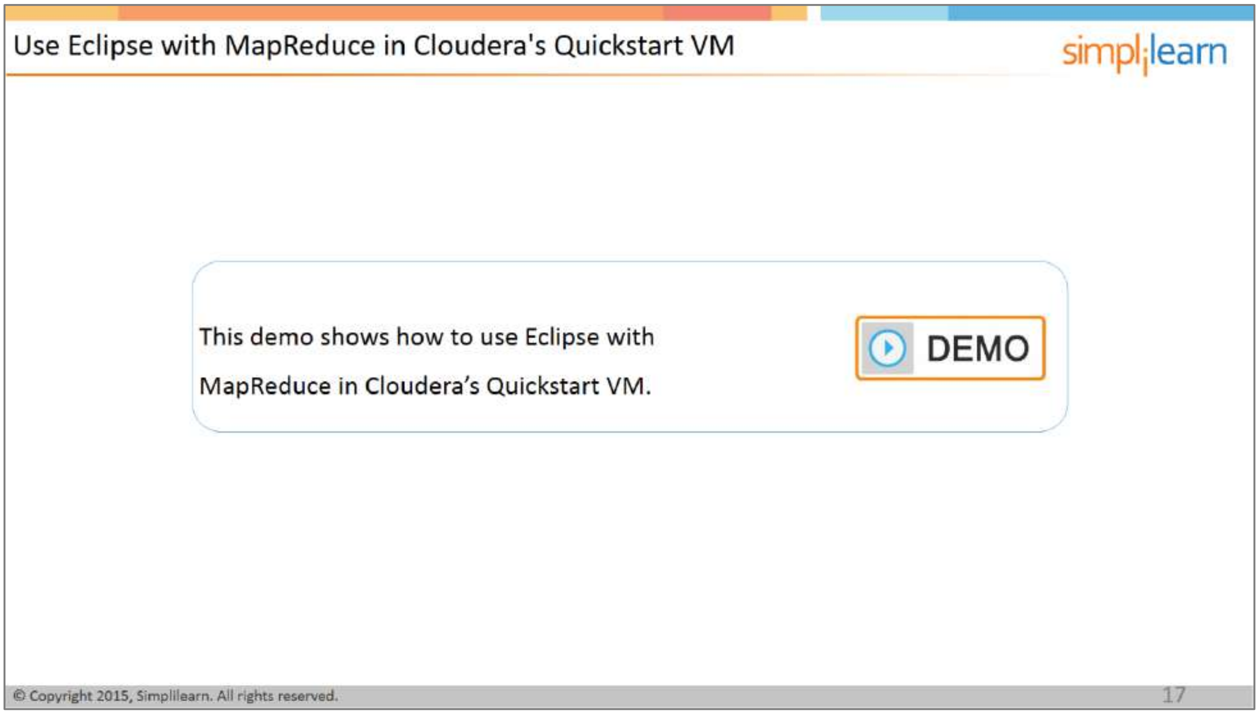
As a fast growing company, Nutri Worldwide, Inc. realizes the need to advance to the next level in their usage of Hadoop. Olivia, the EVP of IT Operations, has been successfully piloting a variety of key features and tools for Hadoop. Thus, the company decides to use the Cloudera enterprise version to leverage efficiencies and Return on Investment or ROI.



Download, Start, and Work with Cloudera VM

Let us summarize the steps performed:

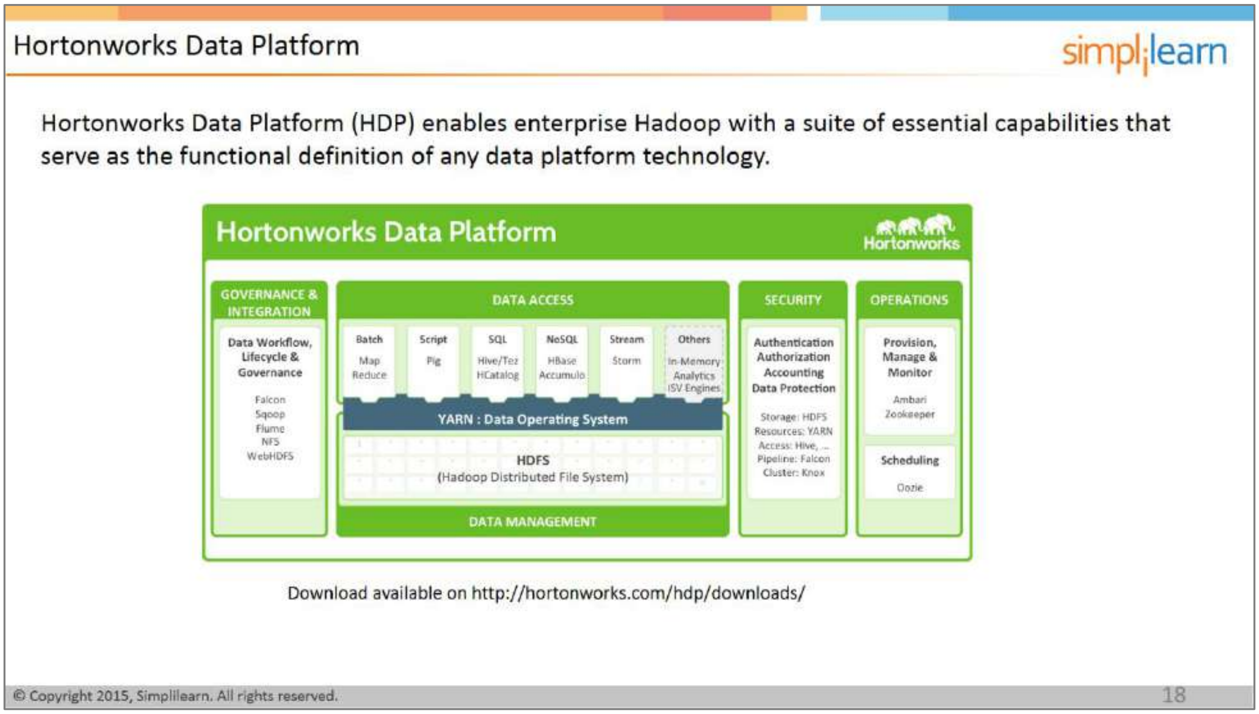
1. Go to cloudera's website: cloudera.com.
2. Select the required version of VM.
3. Download the required VM and extract the zip file.
4. You will notice an ovf file in the contents of the extracted folder.
5. Open Oracle VirtualBox Manager and select the **Import Appliances** menu item.
6. Select the VM file
7. Select the downloaded 'ovf' file.
8. Select Import to import the VM.
9. Once imported, select the VM from the left hand panel and click Start to launch the VM.
10. Once the VM starts booting a browser will open. Click Hue in the browser bookmarks.
11. Sign in to Hue. Enter user ID as 'cloudera' and password as 'cloudera'.



Use Eclipse with MapReduce in Cloudera's Quickstart VM

Let us summarize the steps performed:

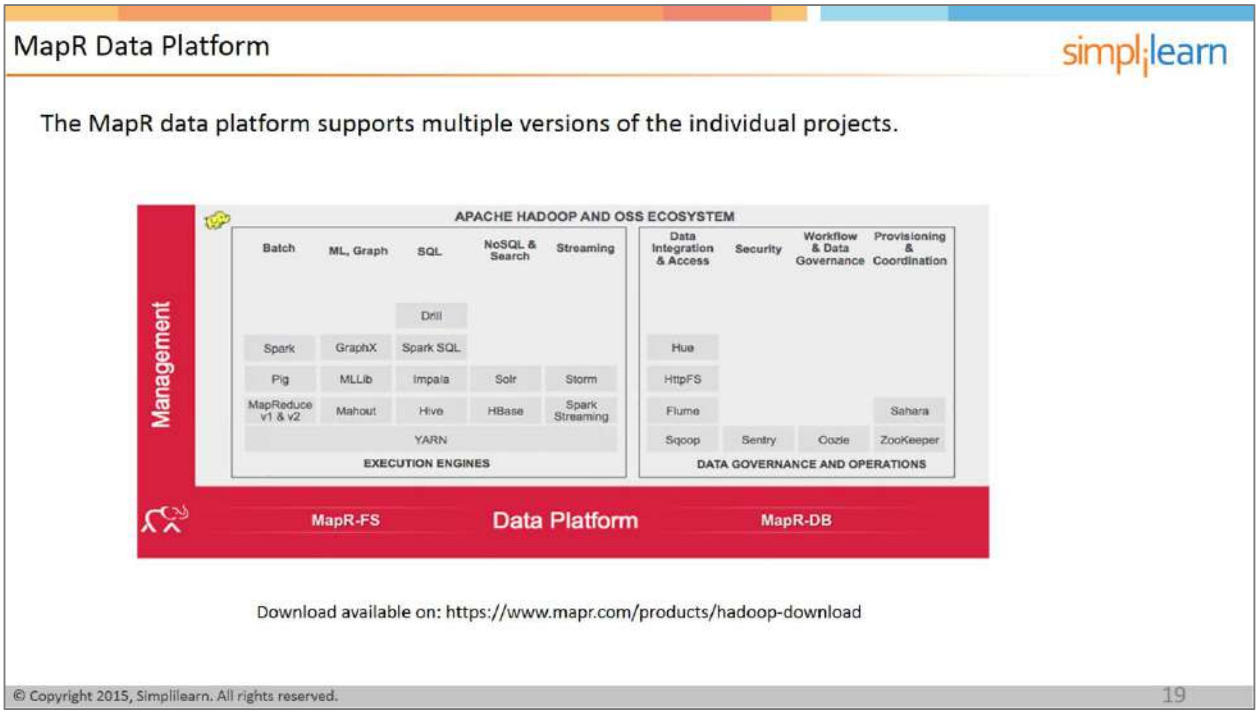
1. Start Hue. Click the **Hue** option.
2. Click the **File Browser** icon and load data in the specified folder.
3. Click the **Select Files** button to upload the files.
4. Browse the location and select the particular file.
5. Click **Open** and Cloudera is successfully loaded.
6. Click New > Java Project to create a new Java project.
7. Select the **Libraries** folder.
8. Click **Add External JARs…** to add Hadoop external jar files.
9. Navigate to the specified location and add the JAR file.
10. Open WordCount.java, which you used in the earlier MapReduce demo.
11. Click the Play icon to run the demo.
12. You will see that the code runs successfully and creates the output folder. Click the output1 folder.
13. You have successfully used eclipse in Cloudera.



Hortonworks Data Platform

Hortonworks Data Platform or HDP, enables Enterprise Hadoop with a suite of essential capabilities. The suite serves as the functional definition of any data platform technology. It has a comprehensive set of capabilities aligned to functional areas such as data management, data access, data governance and integration, security, and operations.

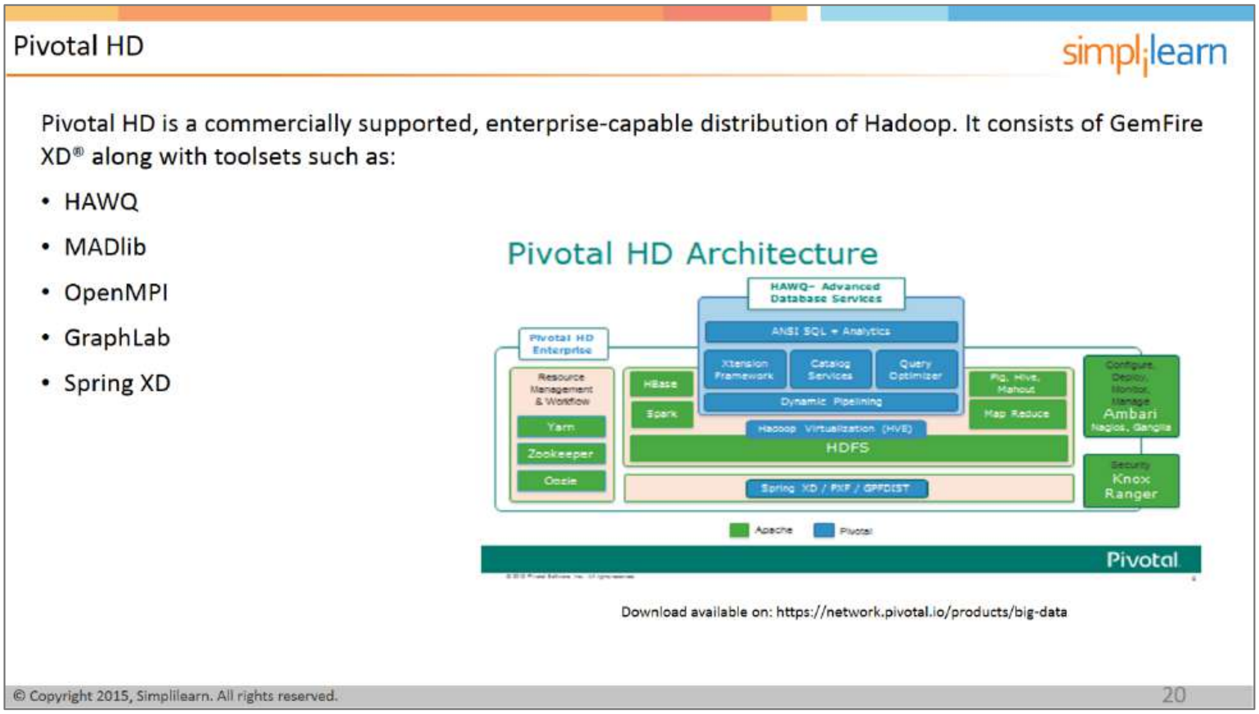
HDP can be downloaded from the URL mentioned.



MapR Data Platform

The MapR data platform supports more than 20 open source projects. It also supports multiple versions of the individual projects. This allows users to migrate to the latest versions at their own pace. The screen shows all the projects actively supported in the current General Availability or GA, version of MapR Distribution for Hadoop—M7.

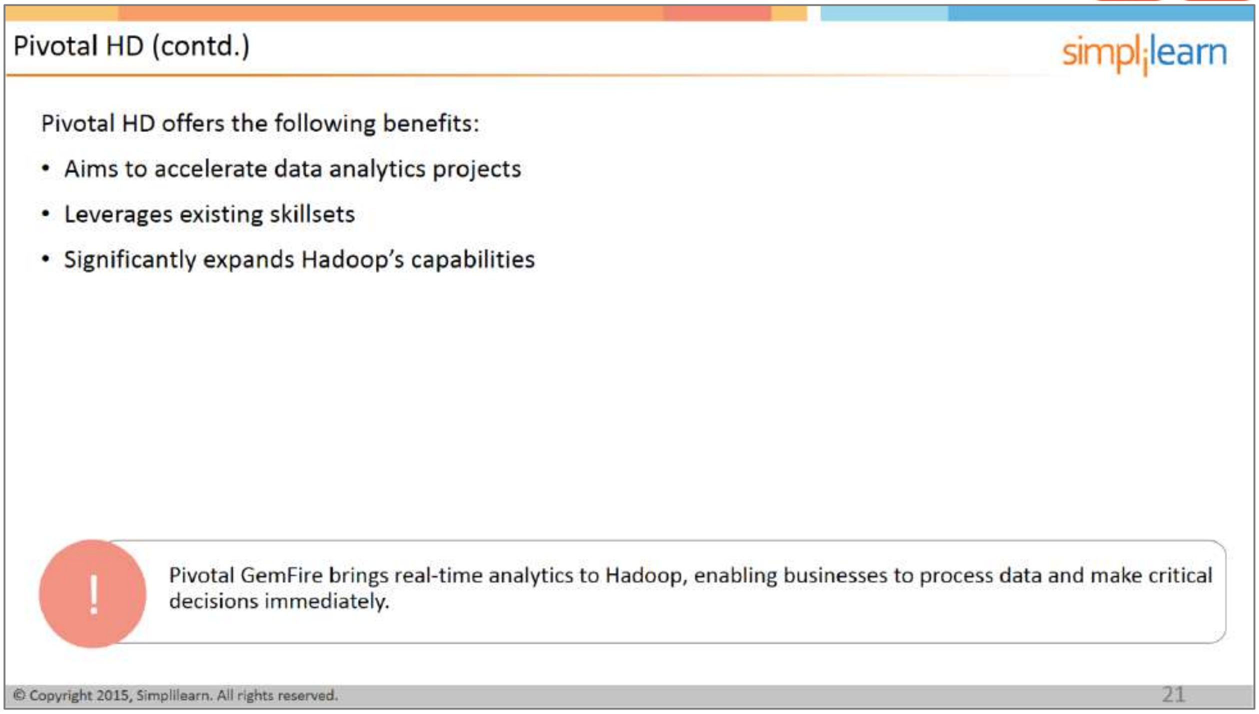
MapR can be downloaded from the mentioned URL.



Pivotal HD

Pivotal HD is a commercially supported, enterprise-capable distribution of Hadoop. It consists of GemFire XD® along with toolsets such as HAWQ, MADlib, OpenMPI, GraphLab, and Spring XD.

Pivotal HD can be downloaded from the mentioned URL.



Pivotal HD (contd.)

Pivotal HD offers the following benefits:

- It aims to accelerate data analytics projects.
- It leverages existing skillsets.
- It significantly expands Hadoop's capabilities.

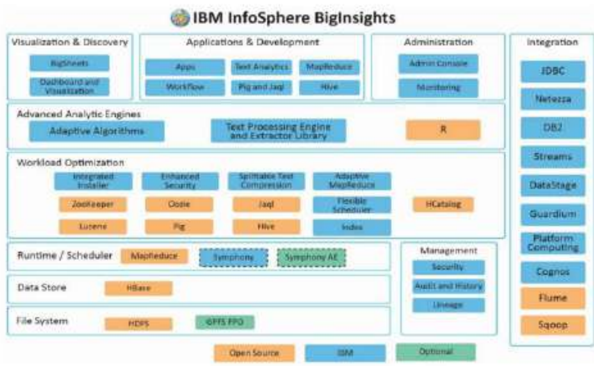
Pivotal GemFire brings real-time analytics to Hadoop, enabling businesses to process data and make critical decisions immediately.

IBM InfoSphere BigInsights

IBM offers InfoSphere BigInsights family of Hadoop distribution offerings.

Benefits of InfoSphere BigInsights:

- Helps organizations transform large, complex data volumes into insights
- Incorporates tooling for numerous users



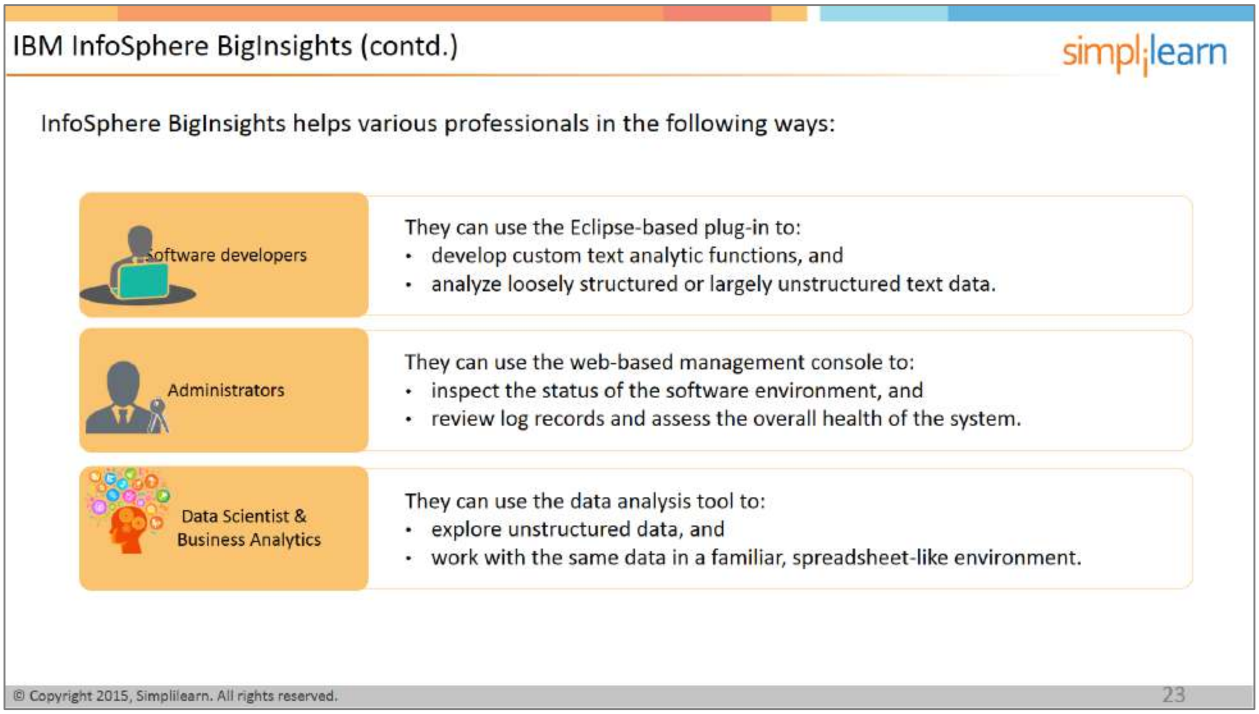
Download available on: http://www-01.ibm.com/software/data/infosphere/hadoop/trials.html

 22

IBM InfoSphere BigInsights

IBM offers InfoSphere BigInsights family of Hadoop distribution offerings. InfoSphere BigInsights extend the value of open-source Hadoop for data processing, warehousing, and analytics. It enables organizations to transform large, complex data volumes into insights. It incorporates tooling for numerous users which increases time to value speeds. It also simplifies development and maintenance of data processing.

InfoSphere BigInsights can be downloaded from the URL shown.



IBM InfoSphere BigInsights (contd.)

InfoSphere BigInsights helps software developers, administrators, data scientists, and business analysts.

- Software developers can use the Eclipse-based plug-in to develop custom text analytic functions to analyze loosely structured or largely unstructured text data.
- Administrators can use the web-based management console to inspect the status of the software environment, review log records, assess the overall health of the system, and more.
- Data scientists and business analysts can use the data analysis tool to explore and work with unstructured data in a familiar spreadsheet-like environment.

QUIZ 1

What is the role of Hue in CDH?

a. To provide access to admin panel

b. To provide access to tools like Pig, Hive, and HBase

c. To provide access to Cloudera Manager

d. To start Apache Hadoop

 25

QUIZ 2

Which of the following tools in CDH is used by administrators to check the status of HDFS?

a. Hue

b. Sqoop

c. Cloudera Manager

d. NodeManager

 27

QUIZ 3

Which of the following areas of CDH can be used for starting the Hive service?

a. Hue

b. Sqoop

c. Cloudera Manager

d. NodeManager

 29

QUIZ 4

Which layer of Hadoop distribution is formed by HBase?

a. SQL

b. Streams

c. NoSQL

d. Search

 31

QUIZ 5

Identify the correct utility in Hadoop distributions which helps to schedule workflow and jobs.

a. Oozie

b. Sqoop

c. Flume

d. Hadoop Streaming

 33

QUIZ 6

Which of the following tools in CDH is used for performing MapReduce development using Java programming?

a. Eclipse

b. NetBeans

c. JCreator

d. JUnit

 35

QUIZ 7

Which of the following companies is a provider of Hadoop distribution?

a. IBM

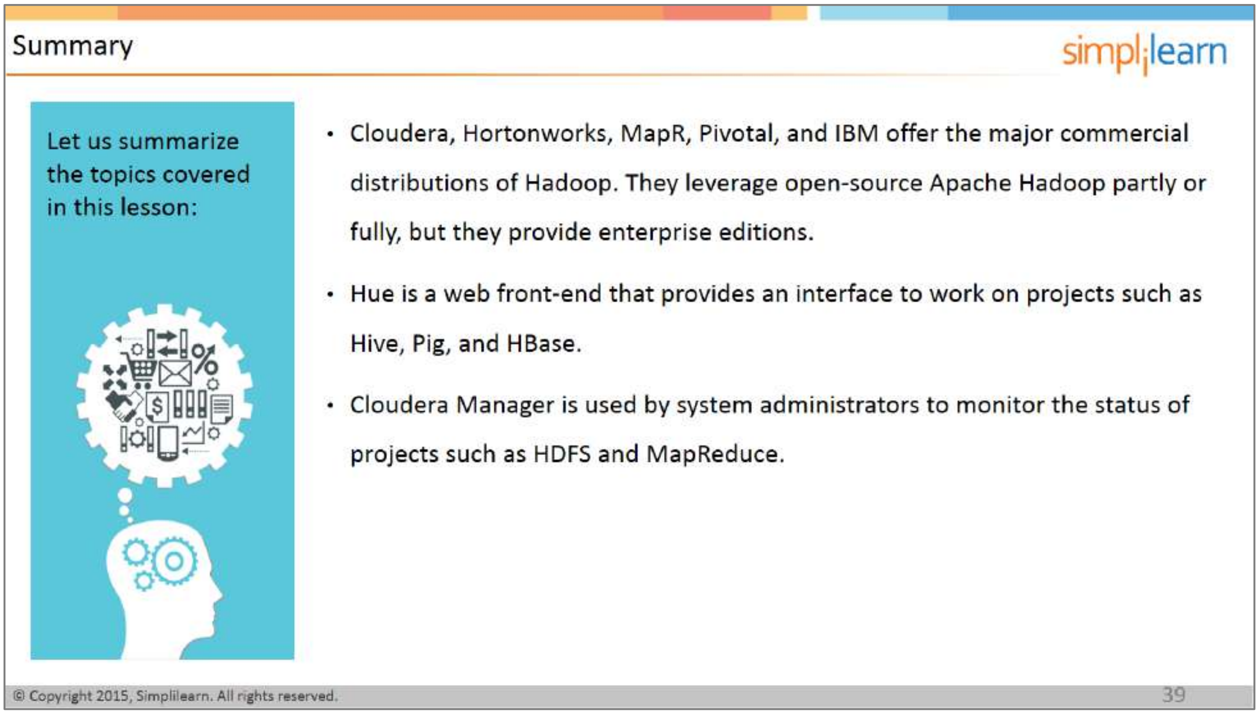
b. Red Hat

c. Accenture

d. SAP

ANSWERS:

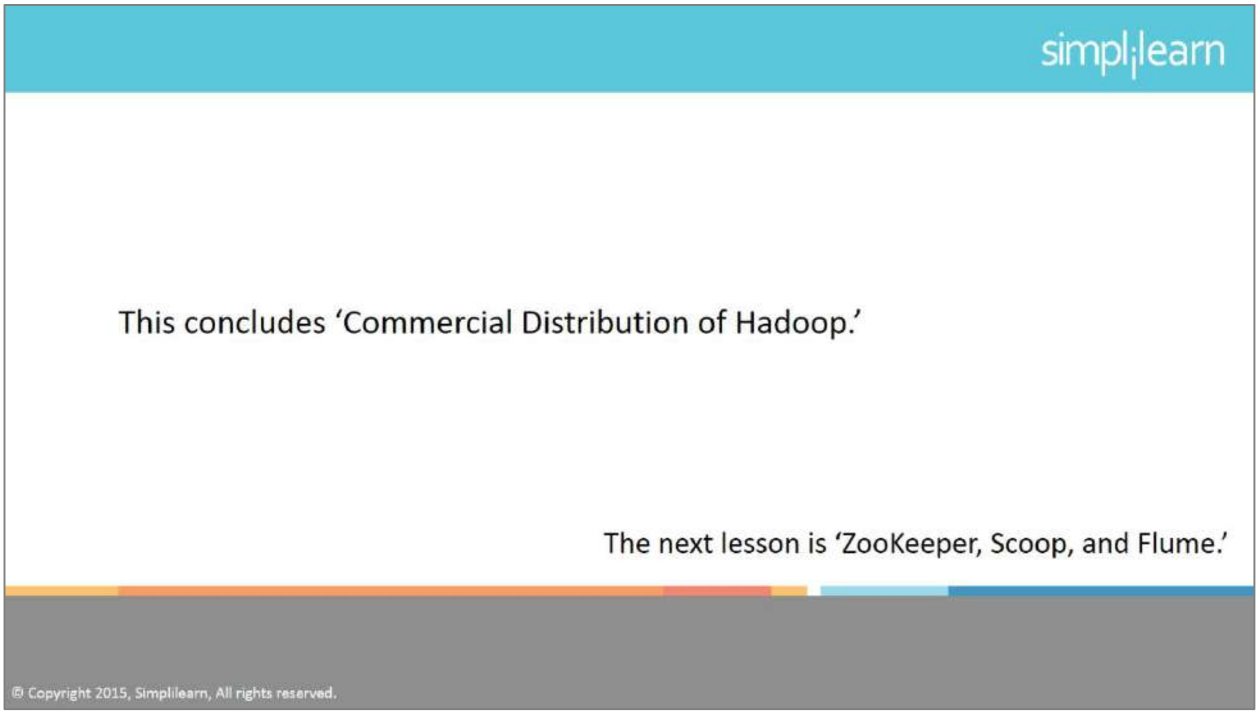
S.No.	Question	Answer & Explanation
1	What is the role of Hue in CDH?	Hue is a web front end which is used by a Hadoop developer to access applications such as Pig, Hive, and HBase.
2	Which of the following tools in CDH is used by administrators to check the status of HDFS?	Cloudera Manager is used by administrators to check the status of HDFS.
3	Which of the following areas of CDH can be used for starting the Hive service?	Hue provides an interface to start Hive service.
4	Which layer of Hadoop distribution is formed by HBase?	HBase is a popular NoSQL database and is a part of Hadoop distributions.
5	Identify the correct utility in Hadoop distributions which helps to schedule workflow and jobs.	Oozie is used for workflow scheduling.
6	Which of the following tools in CDH is used for performing MapReduce development using Java programming?	The Eclipse tool is used for performing MapReduce development using Java programming.
7	Which of the following companies is a provider of Hadoop distribution?	IBM provides commercial distribution of Hadoop by the name of InfoSphere BigInsights.



Summary

Following is a summary of the topics covered in this lesson:

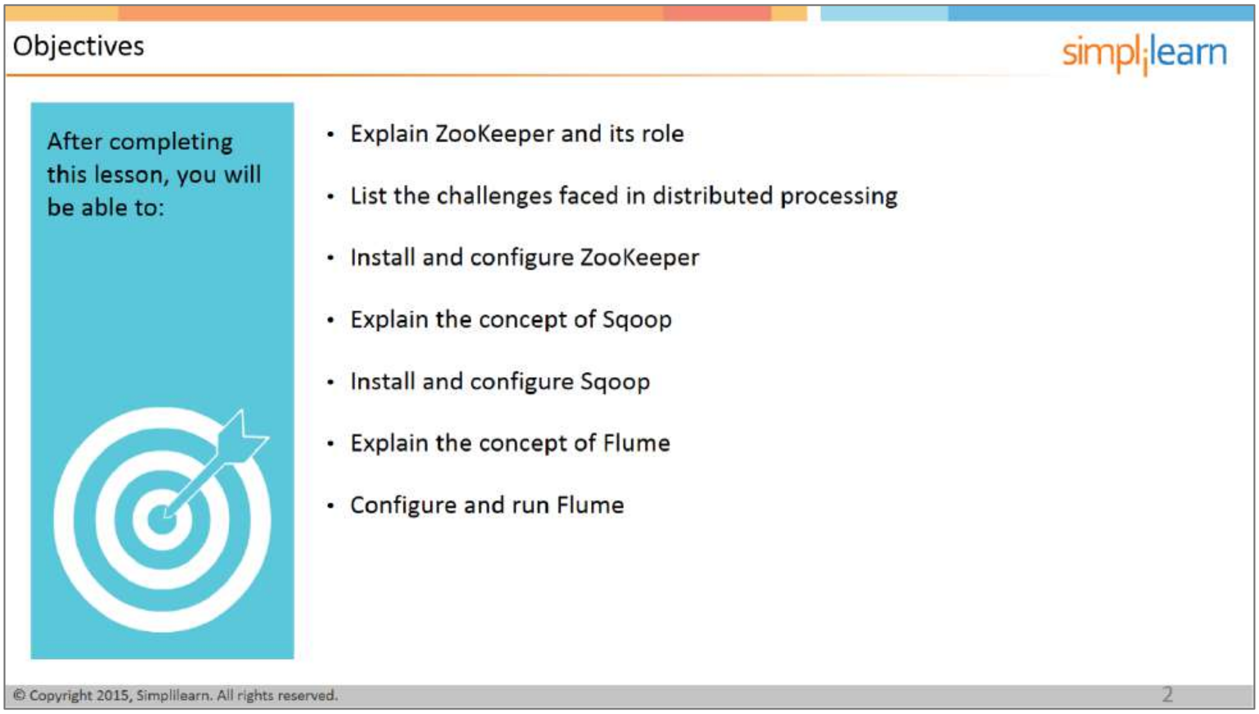
- Cloudera, Hortonworks, MapR, Pivotal, and IBM offer the major commercial distributions of Hadoop. They leverage open-source Apache Hadoop partly or fully, but they provide enterprise editions in addition to it.
- Hue is a web front-end that provides an interface to work on projects like Hive, Pig, HBase, and others.
- Cloudera Manager is used by system administrators to monitor the status of projects such as HDFS and MapReduce.



Conclusion

This concludes 'Commercial Distribution of Hadoop.' In the next lesson, we will focus on 'ZooKeeper, Scoop, and Flume.'

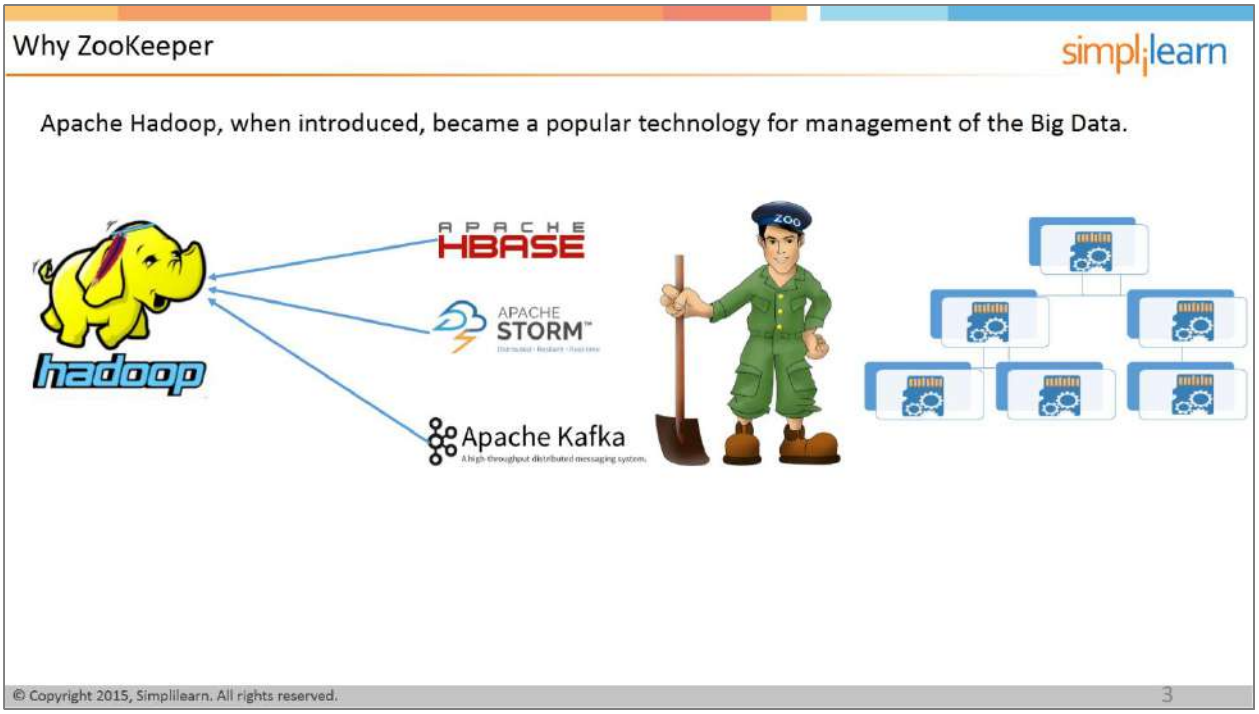
Lesson 10—ZooKeeper, Sqoop, and Flume



Objectives

After completing this lesson, you will be able to:

- Explain ZooKeeper and its role
- List the challenges faced in distributed processing
- Install and configure ZooKeeper
- Explain the concept of Sqoop
- Install and configure Sqoop
- Explain the concept of Flume
- Configure and run Flume



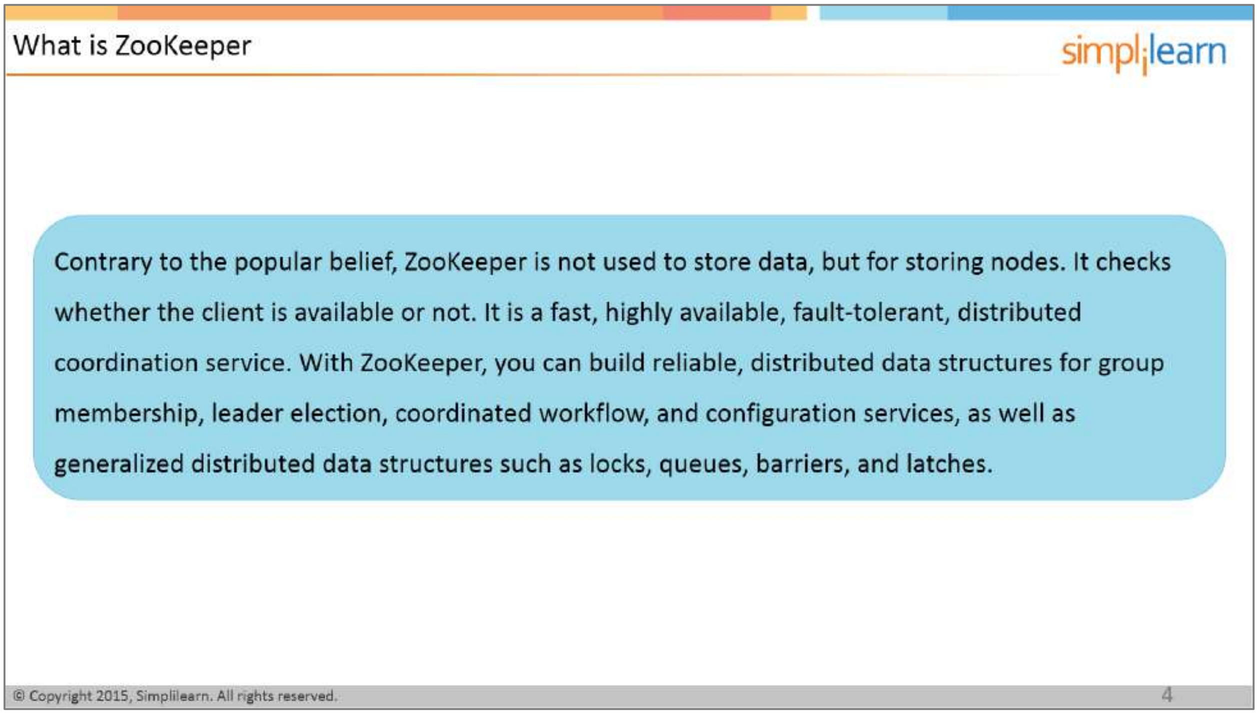
Why ZooKeeper

Apache Hadoop, when introduced, became a popular technology for management of the Big Data. Consequently, different services like HBase, Storm, and Kafka were added as part of a Hadoop cluster. With increasing services and computing nodes, there was a need for the integrated management.

This is where ZooKeeper is used in out-of-work services to support high availability of data processing.

Consider a distributed system with multiple servers, which are responsible for holding data and perform operations on that data.

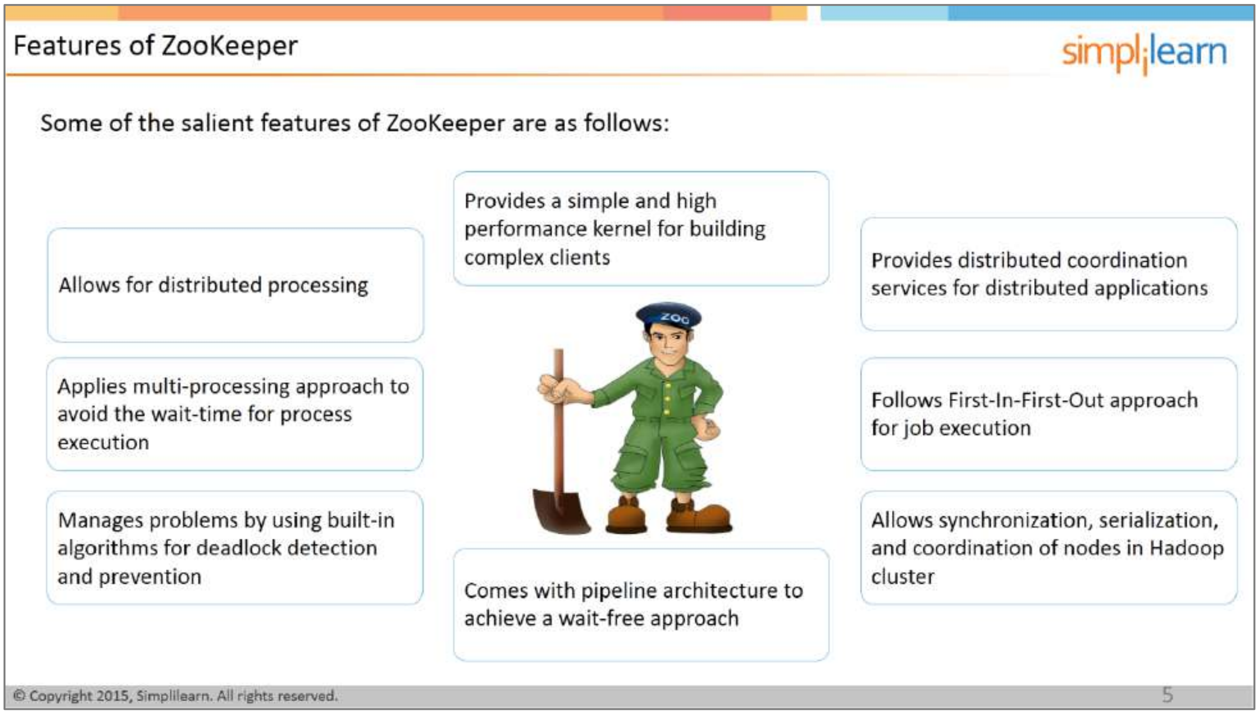
How do you determine the servers that are alive and operating at any moment of time? Or, how do you determine the servers available to process a build in a distributed build system?



What is ZooKeeper

Contrary to the popular belief, ZooKeeper is not used to store data, but for storing nodes. It checks whether the client is available or not.

It is a fast, highly available, fault-tolerant, distributed coordination service. With ZooKeeper, you can build reliable, distributed data structures for group membership, leader election, coordinated workflow, and configuration services, as well as generalized distributed data structures such as locks, queues, barriers, and latches.



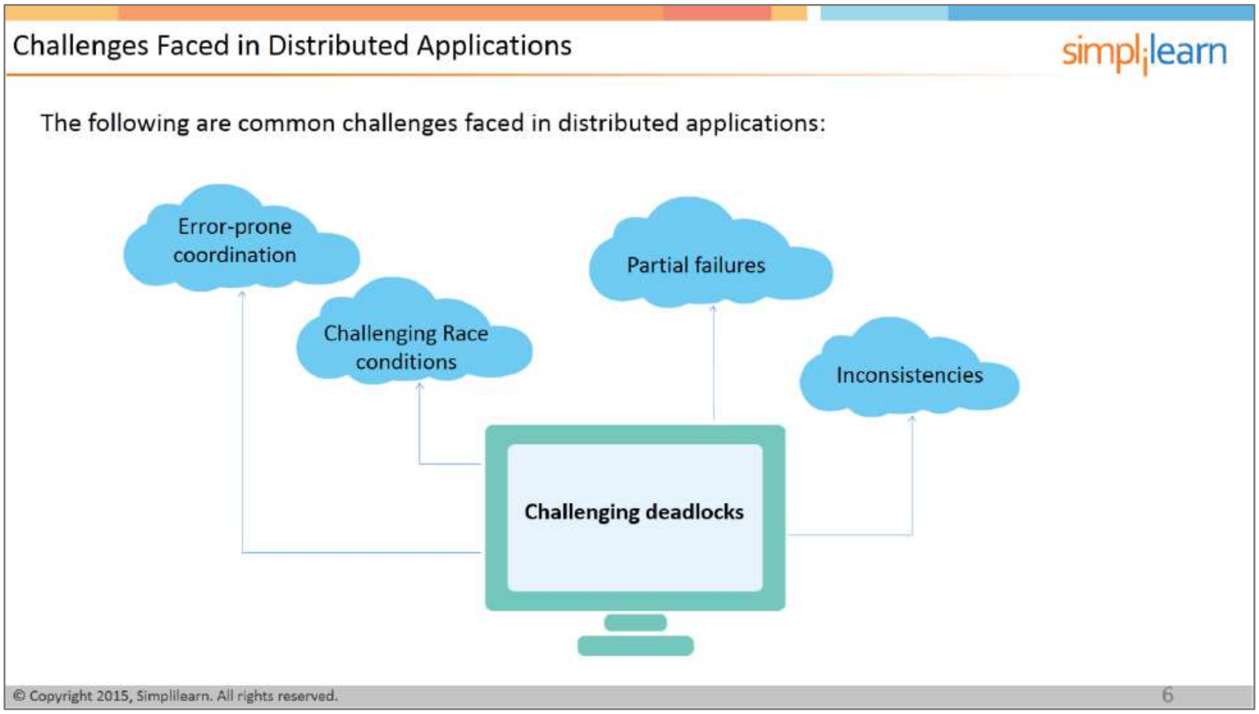
Features of ZooKeeper

Some of the salient features of ZooKeeper are as follows:

ZooKeeper provides a simple and high performance kernel for building complex clients. It also provides distributed coordination services for distributed applications.

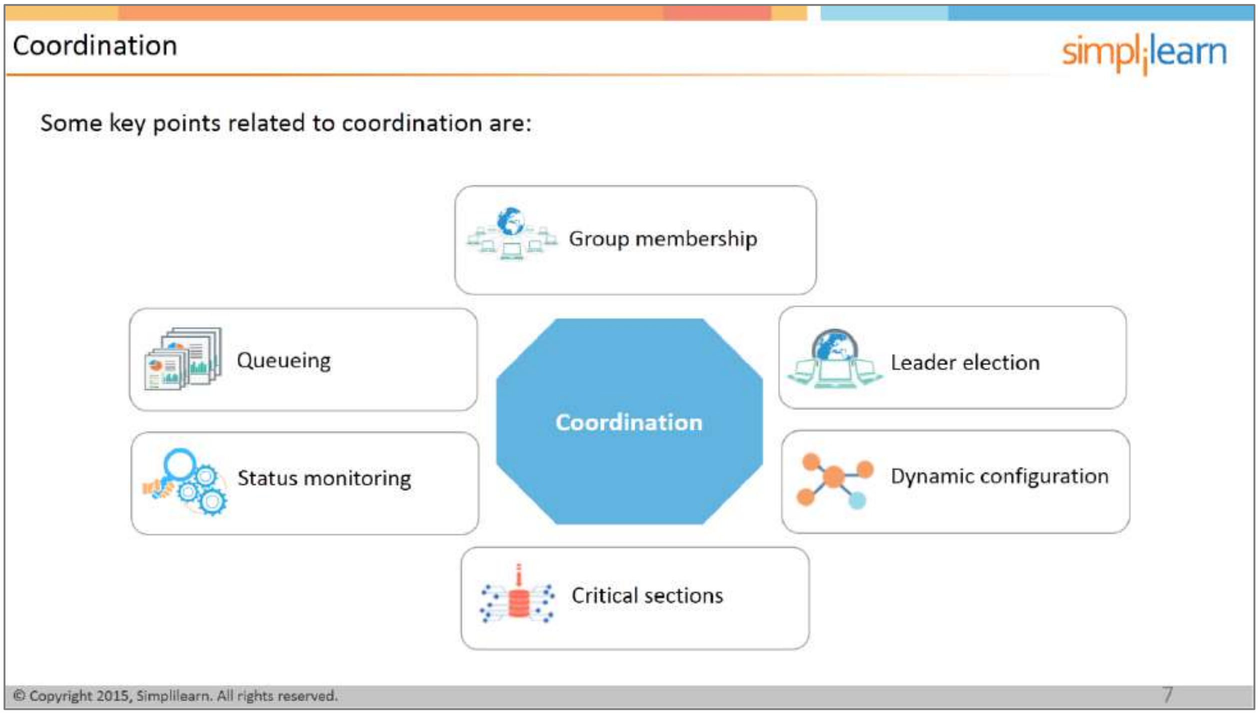
ZooKeeper follows FIFO, the First-In-First-Out Approach, when it comes to job execution. It allows for synchronization, serialization, and coordination of nodes in a Hadoop cluster. It comes with pipeline architecture to achieve a wait-free approach.

Further, ZooKeeper manages problems by using built-in algorithms for deadlock detection and prevention. It applies a multi-processing approach to reduce wait-time for process execution. In addition, ZooKeeper allows for distributed processing. Thus, it is compatible with services related to MapReduce.



Challenges Faced in Distributed Applications

There are certain challenges faced in distributed applications. Coordination of nodes in a cluster is error-prone. The race conditions when it comes to job execution and synchronization and the deadlocks with respect to detection and prevention are challenging. There are chances of partial failure of job execution, resulting in the restarting of all tasks. Further, there are inconsistencies due to hardware or service failures.



Coordination

Some of the key points related to coordination are as follows:

Group membership: It refers to the introduction of a new node in the cluster to perform synchronization and distributed jobs.

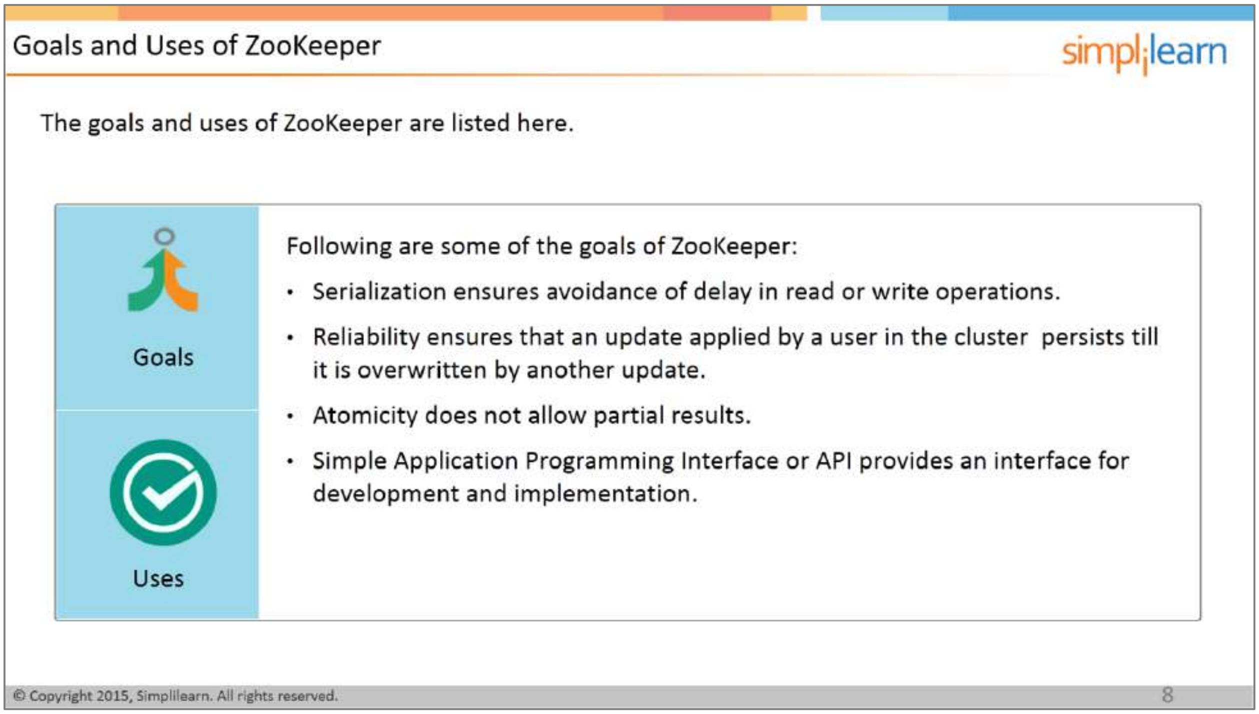
Leader election: In a distributed environment, leader election is the selection of a leader node in a cluster to enable job allocation and monitoring.

Dynamic configuration: It refers to flexibility in configuring a node according to the type of job to be executed.

Critical sections: These are the areas in distributed processing used to store the global variables required to perform job executions.

Status monitoring: This refers to monitoring the status of nodes in the cluster process heartbeat, RAM synchronization, and others.

Queuing: This refers to the generation of a queuing mechanism to perform multiple jobs at a stipulated time.



Goals and Uses of ZooKeeper

The goals and uses of ZooKeeper are described here.

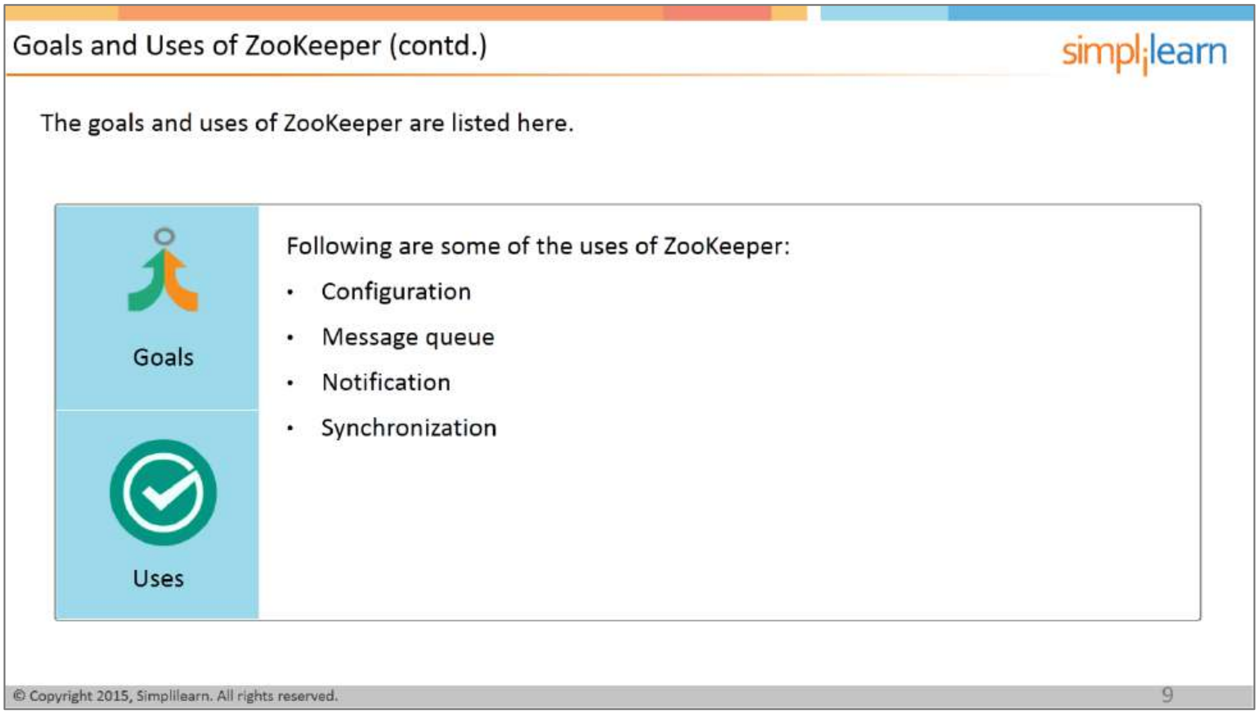
There are a number of goals of ZooKeeper, such as:

Serialization ensures avoidance of delay in read or write operations;

Reliability ensures that an update applied by a user in the cluster persists till it is overwritten by another update;

Atomicity does not allow partial results. Any user update can either succeed or fail; and

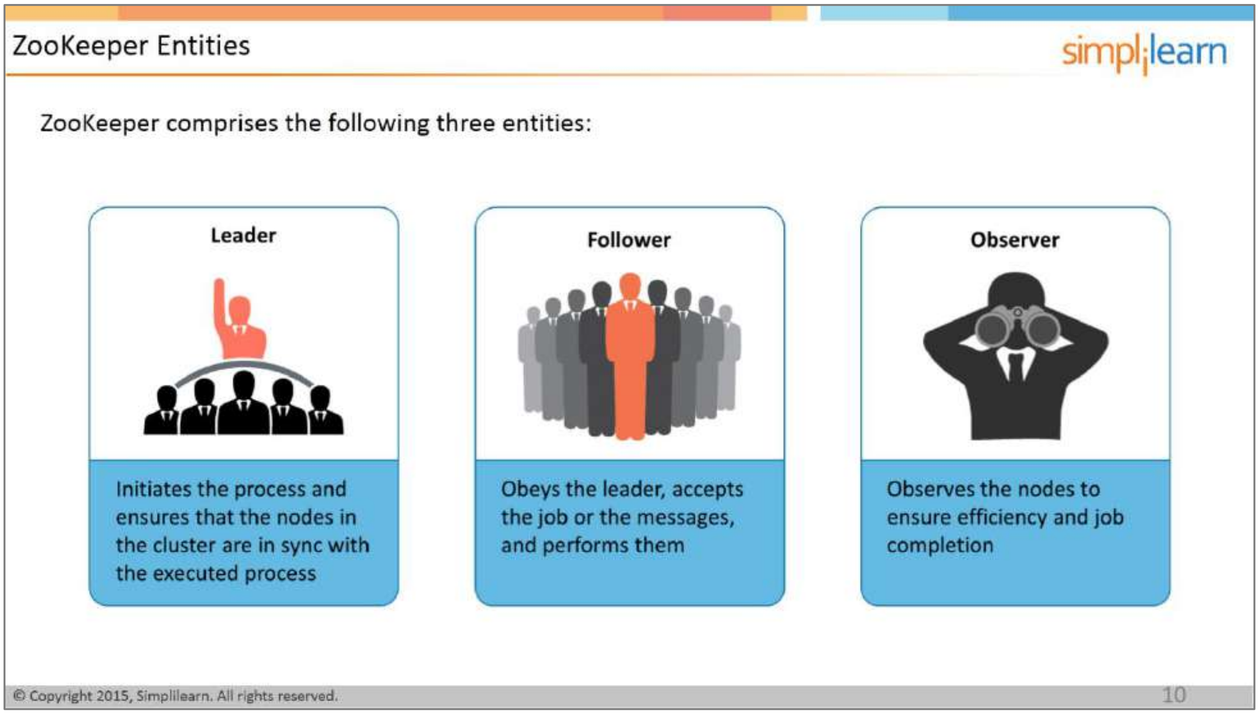
Simple Application Programming Interface or API provides an interface for development and implementation.



Goals and Uses of ZooKeeper (contd.)

There are many uses of ZooKeeper. Some of them are:

Configuration ensures the nodes in the cluster are in sync with each other and also with the NameNode server. Message queue is the communication with nodes present in the cluster. Notification refers to the process of notifying the NameNode of any failure in the cluster so that the specific task can be restarted from another node. Further, synchronization ensures all nodes in the cluster are in sync with each other, and the services are working well.



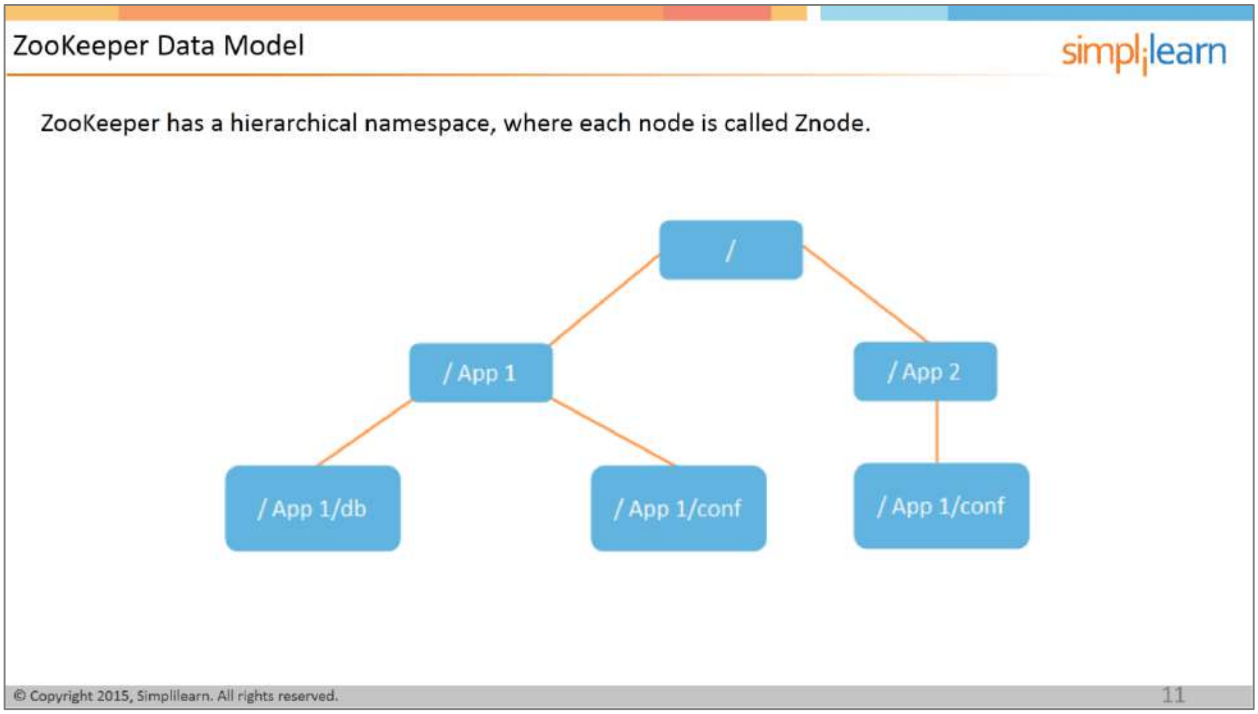
ZooKeeper Entities

ZooKeeper comprises three entities: leader, follower, and observer.

Leader, as a system, is responsible for initiating the process and ensuring that the nodes in the cluster are in sync with the process that is executed. Only one leader can exist in a cluster.

Follower is the system that obeys the leader, accepts the job or the messages from the leader, and performs them. There can be several followers in a cluster.

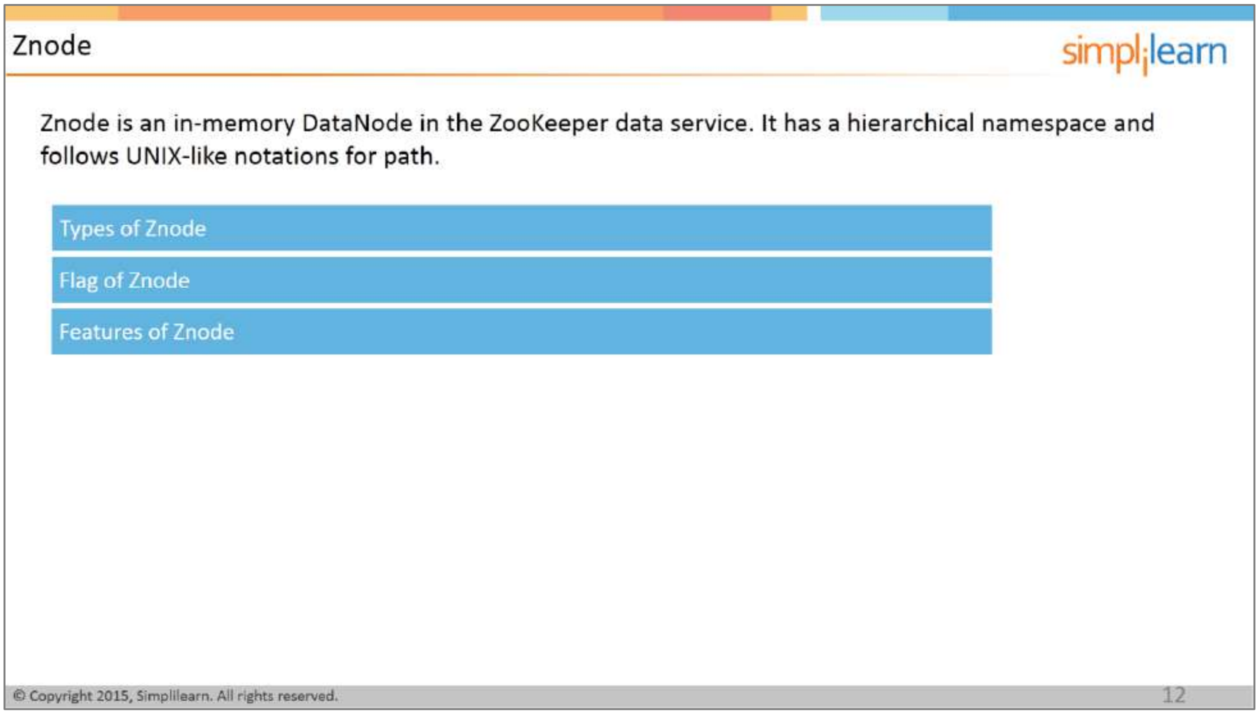
Observer is a system that observes the nodes to ensure efficiency and job completion. The observer helps the leader to assign specific types of jobs to different nodes ensuring that the busy nodes do not receive multiple jobs.



ZooKeeper Data Model

ZooKeeper has a hierarchical namespace, where each node is called Znode.

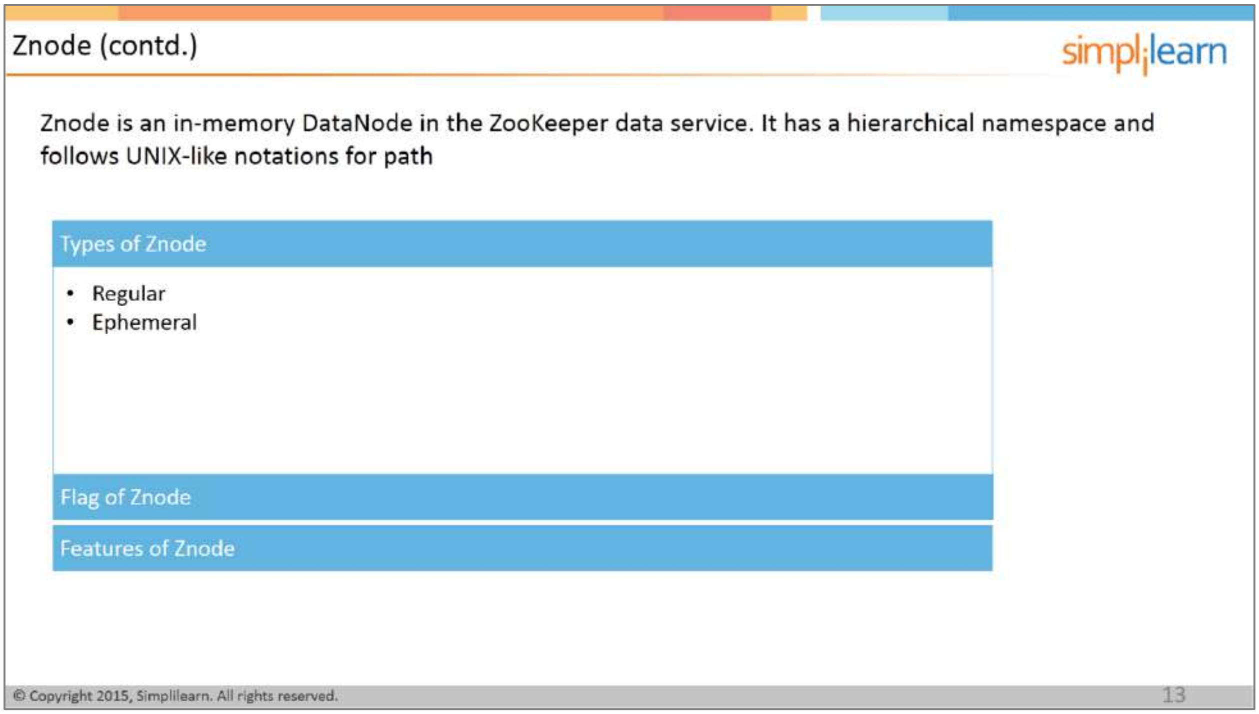
An example is shown on the image, where the tree diagram represents the namespace. The tree follows a top-down approach where '/' is the root, and App1 and App2 reside in the root. The path to access the db is /App1/db, which is called the hierarchical path. Similarly, for accessing conf under App1, the path is /App/conf and under App 2 the path is '/App2/conf.



Znode

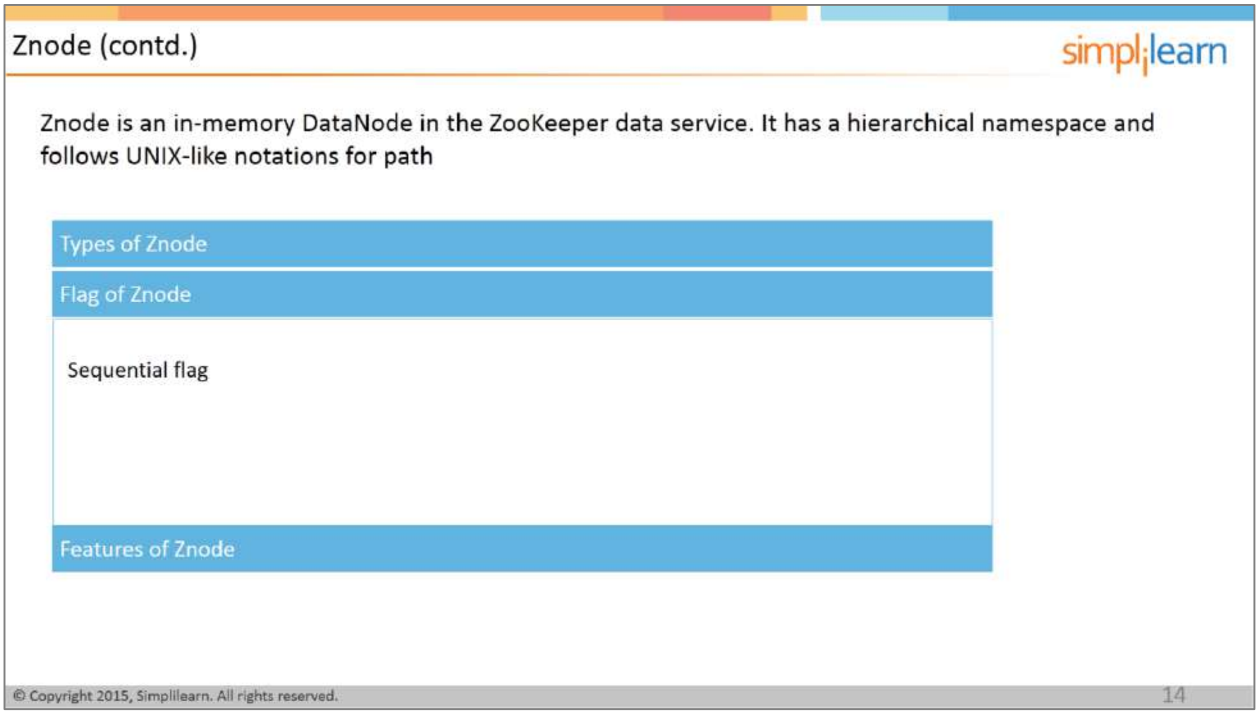
Znode is an in-memory data node in the ZooKeeper data service. It has a hierarchical namespace and follows UNIX-like notations for path.

The screen further examines the contents related to Znode.



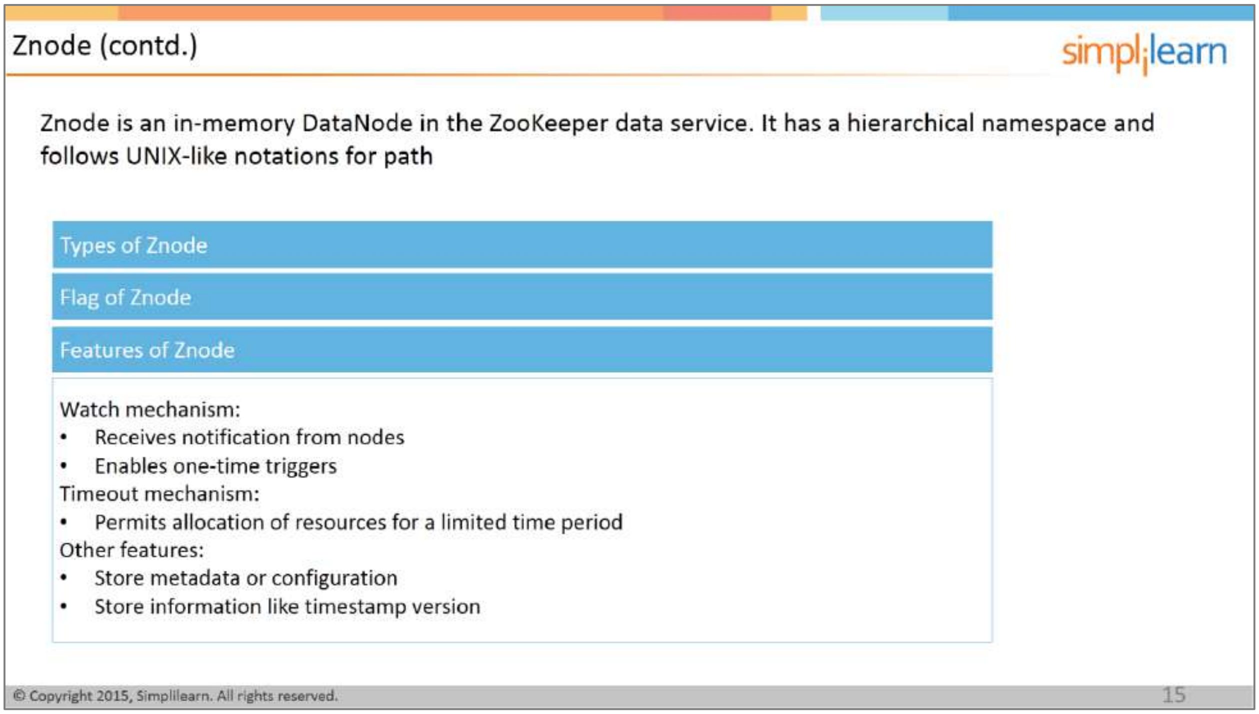
Znode (contd.)

Types of Znodes: There are two types of Znodes, Regular Znode and Ephemeral Znode.



Znode (contd.)

Flag of Znode: Sequential flag is the only flag of Znode that is responsible for accessing data sequentially.

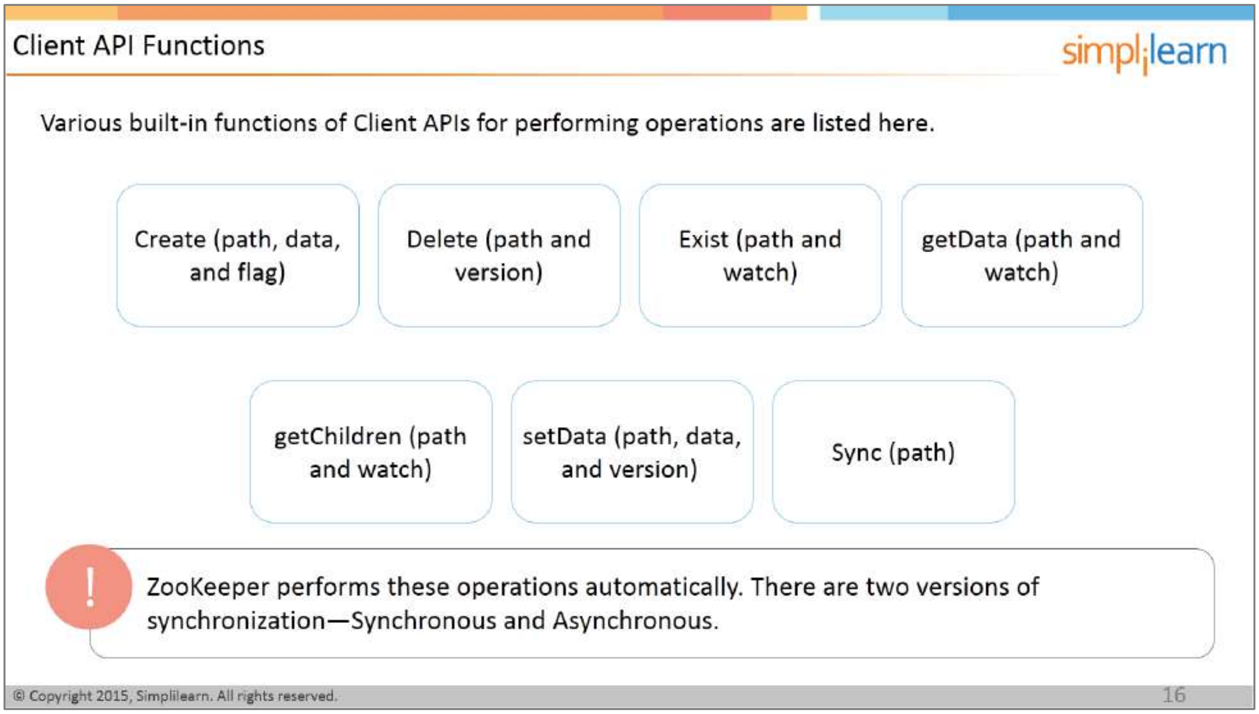


Znode (contd.)

Features of Znode: Watch mechanism is one of the features of Znode. The watch mechanism receives notifications from nodes that reside in the cluster. It also enables one-time triggers for job execution.

Timeout mechanism is also a feature of Znode, where session can be used. Session is a term applied for a connection to a server from the client. The timeout mechanism permits allocation of resources for a limited time period.

Znode is not designed for data storage; it stores metadata or configuration in the leader system; and it also stores information such as timestamp version.

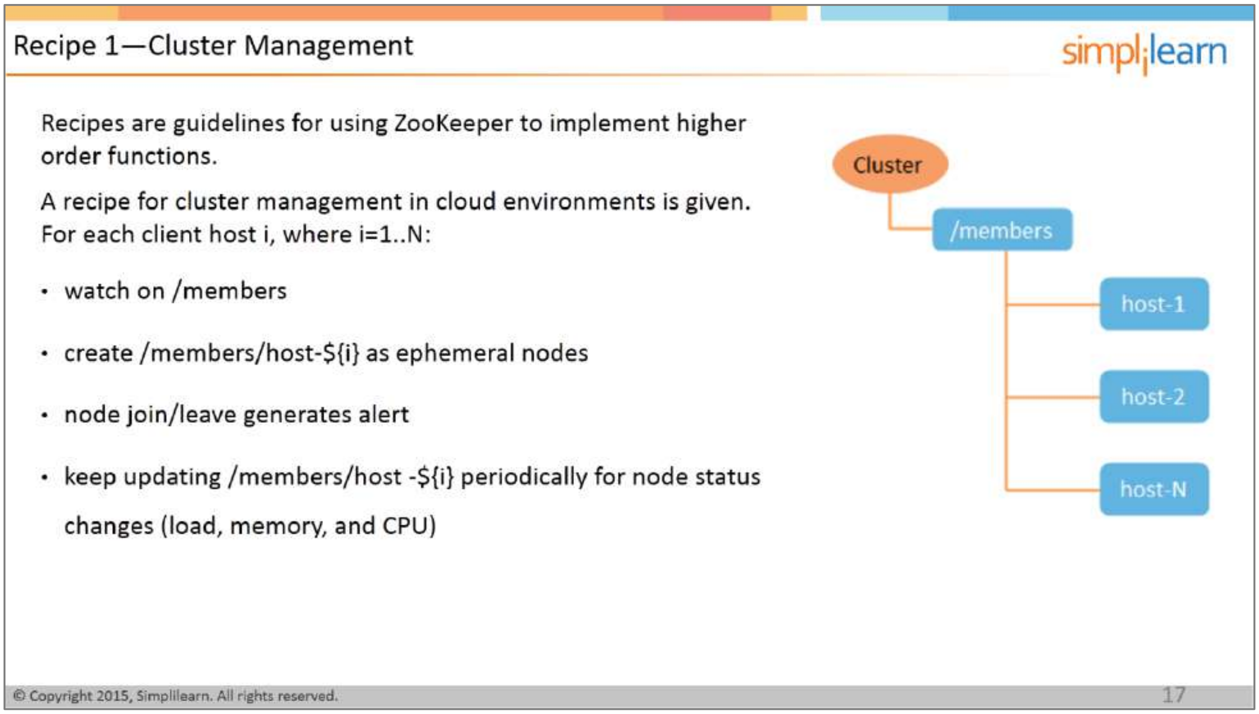


Client API Functions

Client APIs have built-in functions such as create, delete, exist, getData, getChildren, setData, and Sync for performing operations. The parameters for each function are as follows:

Path, data, and flag for create function; path and version for delete function; path and watch for exist, getData, and getChildren functions; path, data, and version for setData function; and path for sync function.

ZooKeeper performs these operations automatically. However, a user can change the behavior of ZooKeeper by using any of these functions. There are two versions of synchronization. They are synchronous and asynchronous.

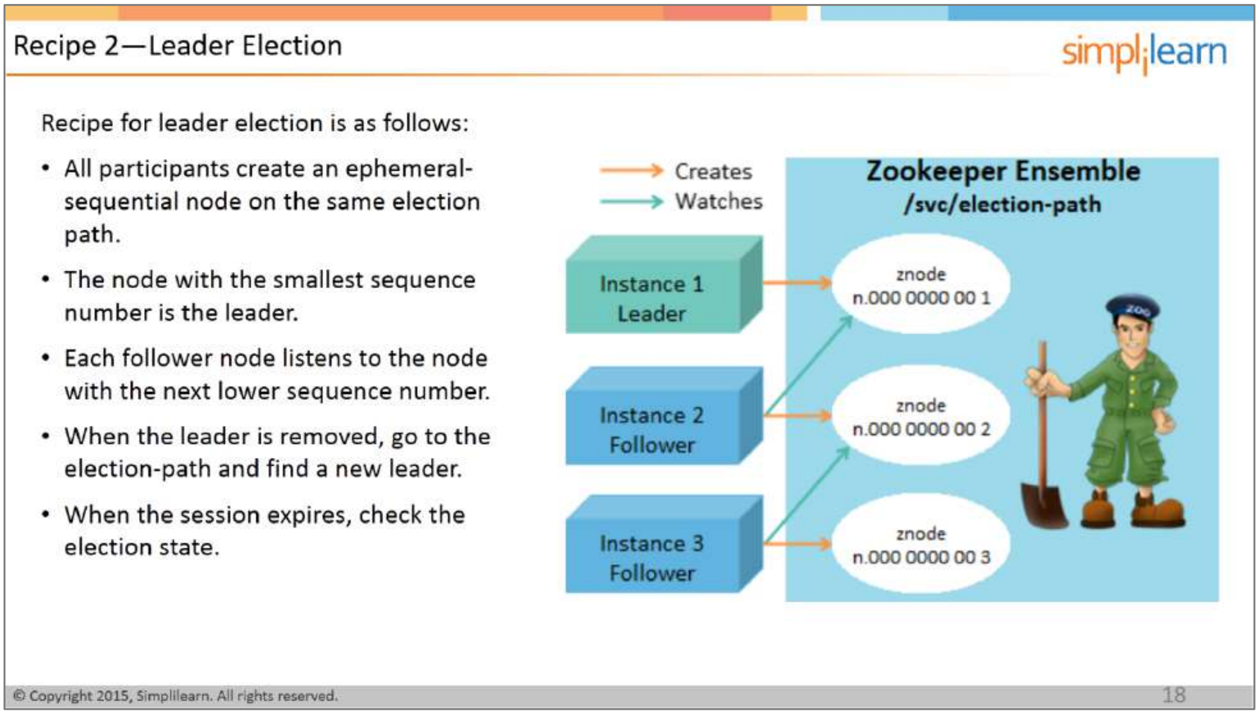


Recipe 1—Cluster Management

Recipes are the guidelines for using ZooKeeper to implement higher order functions.

The following is a recipe for cluster management used in cloud environments:

- For each client host i, where i equals 1 to N:
- Watch on /members
- Create slash members slash host array as ephemeral nodes
- A node joining or leaving cluster generates an alert
- Keep updating slash members slash host array periodically for node status changes. For example, change in load, memory, and CPU.

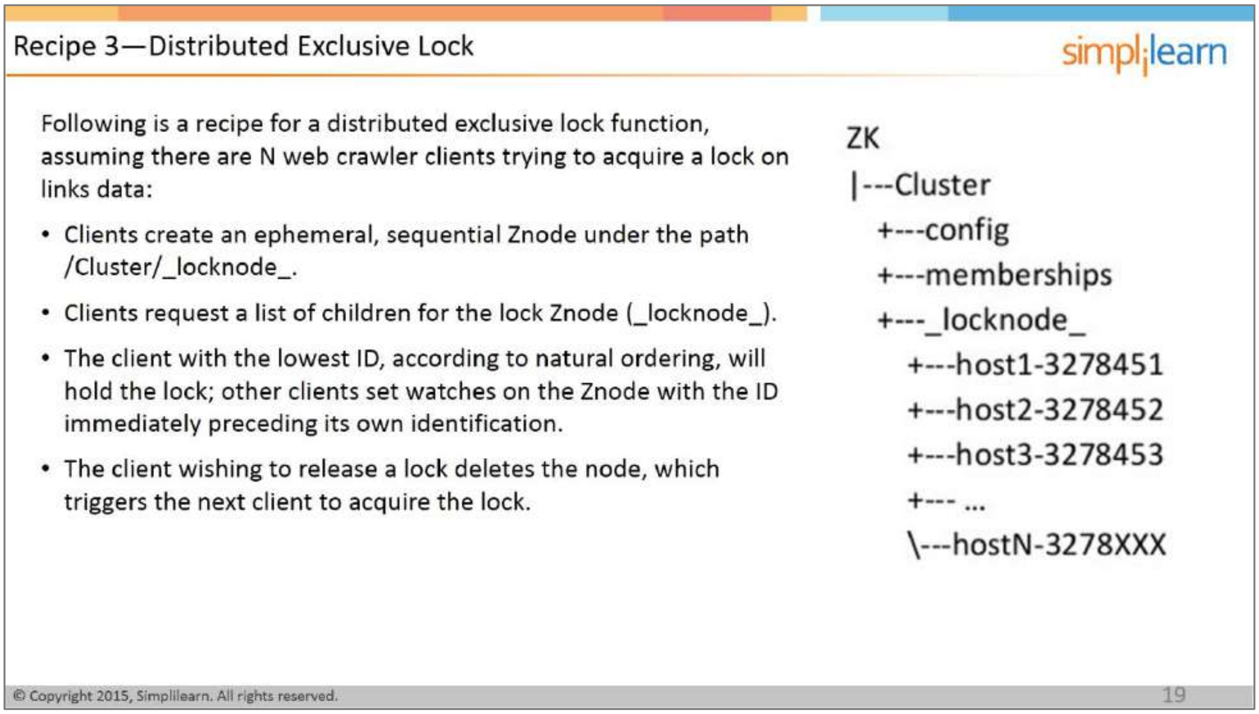


Recipe 2—Leader Election

The following is an example of a recipe for leader election:

- All participants of the election process create an ephemeral-sequential node on the same election path.
- The node with the smallest sequence number is the leader.
- Each follower node listens to the node with the next lower sequence number.
- When the leader is removed, go to election-path and find a new leader; the node with the lowest sequence number becomes the leader.
- When the session expires, check the election state, and go to election if needed.

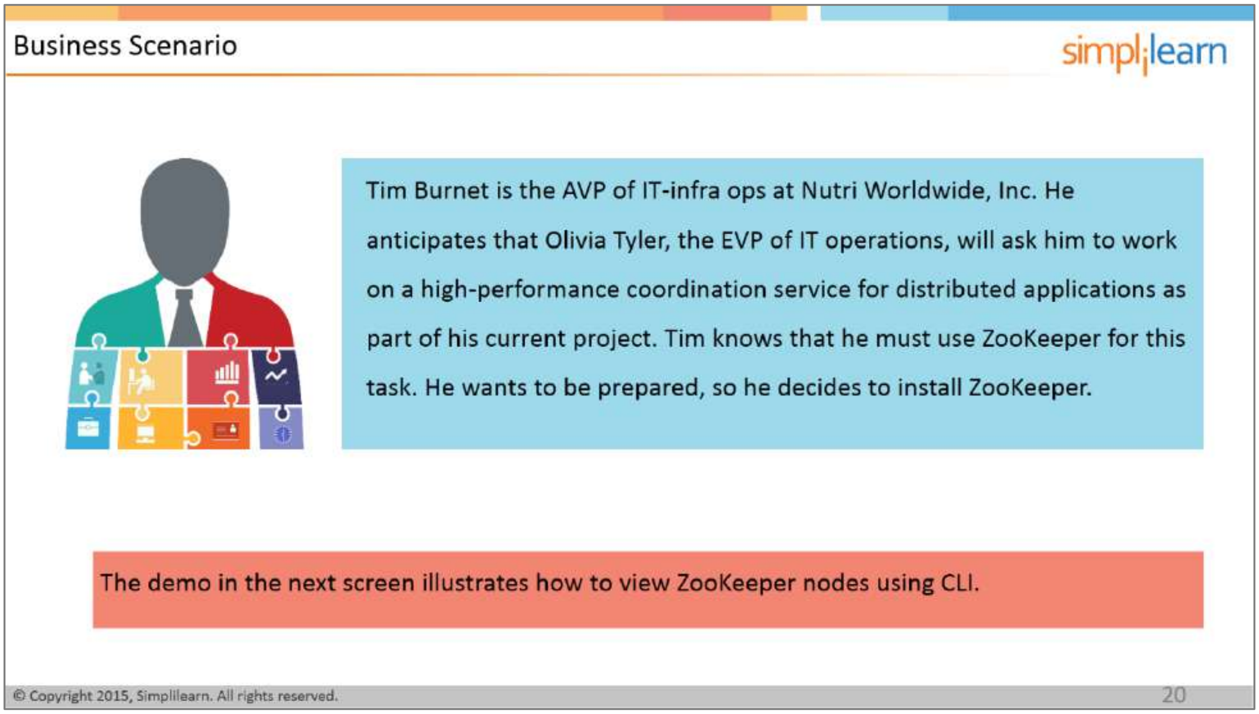
The given image illustrates this example in detail. Please spend some time to go through the image.



Recipe 3: Distributed Exclusive Lock

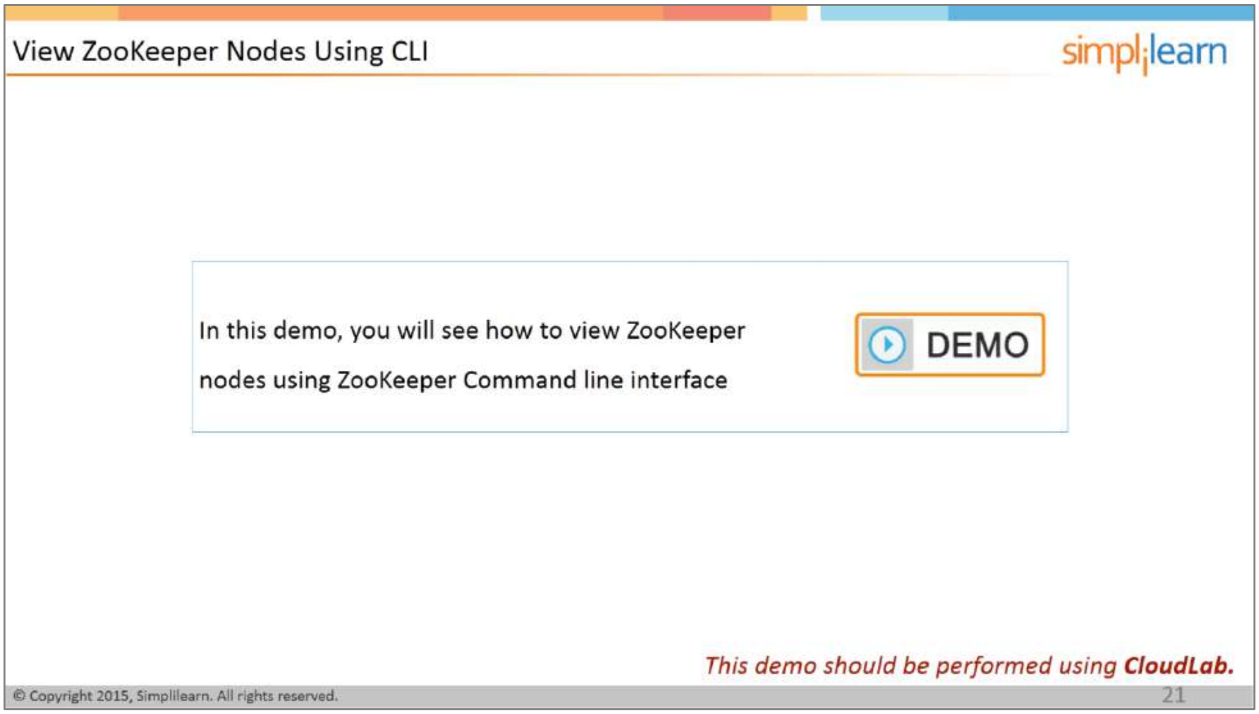
The following is a recipe for distributed exclusive lock function, assuming there are N web crawler clients trying to acquire a lock on links data.

- Clients create an ephemeral, sequential Znode under the path given.
- Clients request a list of children for the lock Znode, that is, locknode.
- The client with the lowest ID, according to natural ordering, will hold the lock; other clients set watches on the Znode with the ID immediately preceding its own identification. They periodically check for the lock in case of notification.
- The client wishing to release a lock deletes the node, which triggers the next client in line to acquire the lock.



Business Scenario

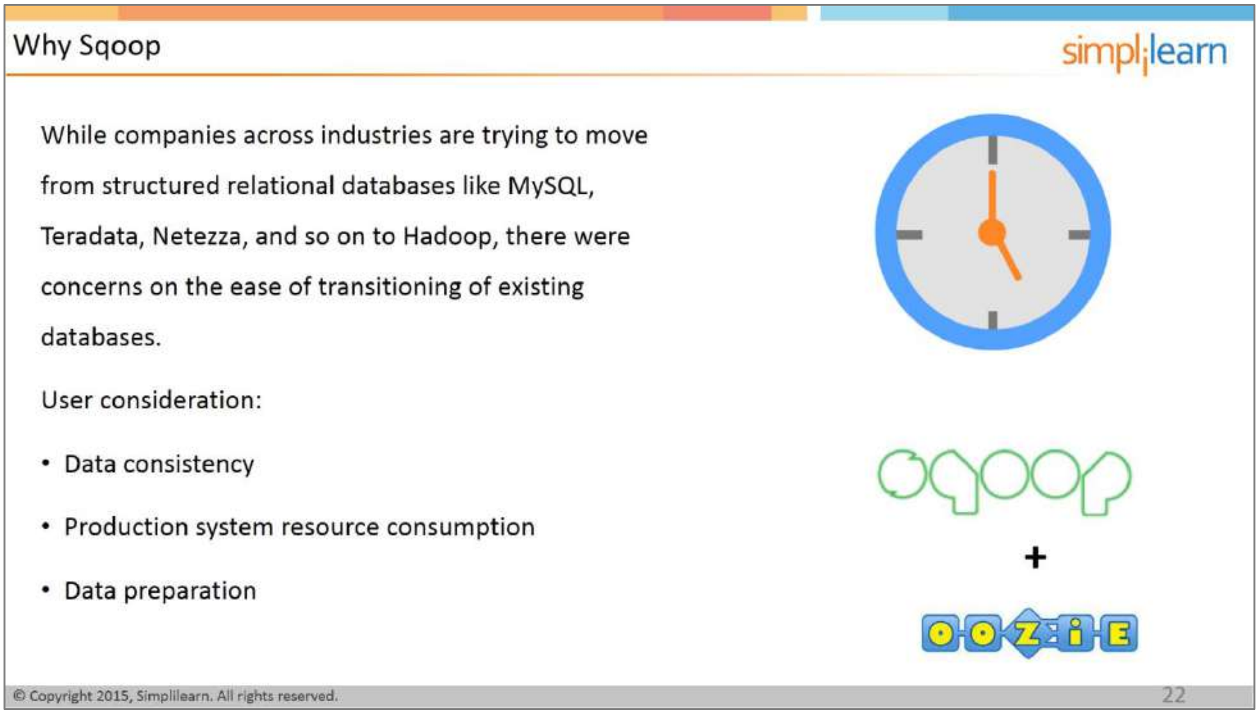
Tim Burnet is the AVP of IT-infra ops at Nutri Worldwide, Inc. He anticipates that Olivia Tyler, the EVP of IT operations, will ask him to work on a high-performance coordination service for distributed applications as part of his current project. Tim knows that he must use ZooKeeper for this task. He wants to be prepared, so he decides to install ZooKeeper.



View ZooKeeper Nodes Using CLI

Let us summarize the steps performed:

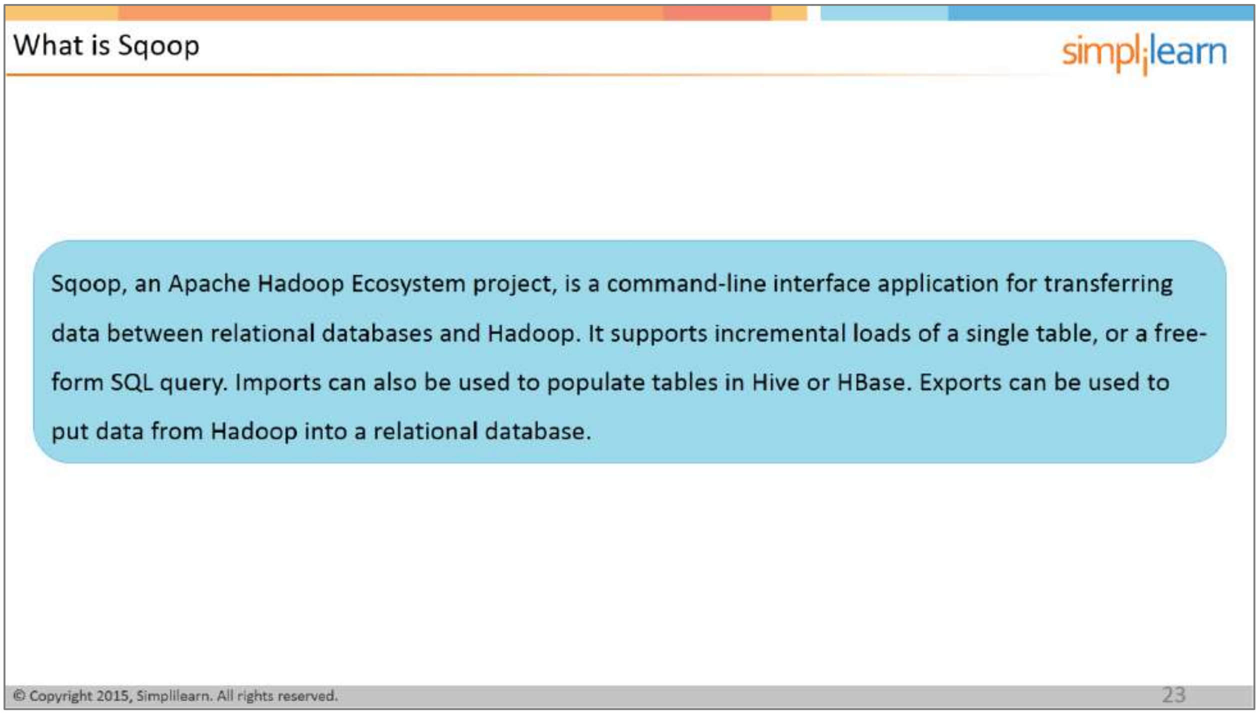
1. Use the Linux **find** command to find the Zookeeper CLI, that is, z-k-C-l-i,
2. CD to the directory containing z-k-C-l-i and launch it using local host IP address 127.0.0.1 and port 2-1-8-1.
3. Certain info messages appear.
   Enter the help command once the ZooKeeper prompt appears.
4. List the nodes using the l-s slash command.
5. List one of the nodes using 'l-s slash node name'.
6. You can also get information of the node using 'get slash hbase hyphen unsecure'.
7. Observe the listed attributes, like the number of children nodes.



Why Sqoop

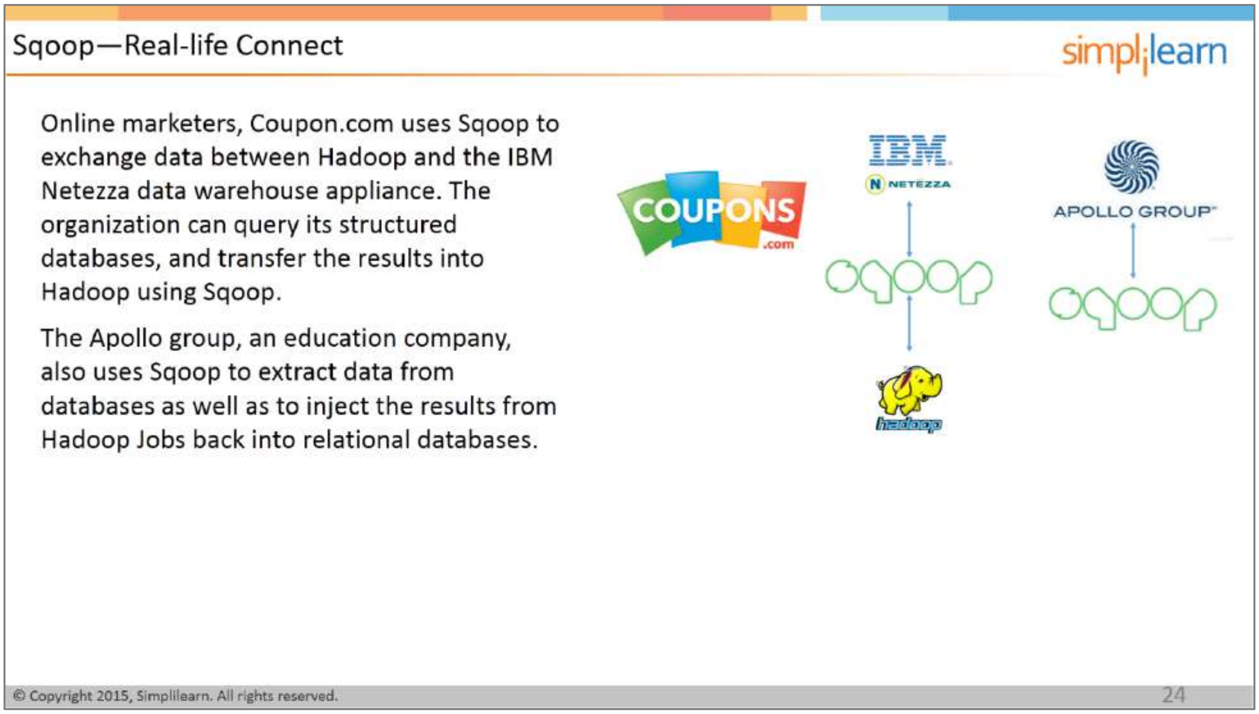
While companies across industries are trying to move from structured relational databases like MySQL, Teradata, Netezza, and so on to Hadoop, there were concerns on the ease of transitioning of existing databases. It was challenging to load bulk data into Hadoop, or accessing it from MapReduce. Users had to consider data consistency, production system resource consumption, and data preparation. Data transfer using scripts was both time consuming and inefficient. Direct access of data from external systems was also complicated.

This was solved with the introduction of Sqoop. Sqoop allows smooth import and export of data from structured databases. Along with Oozie, Sqoop helps in scheduling and automation of import and export tasks.



What is Sqoop

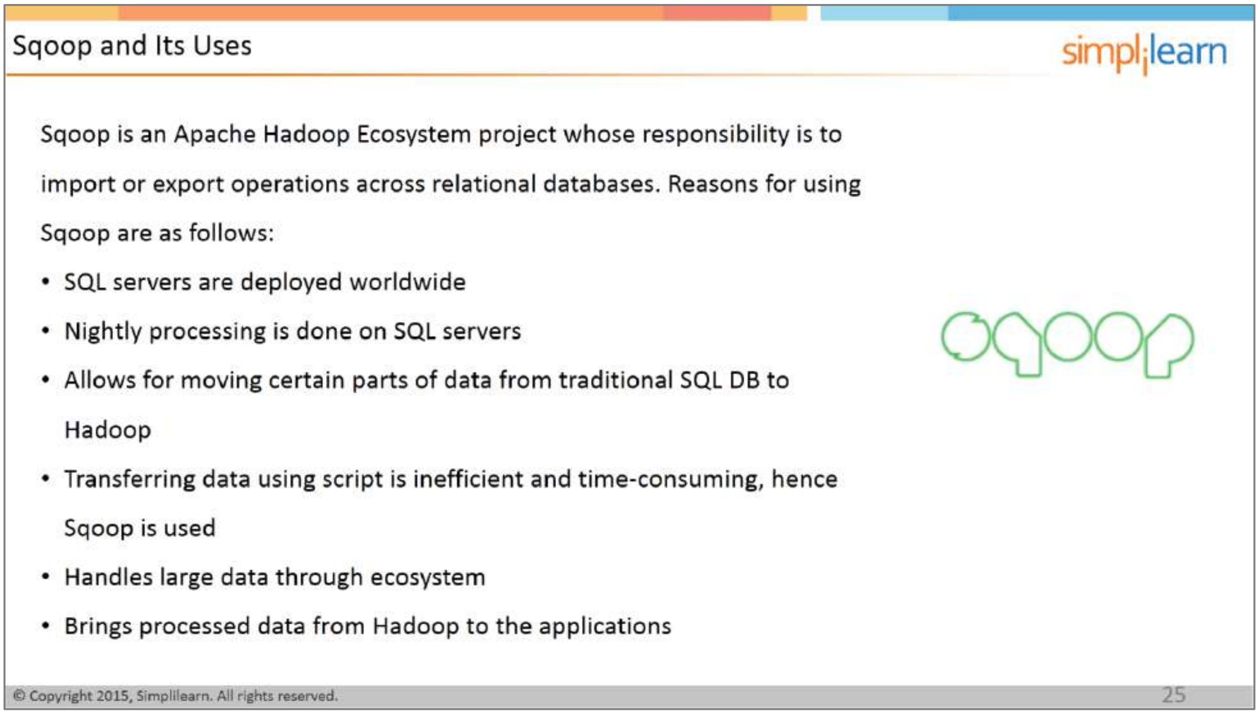
Sqoop, an Apache Hadoop Ecosystem project, is a command-line interface application for transferring data between relational databases and Hadoop. It supports incremental loads of a single table, or a free-form SQL query. Imports can also be used to populate tables in Hive or HBase. Exports can be used to put data from Hadoop into a relational database.



Sqoop—Real-life Connect

Online marketers, Coupon.com uses Sqoop to exchange data between Hadoop and the IBM Netezza data warehouse appliance. The organization can query its structured databases, and transfer the results into Hadoop using Sqoop.

The Apollo group, an education company, also uses Sqoop to extract data from databases as well as to inject the results from Hadoop Jobs back into relational databases.

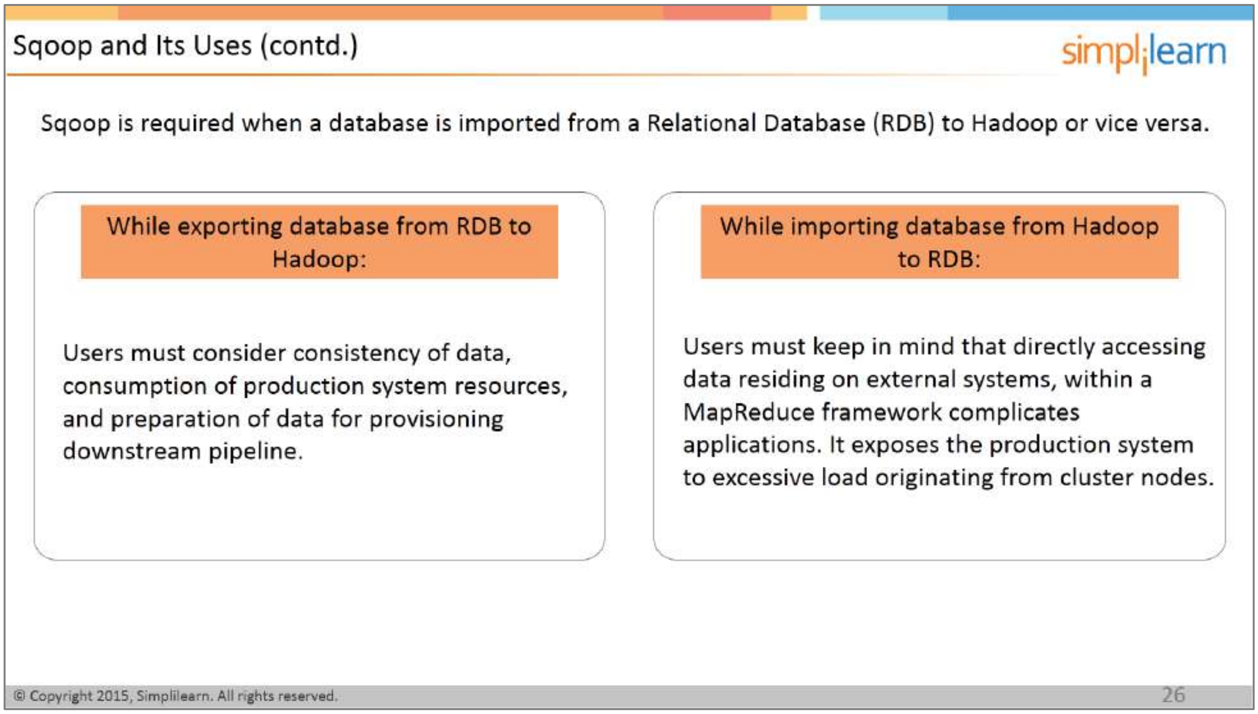


Sqoop and Its Uses

Sqoop is an Apache Hadoop Eco-system project; its responsibility is to import or export operations across relational databases like MySQL, MS SQL, and Oracle to HDFS.

Following are the various reasons for using Scoop:

SQL servers are deployed worldwide and are the primary ways to accept the data from a user. Nightly processing is being done on SQL servers for years. Sqoop allows for moving the data from traditional SQL DB to Hadoop HDFS as Hadoop makes its way into enterprises. Transferring the data using automated scripts is inefficient and time-consuming; hence Sqoop is used. Traditional DB has reporting, data visualization, and other enterprise built-in applications; however, to handle large data, you need an ecosystem. The need to bring the processed data from Hadoop HDFS to the applications like database engine or web services is satisfied by Sqoop.



Sqoop and Its Uses (contd.)

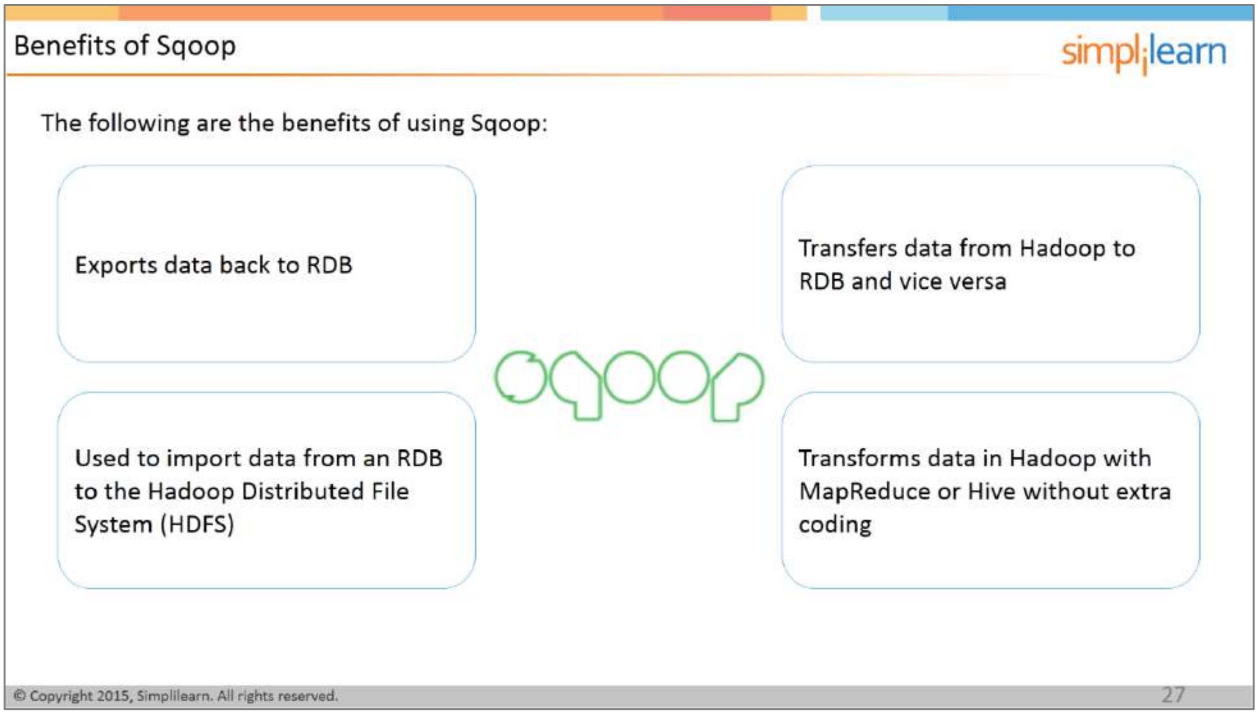
Sqoop is required when a database is imported from a Relational Database (RDB) to Hadoop or vice versa.

A Relational Database or RDB, refers to any data in a structured format. Databases in MySQL or Oracle are examples of RDB.

While exporting databases from a Relational Database to Hadoop, users must consider consistency of data, consumption of production system resources, and preparation of data for provisioning downstream pipeline.

While importing the database from Hadoop to a Relational Database, users must keep in mind that directly accessing data residing on external systems within a MapReduce framework complicates applications. It also exposes the production system to excessive loads originating from cluster nodes.

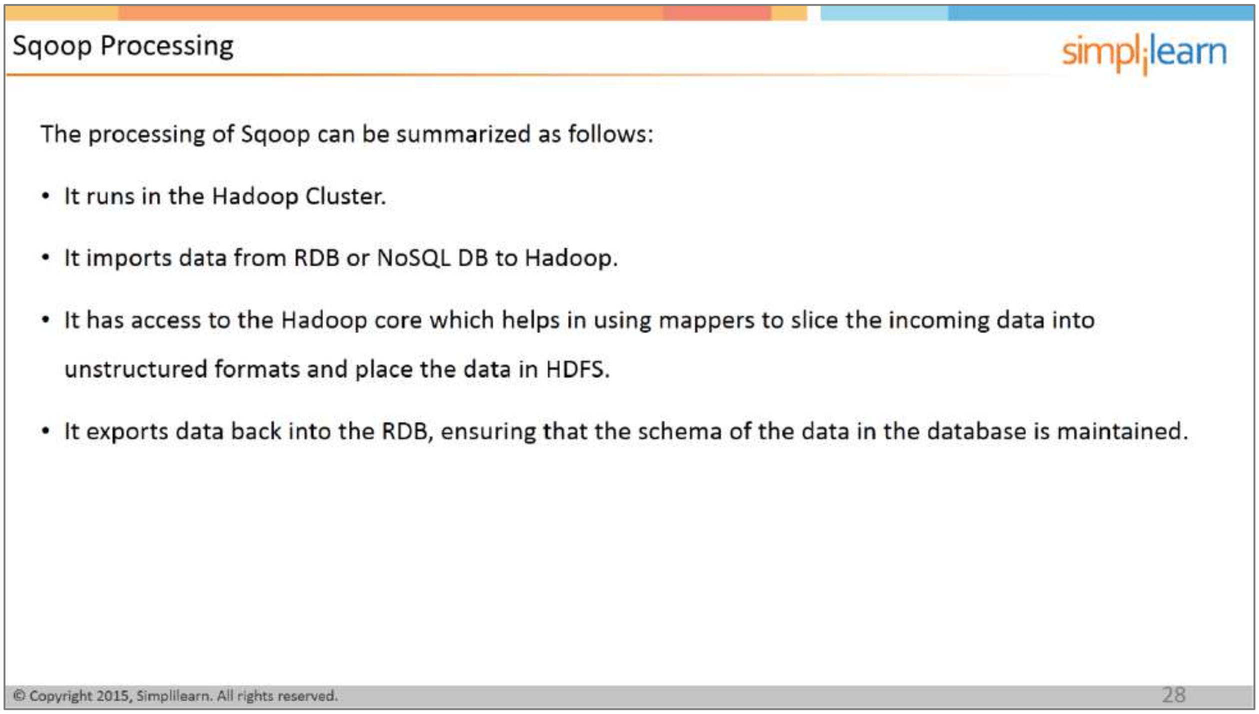
Hence, Sqoop is required in both the scenarios.



Benefits of Sqoop

Using Sqoop can benefit in the following ways:

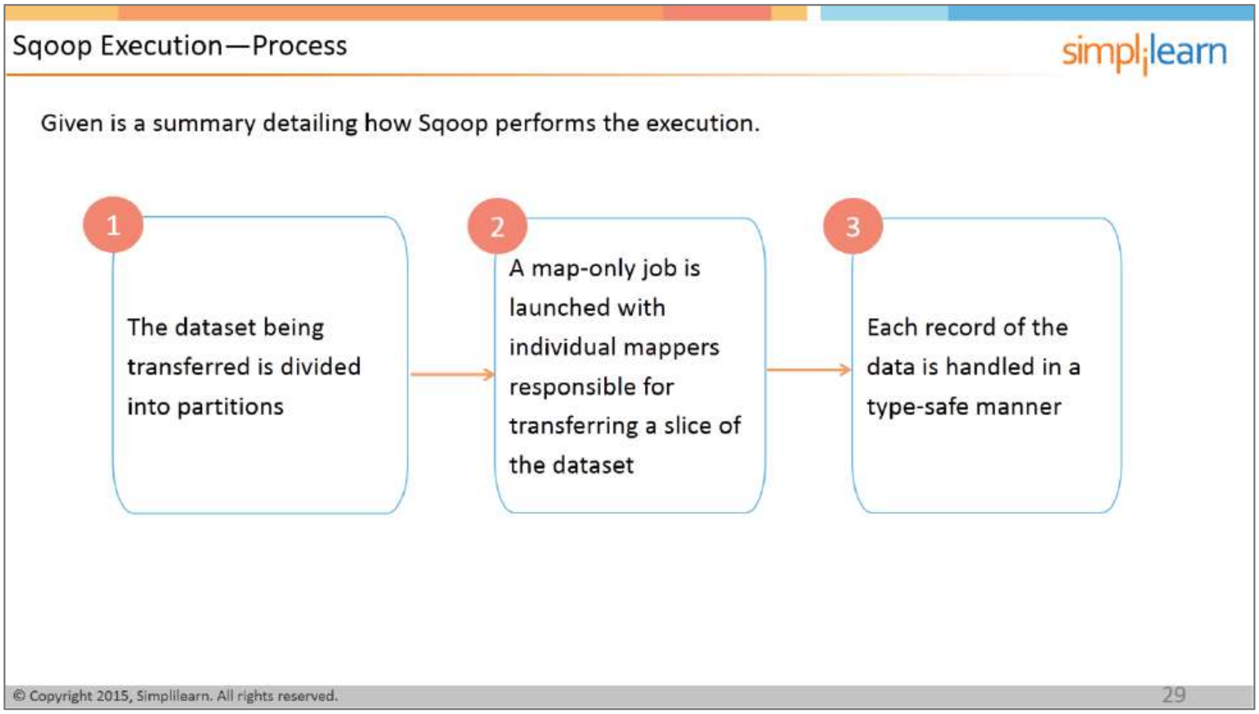
- It transfers data from Hadoop to an RDB and vice versa.
- It transforms data in Hadoop with the help of MapReduce or Hive without extra coding.
- It is used to import data from an RDB, such as SQL, MySQL, or Oracle into the Hadoop Distributed File System or HDFS.
- Sqoop exports data back to the RDB.



Sqoop Processing

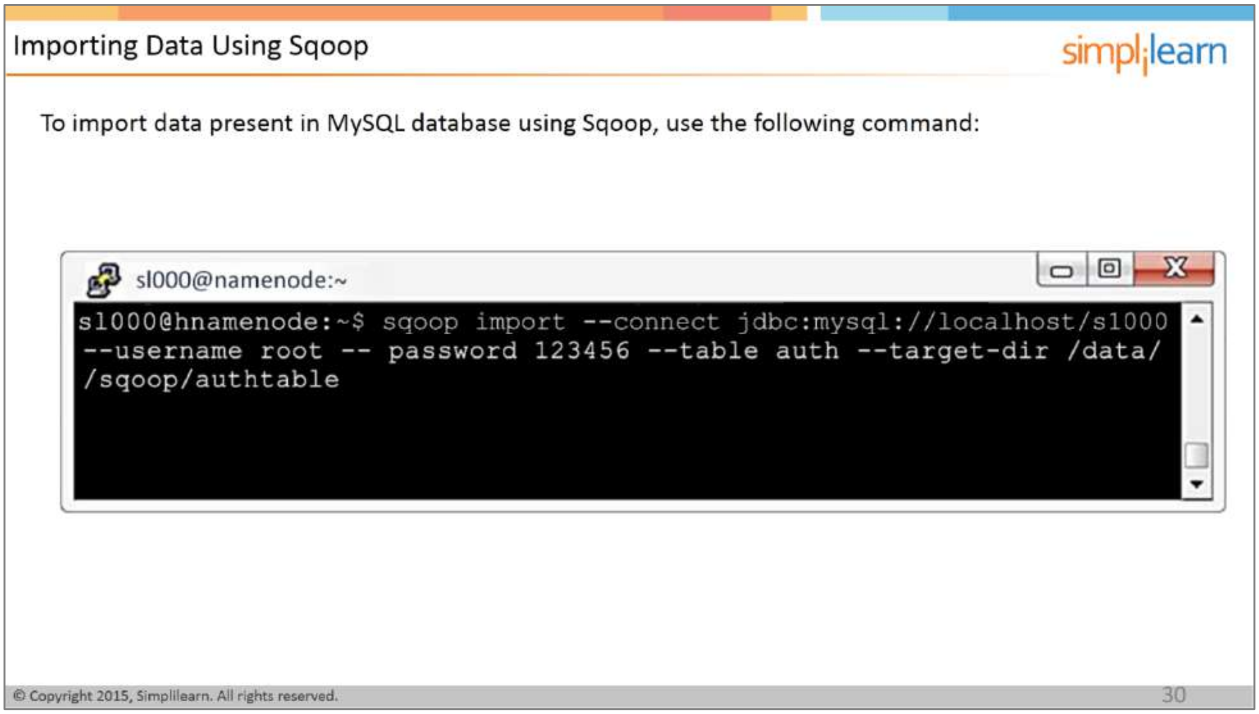
The following is a summary of processing of Sqoop.

Sqoop runs in a Hadoop Cluster. It imports data from the RDB or NoSQL DB to Hadoop. It has access to the Hadoop core which helps in using mappers to slice the incoming data into unstructured formats and place the data in HDFS. It exports data back into RDB, ensuring that the schema of the data in the database is maintained.



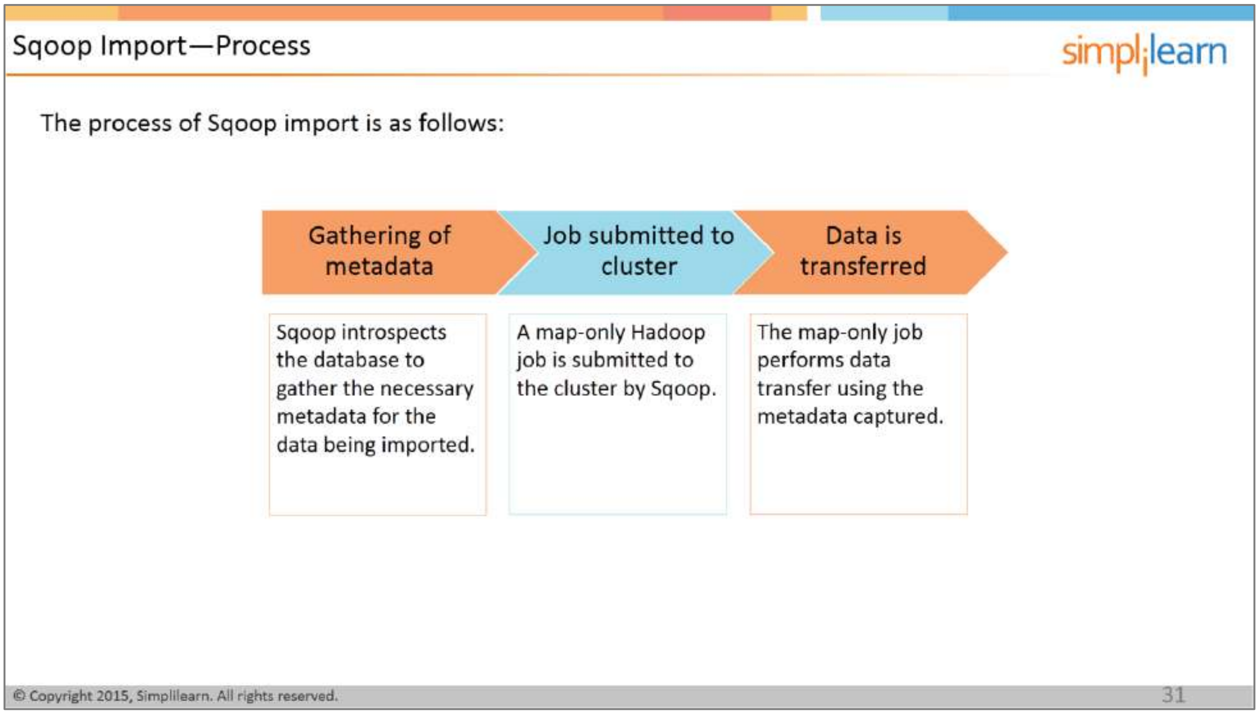
Sqoop Execution—Process

This describes how Sqoop performs its execution. First, the dataset being transferred is divided into partitions. Next, a map-only job is launched with individual mappers responsible for transferring a slice of the dataset. Lastly, each record of the data is handled in a type-safe manner as Sqoop uses metadata to infer the data types.



Importing Data Using Sqoop

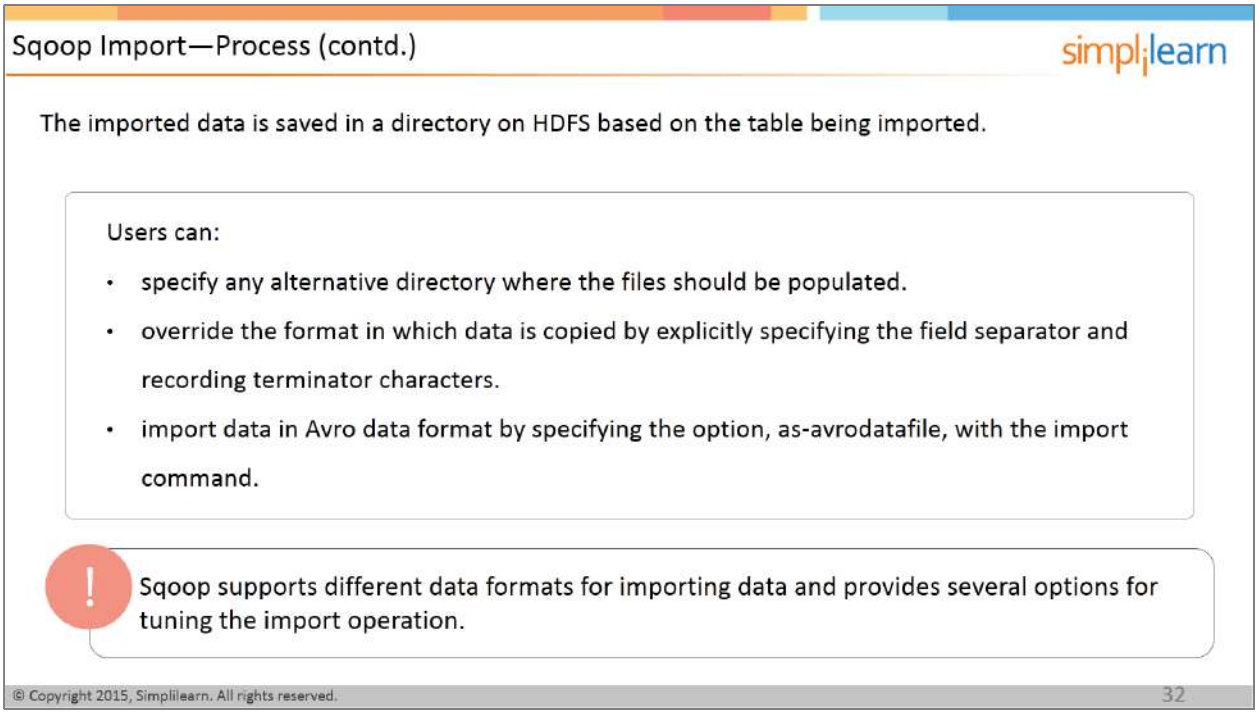
Use the command shown on the image to import data present in MySQL database using Sqoop where sl000 is the database name, and auth is the table name.



Sqoop Import—Process

The process of the Sqoop import is summarized:

- Sqoop introspects the database to gather the necessary metadata for the data being imported.
- A map-only Hadoop job is submitted to the cluster by Sqoop.
- The map-only job performs data transfer using the metadata captured in step one.

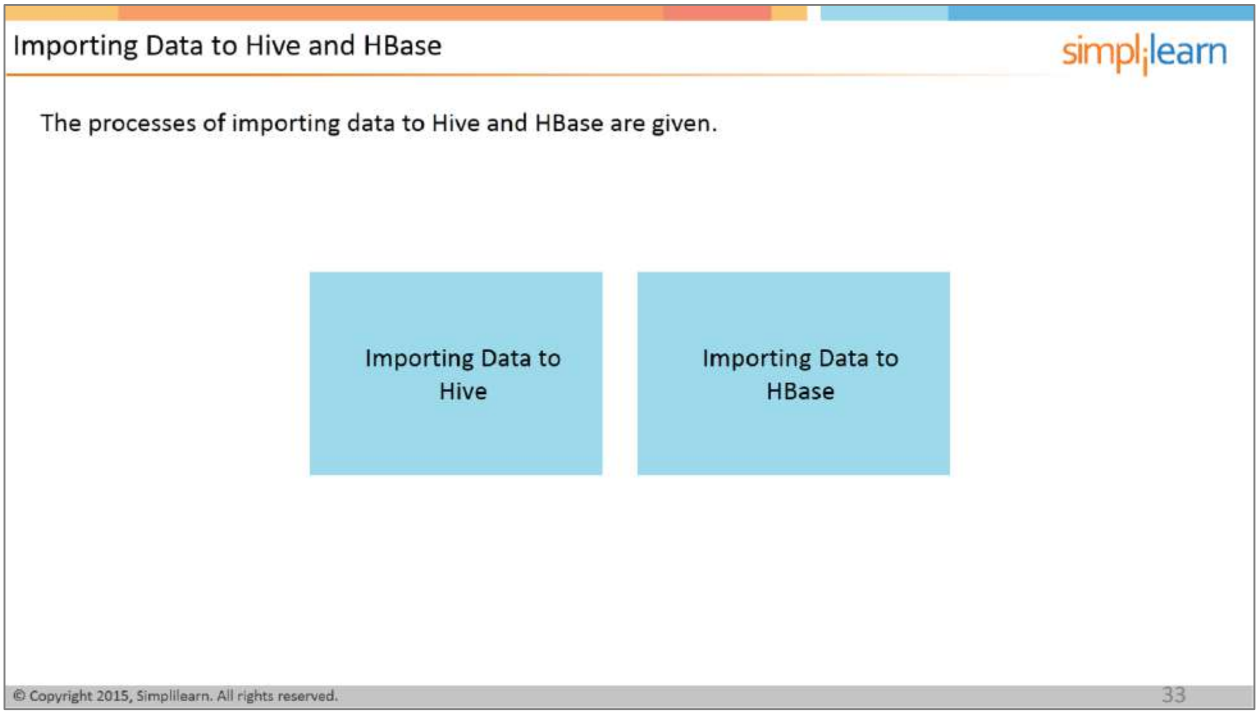


Sqoop Import—Process (contd.)

The imported data is saved in a directory on HDFS based on the table being imported.

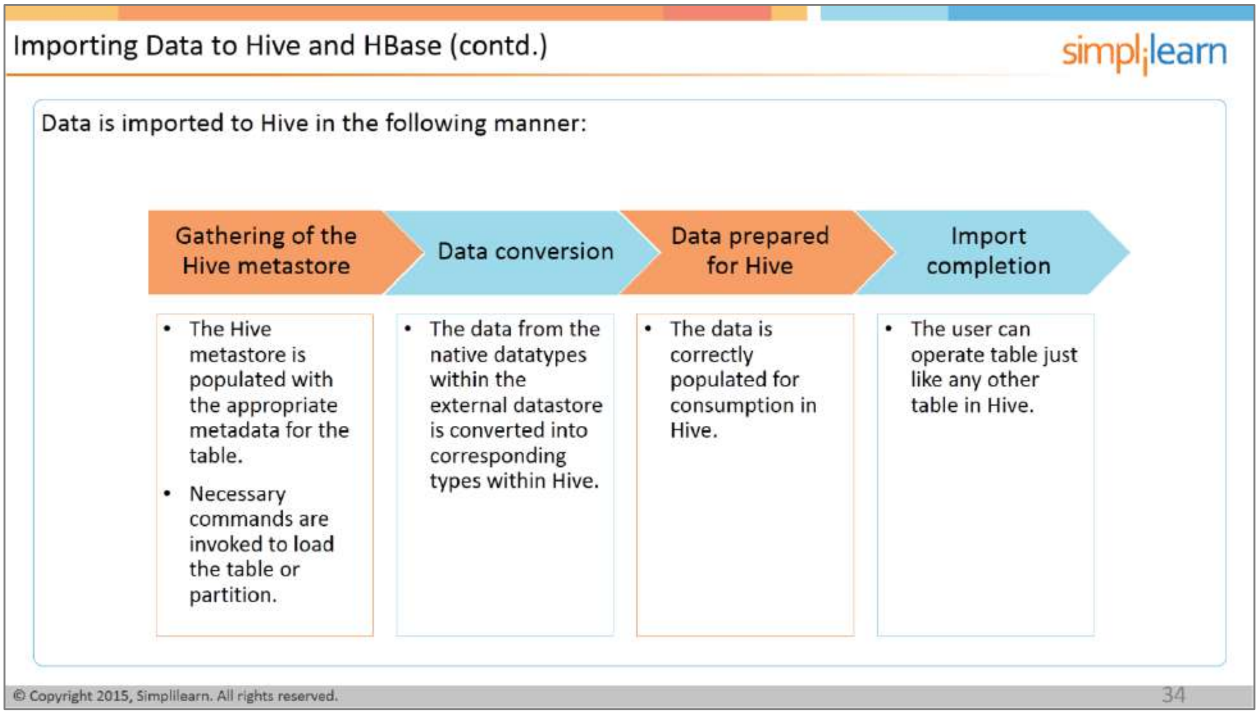
Users can specify any alternative directory where the files should be populated. By default, these files contain comma delimited fields with new lines separating the different records. Users can also override the format in which data is copied by explicitly specifying the field separator and recording terminator characters. Further, users can easily import data in Avro data format by specifying the option '-as-avrodatafile' with the import command.

Sqoop supports different data formats for importing data. It also provides several options for tuning the import operation.



Importing Data to Hive and HBase

The processes of importing data to Hive and HBase are given.



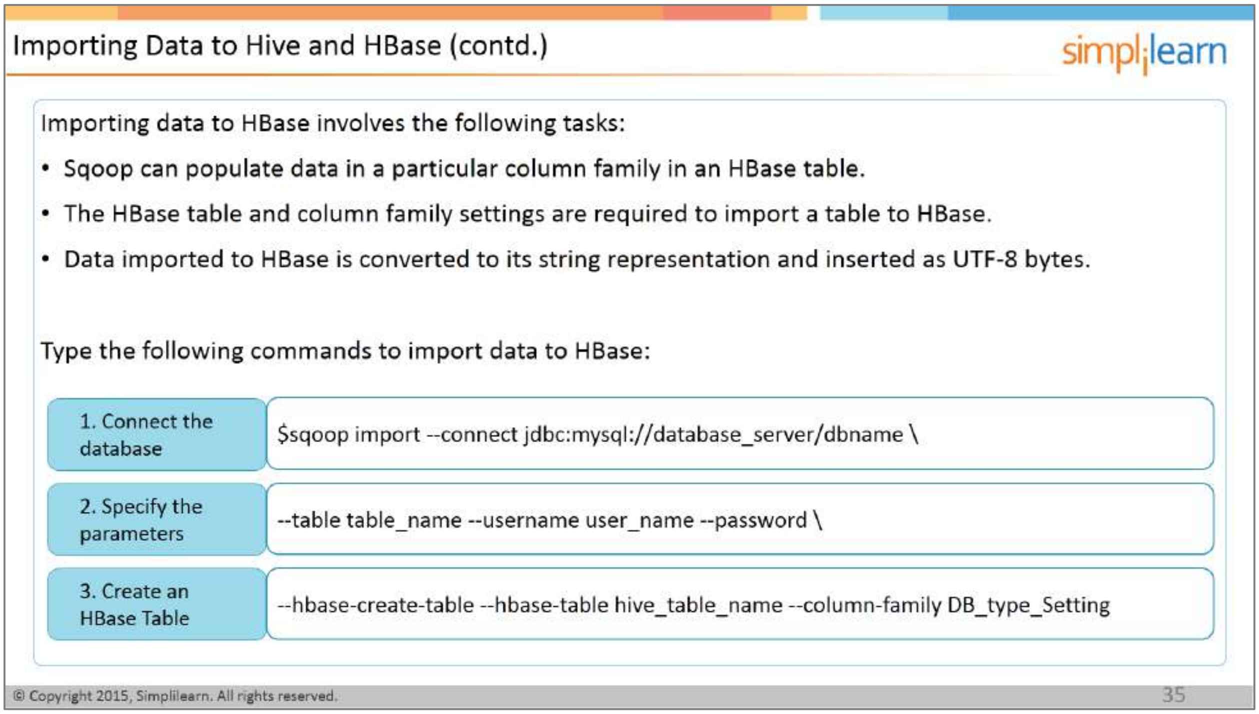
Importing Data to Hive and HBase (contd.)

First, Sqoop takes care of populating the Hive metastore with appropriate metadata for the table, and also invokes the necessary commands to load the table or partition.

Next, using Hive import, Sqoop converts the data from the native datatypes in the external datastore into the corresponding types within Hive.

Further, Sqoop automatically chooses the native delimiter set used by Hive. If the data being imported has new line or other Hive delimiter characters in it, Sqoop allows the removal of such characters. The data is then correctly populated for consumption in Hive.

Lastly, after the import is completed, the user can operate on table just like any other table in Hive.

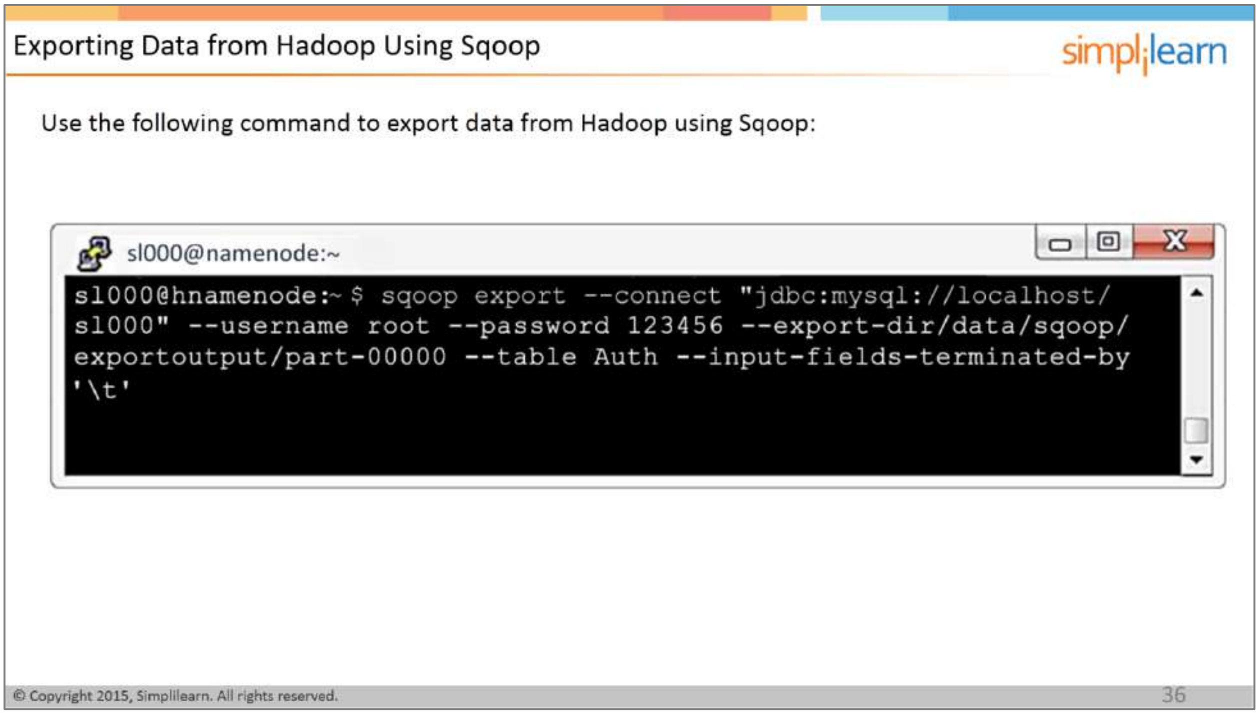


Importing data to HBase (contd.)

When data is imported to HBase, Sqoop can populate the data in a particular column family in an HBase table. The HBase table and the column family settings are required to import a table to HBase. Data imported to HBase is converted to its string representation and inserted as UTF-8 bytes.

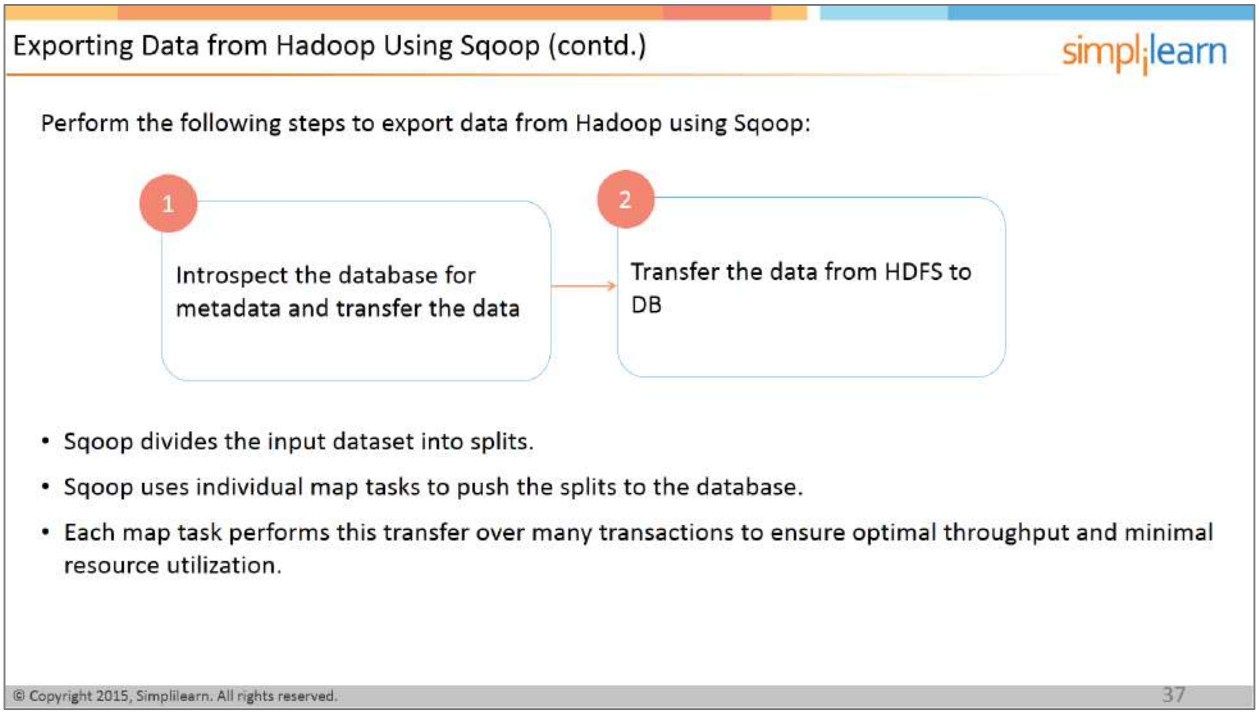
Use the commands shown to import data to HBase.

- Connect to the database using the first command.
- Specify the parameters such as username, password, and table-name using the second command.
- Create an HBase table with the column family as specified in MySQL using the third command.



Exporting Data from Hadoop Using Sqoop

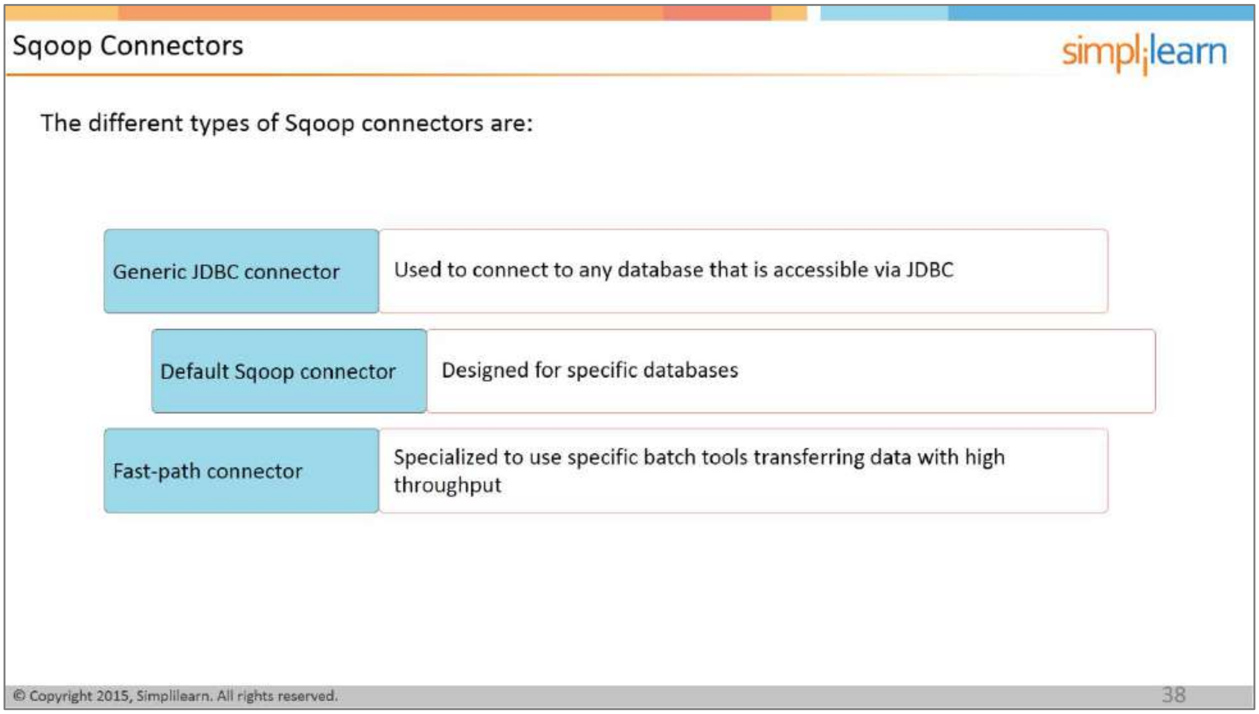
Use the command shown on the image to export data from Hadoop using Sqoop.



Exporting Data from Hadoop Using Sqoop

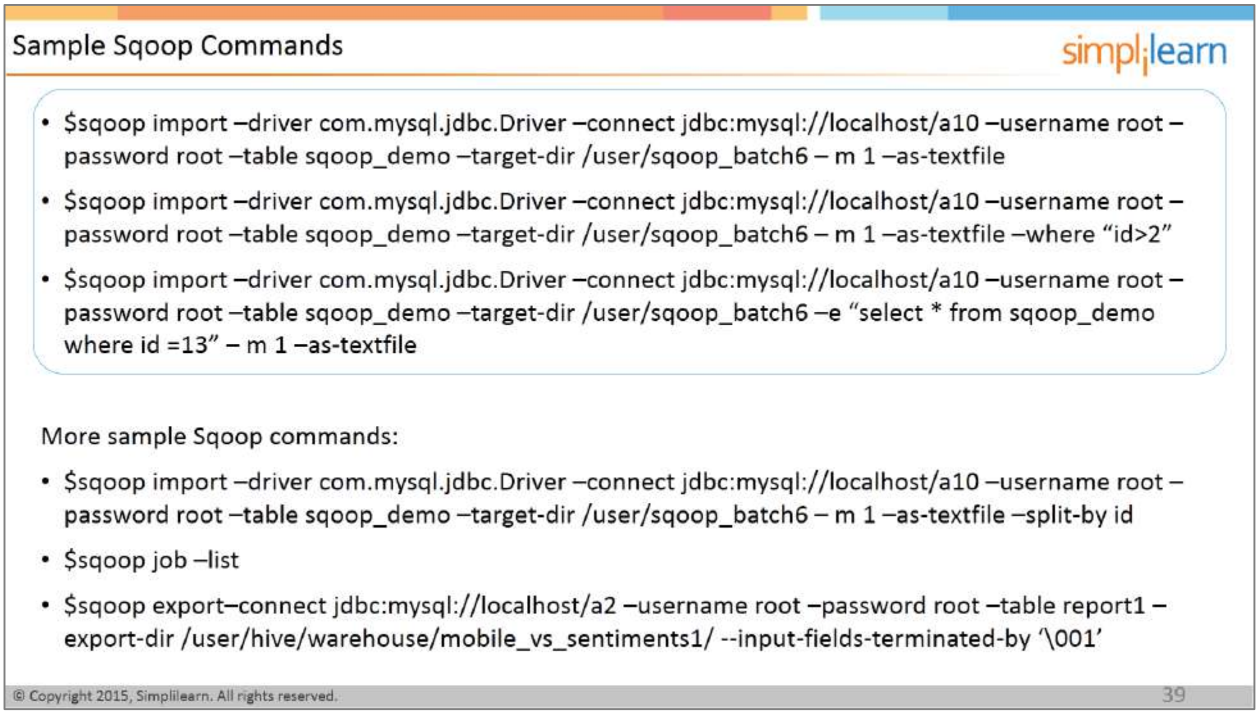
There are a number of steps to export data from Hadoop using Sqoop. First, introspect the database for metadata and transfer the data. Next, transfer the data from HDFS to the DB.

Further, Sqoop divides the input dataset into splits. Sqoop uses individual map tasks to push the splits to the database. Each map task performs this transfer over many transactions to ensure optimal throughput and minimal resource utilization.



Sqoop Connectors

The different types of Sqoop connectors are Generic JDBC, Default Sqoop, and Fast-path connectors. The Generic JDBC connector can be used to connect to any database that is accessible via JDBC. The Default Sqoop connector is designed for specific databases such as MySQL, PostgreSQL, Oracle, SQL Server, and DB2. The Fast-path connector specializes in using specific batch tools to transfer data with high throughput. For example, MySQL and PostgreSQL databases.



Sample Sqoop Commands

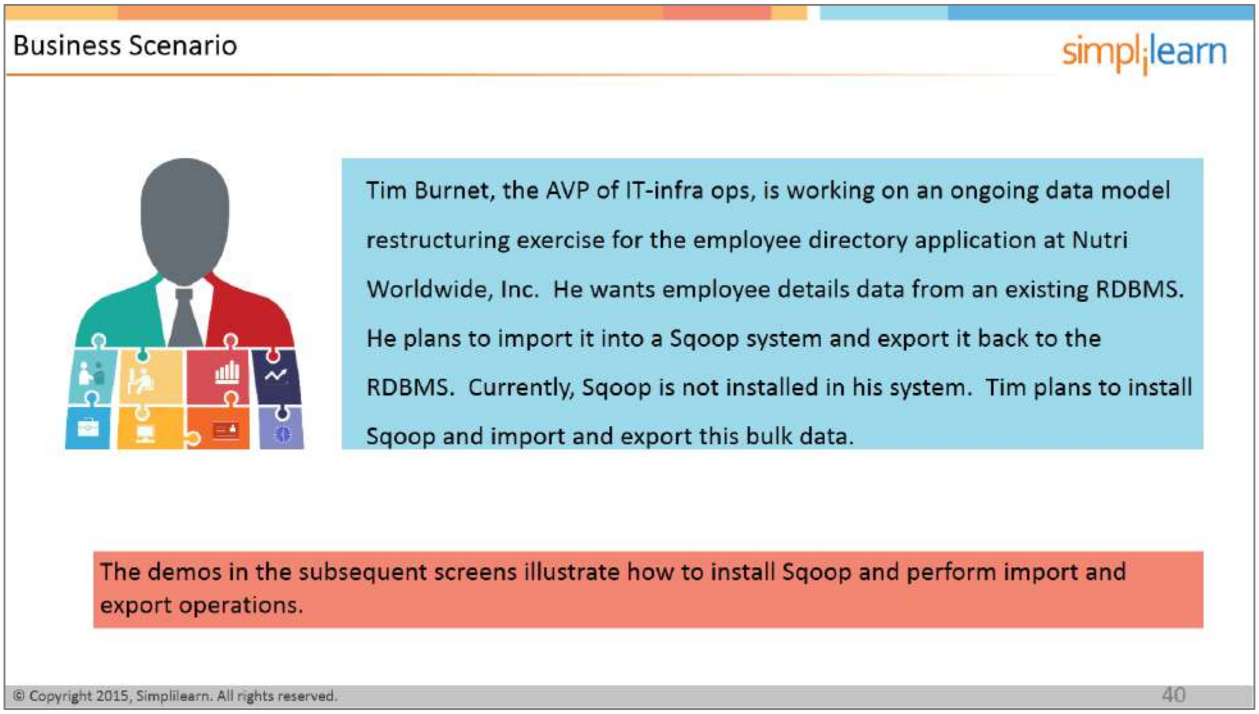
The various common Sqoop commands are listed.

The first command is for importing the data from the mysql table sqoop demo to an HDFS directory. Note the –m 1 which ensures there is only one mapper output.

In the second command, note that id is greater than two, which places a condition on data to be imported. You can also specify a specific sql query as shown in third command, where it mentions –e select start from sqoop underscore demo where id equals to thirteen.

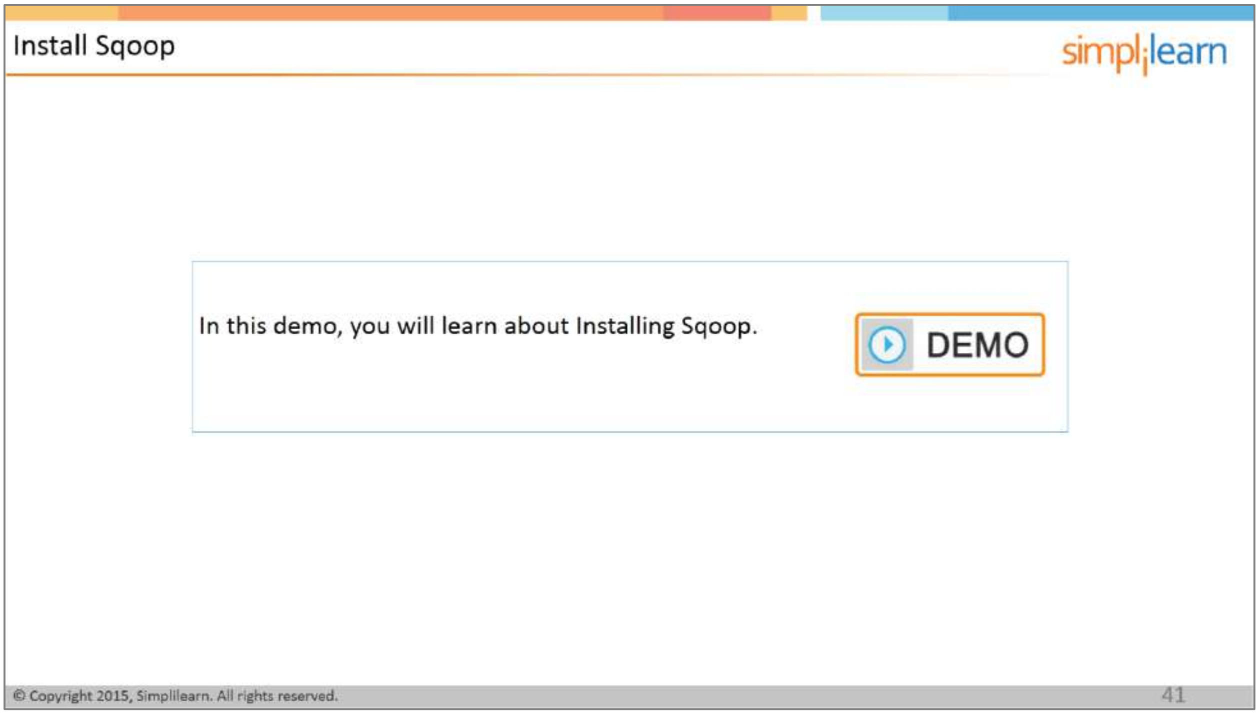
The third command shows an export function. Please note the hyphens and the double hyphen before driver, connect, username, and password.

There are some more commands listed for your reference. Please go through the same for better understanding.



Business Scenario

Tim Burnet, the AVP of IT-infra ops at Nutri Worldwide, Inc., is working on an ongoing data model restructuring exercise for the employee directory application. He wants employee details data from an existing RDBMS. He plans to import the data into a Sqoop system and export it back to the RDBMS. Currently, Sqoop is not installed in his system. Tim plans to install Sqoop and import and export this bulk data.

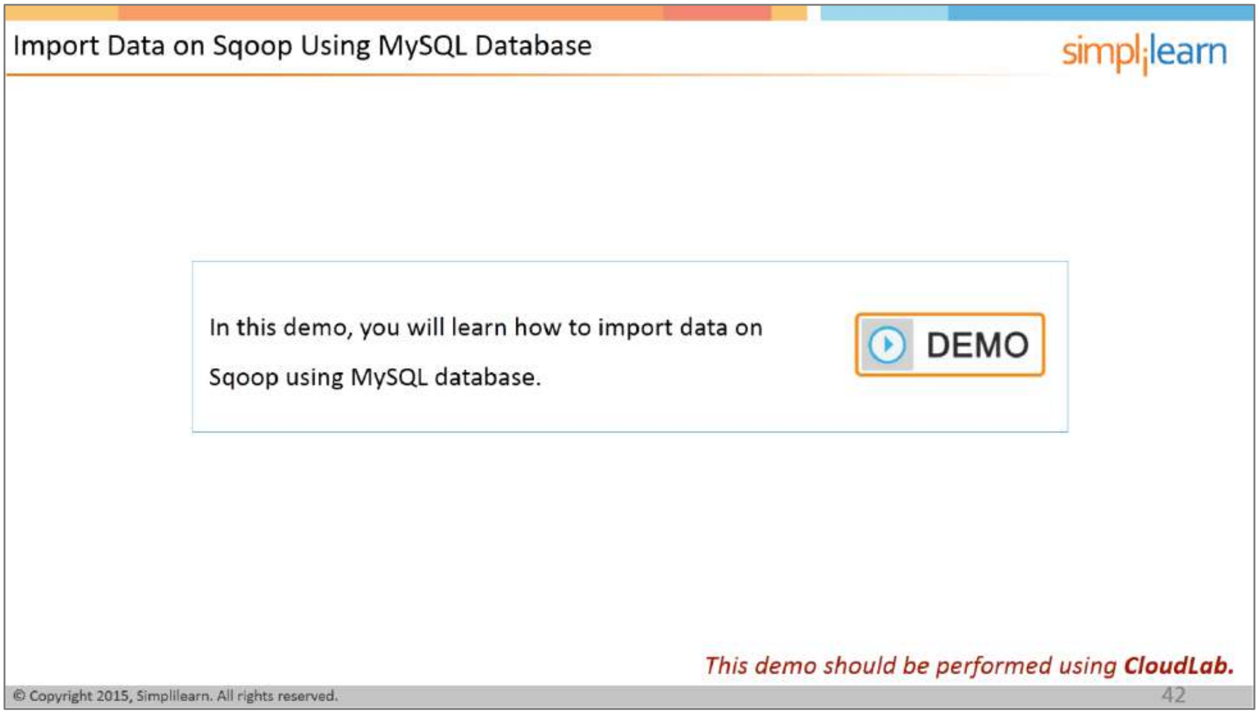


Install Sqoop

Let us summarize the steps performed:

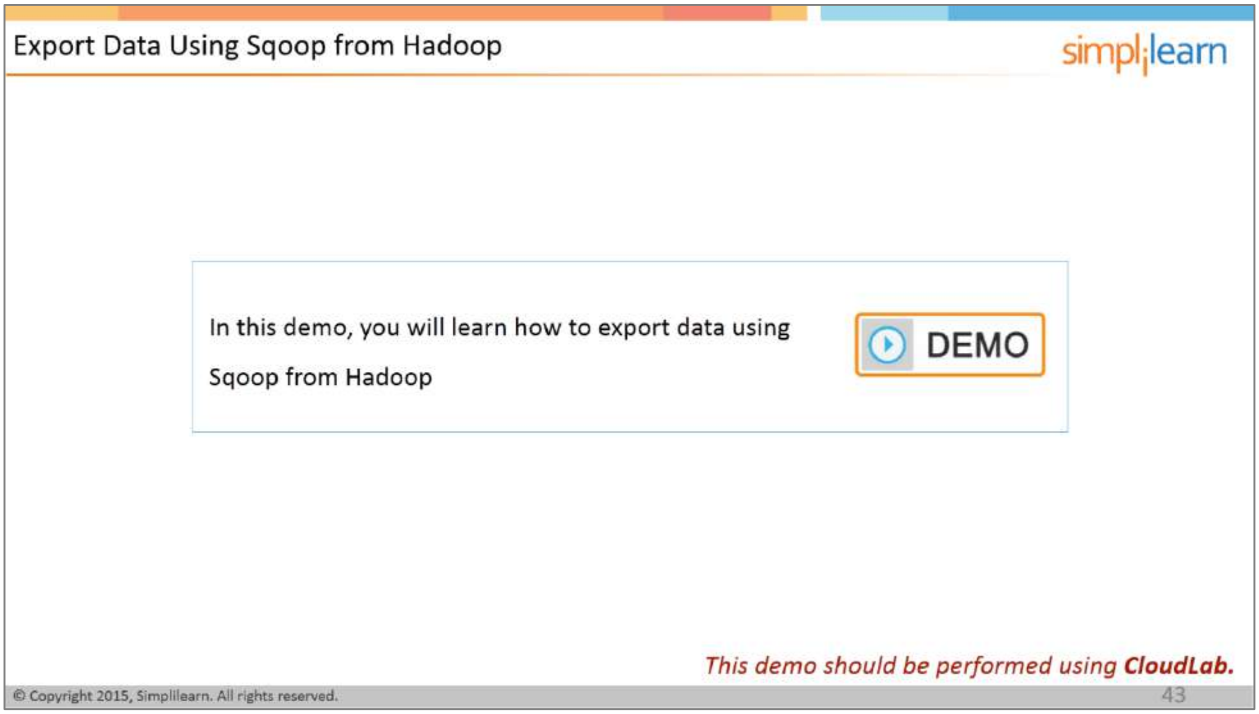
1. Visit the website sqoop.apache.org
2. Click the nearby mirror link.
3. Select a mirror link to download Pig.
4. Ensure you select the version that supports hadoop 1.0.
5. Go to Ubuntu server and type **wget** followed by the copied link to download sqoop.
6. Enter the file to extract it in the folder. Type tar -xvzf sqoop.
7. Copy the folder in the location, /usr/local/sqoop.
8. Type sudo cp –r sq.
9. Setup bashrc. Type sudo vi $HOME/ .bashrc.
10. Set the highlighted part to ensure sqoop has all the dependencies required.
11. Update the bash prompt.
12. To perform a demo on sqoop, you need a database server.
13. Create a database named SL.
14. Type the command **use sl** and press **Enter** so that MySQL will use this as the main database.
15. Create a table called authentication having columns called username and password as headings.
16. Insert data in the table.
17. Download the database driver that will be used by sqoop by visiting
www.mysql.com/downloads/
18. Download **MySQL Connectors**, click the **Download** link.
19. Select **Connector/J** to download your required distribution.

20. Select the **Platform Independent** option from the Select Platform drop-down list.
21. Click the **Download** button next to Platform-Independent, Architecture-Independent, Compressed TAR Archive connector.
22. Ensure that the downloaded file is copied in **/usr/local/sqoop/lib.**
23. Change the ownership for **/usr/local/sqoop/lib.**
24. You have successfully installed and configured sqoop.



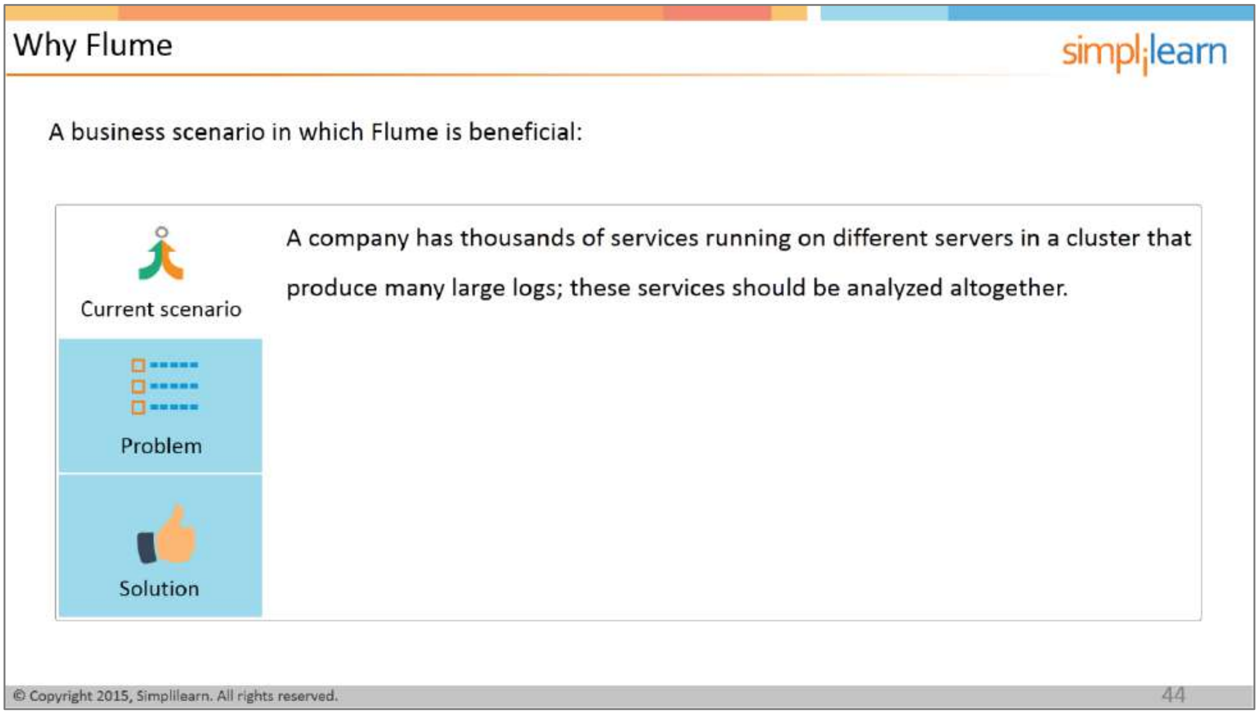
Import Data on Sqoop Using MySQL Database

1. List the databases in MySQL using the 'show databases' command.
2. Use a test database in this case.
3. Create a table named my sql data with the id as integer primary key and name as varchar 50.
4. Once the table is created, insert a row in the table with a value like 1, USA.
5. To verify whether the row has been inserted successfully, list the contents of the table using the select query.
6. View the contents of the table.
7. Run the sqoop utility.
8. Enter the sqoop command as shown and execute it.
9. You can see some log messages.
10. In the background, sqoop invokes MapReduce programs. Finally, you will get a confirmation and some job metrics.
11. Check the data that has been imported in HDFS.
12. Observe that two or more files are created.
13. Cat the contents of the part outputs, once for each.
14. Now, run sqoop with one mapper. Execute the command with a –m 1.
15. List the contents of the HDFS directory.
16. You will observe only one output file as you ran only one mapper.
17. Cat the contents to view the data.
18. You will observe that all the records have been imported successfully in one file in HDFS.



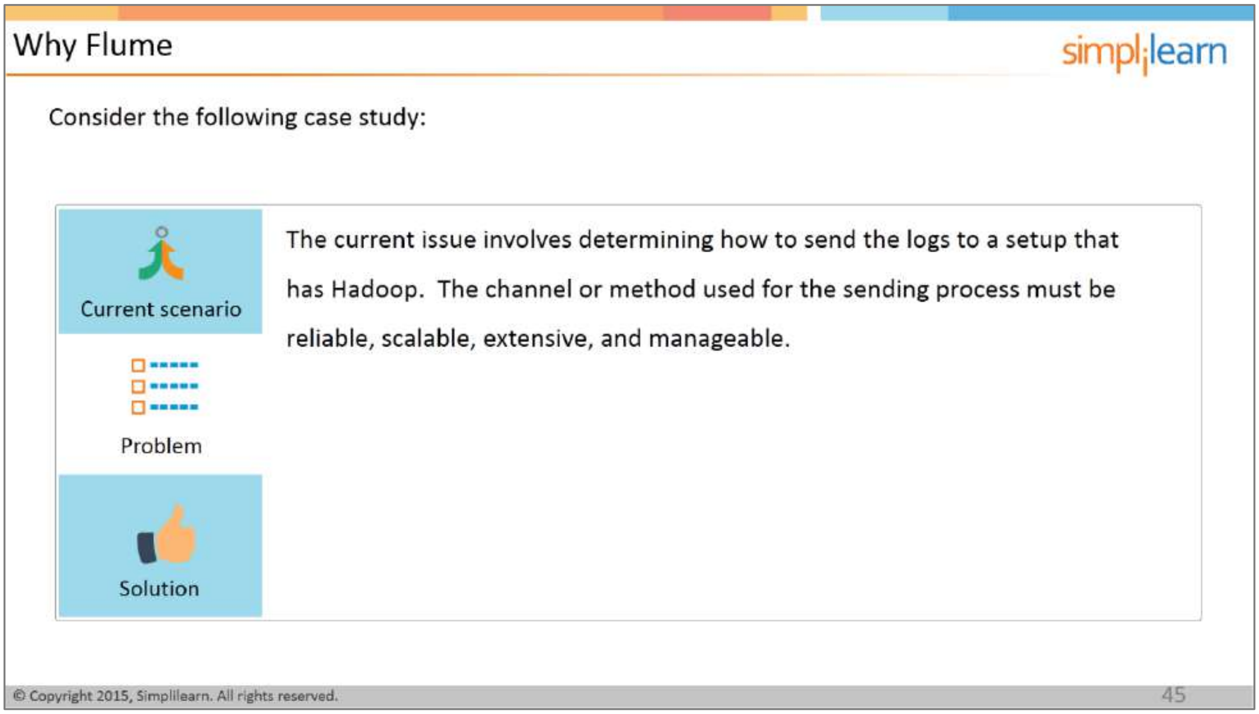
Export Data Using Sqoop from Hadoop

1. Let us now export data from HDFS to MySQL database using Sqoop.
2. Launch MySQL by typing mysql on command prompt.
3. Type use test to change the database.
4. Create a table named ‘ exported_table’ with ID as primary key integer and name as Variable character.
5. Select star from exported_ table to check the existing data.
6. You will see zero results indicating table has no data.
7. You can list the contents of the file using the command shown.
8. You will notice the datasets contain 2 rows. First row is laptop and the second row is desktop.
9. Copy the file to HDFS using the command shown.
10. Let us now use the Sqoop export command to export the data from HDFS to MySQL database.
11. Enter the JDBC URL of MySQL database. Also mention the table in which data has to be exported.
12. Mention the HDFS file or directory from which data has to be exported MySQL table and press Enter.
13. You will notice the MapReduce percentage.
14. You will see Map is 100 percent while Reduce is 0 percent. This indicates that Reducers are not running as a part of Sqoop job.
15. Launch MySQL again.
16. Type use test to change the database.
17. Select star from exported_ table to check the existing data.
18. You will notice the 2 records have been exported from HDFS to MySQL database.



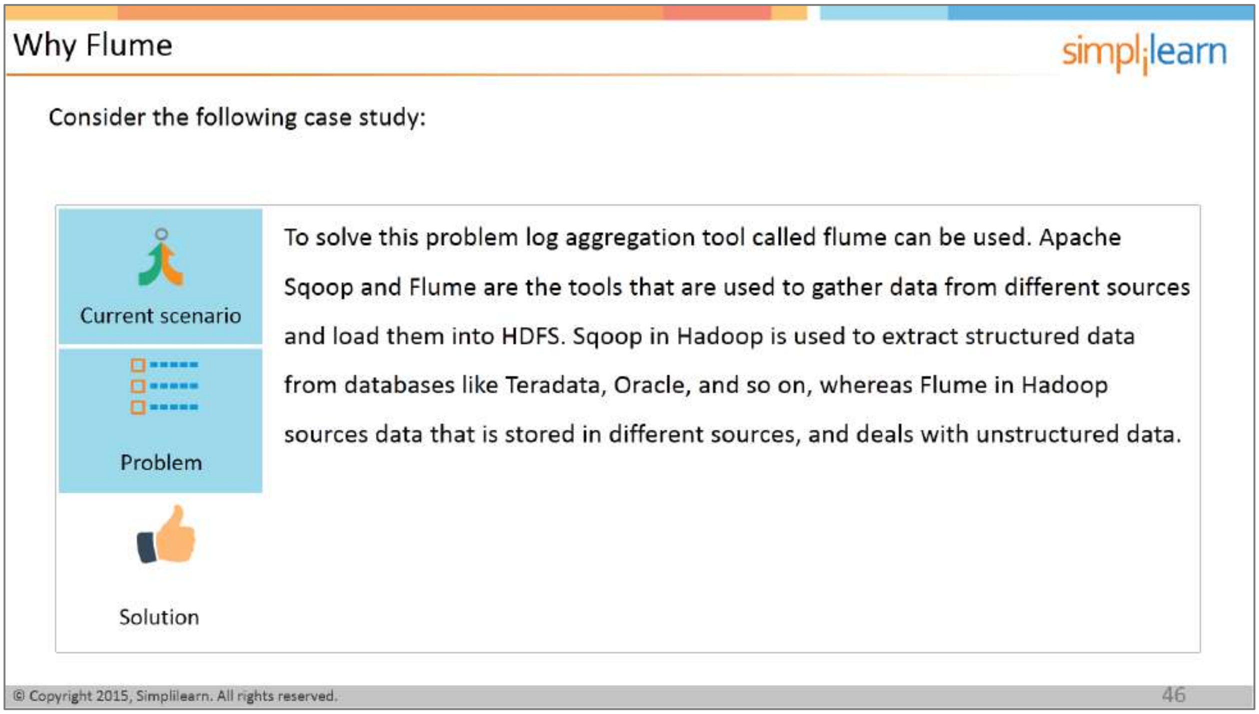
Why Flume

Consider a scenario where a company has thousands of services running on different servers in a cluster that produce many large logs. These logs should be analyzed together.



Why Flume

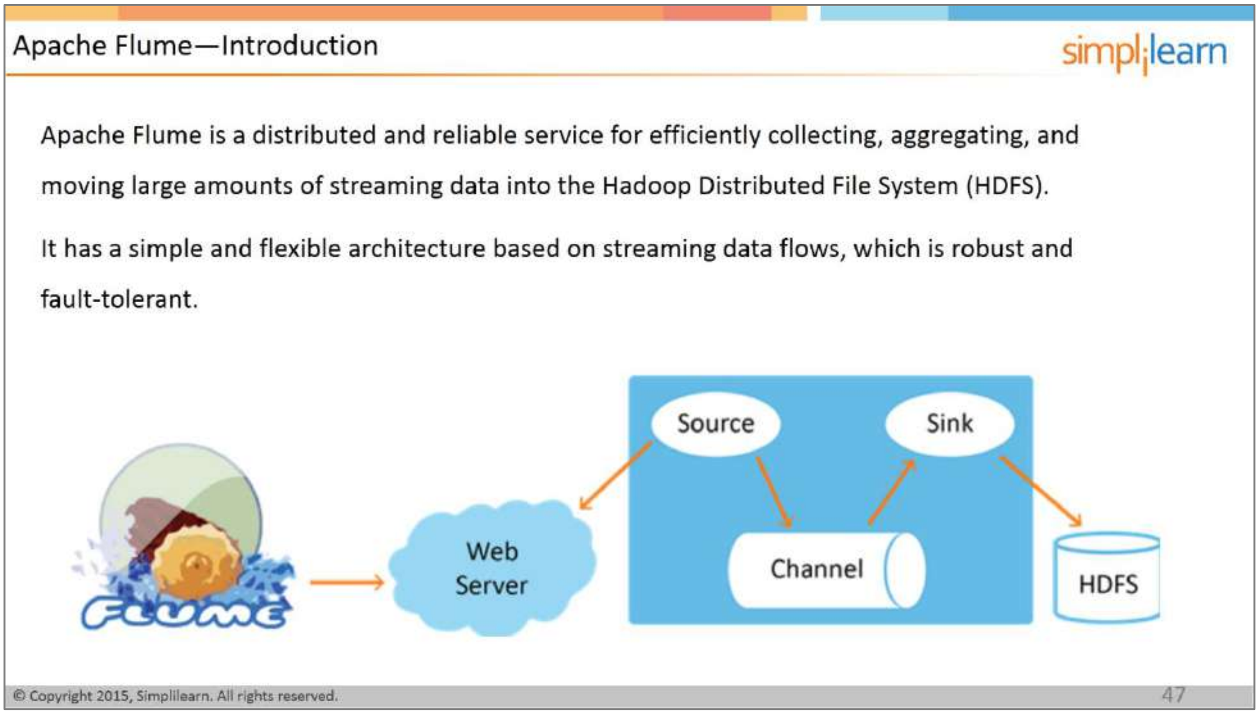
The current issue involves determining how to send the logs to a setup that has Hadoop. The channel or method used for sending them must be reliable, scalable, extensive, and manageable.



Why Flume

To solve this problem log aggregation tool called flume can be used.

Apache Sqoop and Flume are the tools that are used to gather data from different sources and load them into HDFS. Sqoop in Hadoop is used to extract structured data from databases like Teradata, Oracle, and so on, whereas Flume in Hadoop sources data that is stored in different sources, and deals with unstructured data.



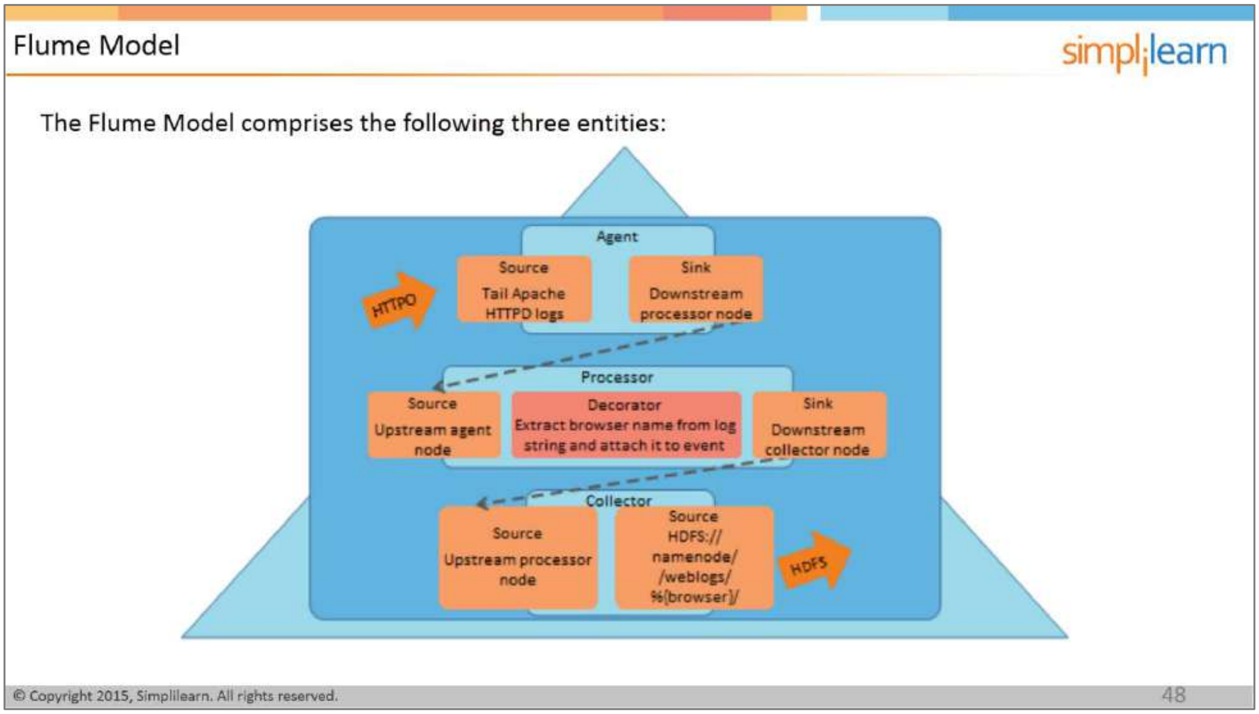
Apache Flume – Introduction

Apache Flume is a distributed and reliable service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS).

It has a simple and flexible architecture based on streaming data flows, which is robust and fault-tolerant.

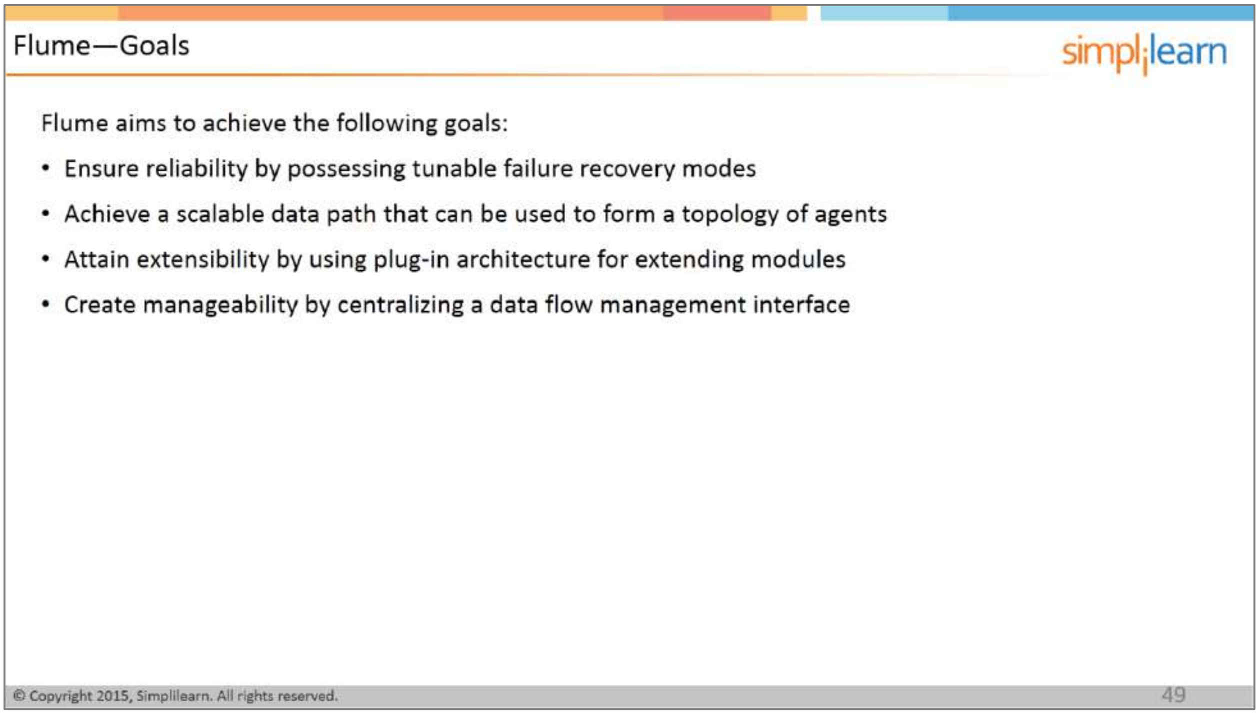
The diagram depicts how the log data is collected from a web server and is sent to HDFS using the source, channel, and sink components.

A Flume source consumes events delivered to it by an external source like a web server. When a Flume source receives an event, it stores it into one or more channels. The channel is a passive store that keeps the event until it's consumed by a Flume sink. One example is the file channel, which is backed by the local filesystem. The sink removes the event from the channel and puts it into an external repository like HDFS or forwards it to the Flume source of the next Flume agent or next hop in the flow.



Flume Model

The Flume model comprises three entities: an agent, a processor, and a collector. The agent is responsible for receiving data from an application. Running flume agents will ensure data is continuously ingested into Hadoop. The processor component is responsible for performing intermediate processing of jobs. The collector component is responsible for writing data to the permanent storage (HDFS).



Flume—Goals

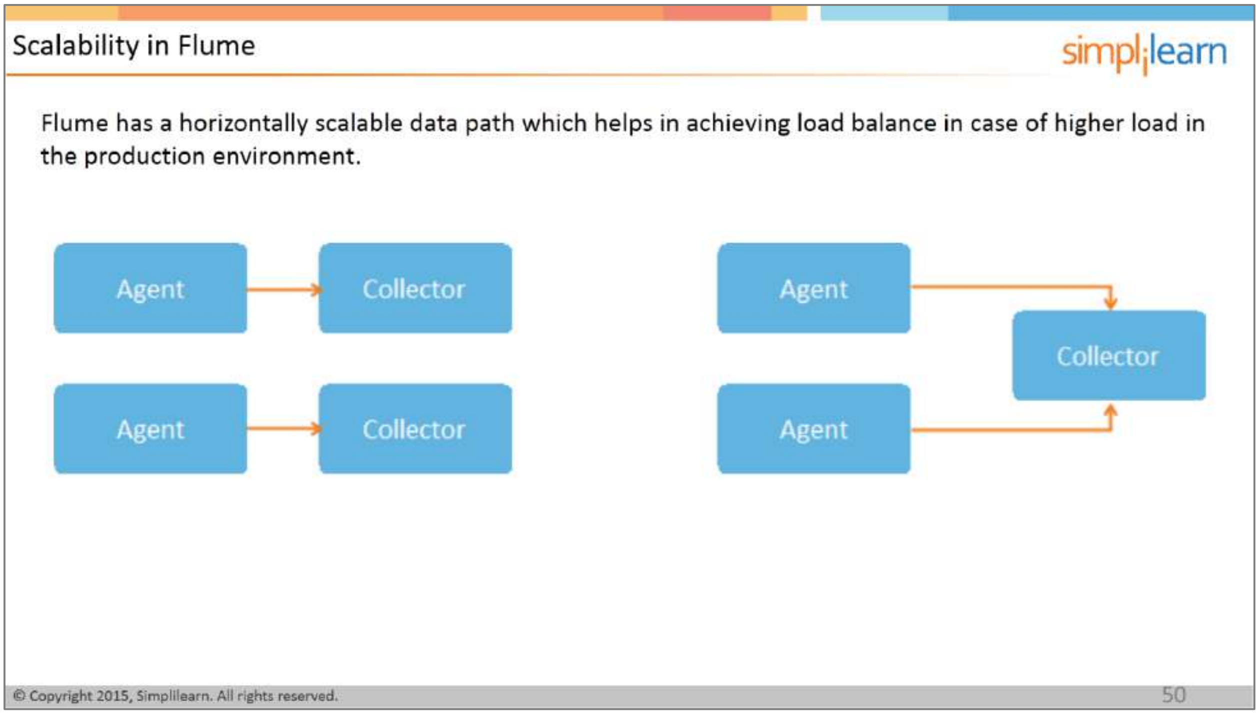
Flume works towards attaining certain key goals for data ingestion.

Flume aims to ensure reliability, as it has tunable failure recovery modes.

It also aims to achieve scalability by providing a horizontally scalable data path that can be used to form a topology of agents.

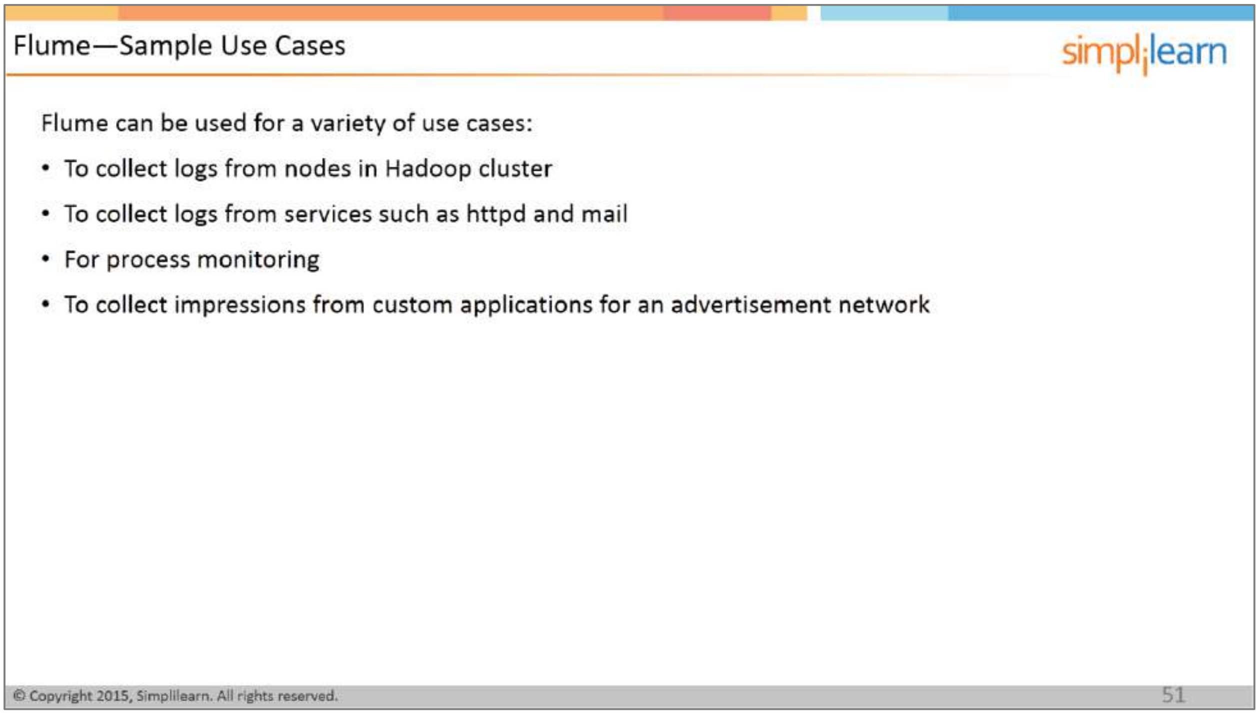
For extensibility, Flume leverages plug-in architecture for extending modules.

For manageability, Flume provides a centralized data flow management interface, hence making it convenient to manage across nodes.



Scalability in Flume

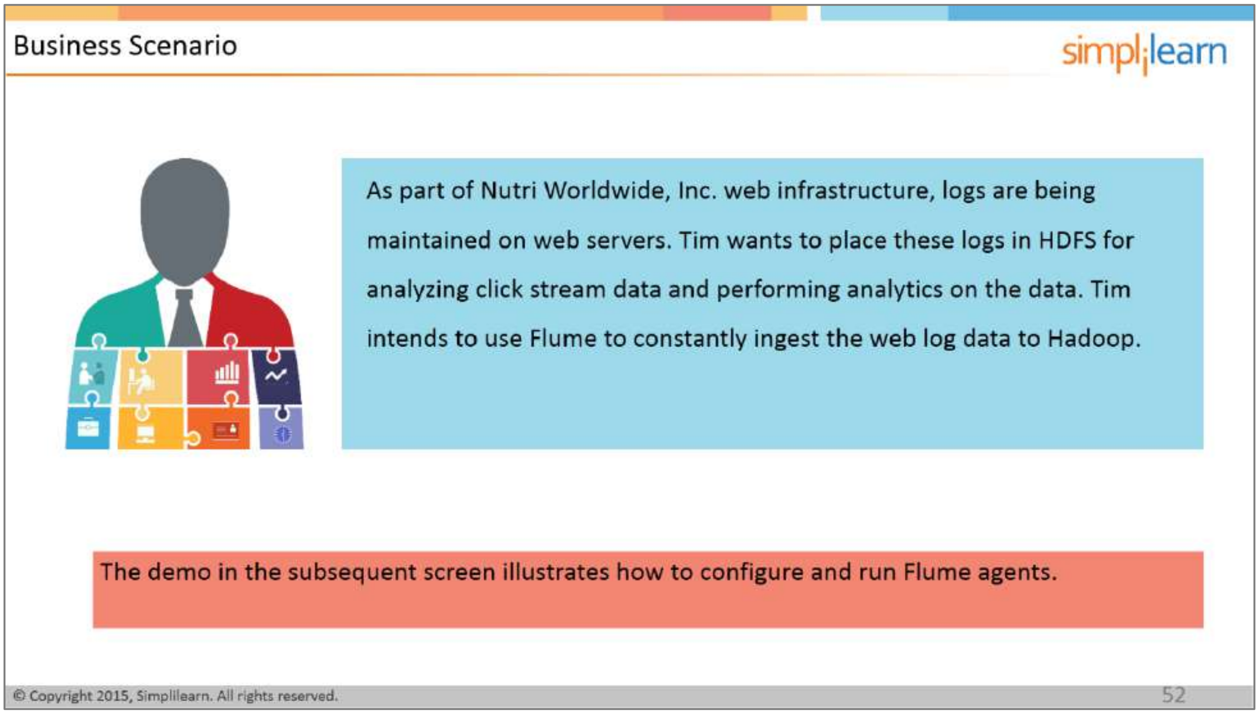
Flume is scalable. It has a horizontally scalable data path, which helps in achieving load balance in case of higher load in the production environment. The given image depicts an example of horizontal scalability. Assume there are two agents and two collectors. If one collector is down, then the two agents will scale down to just one collector.



Flume: Sample Use Cases

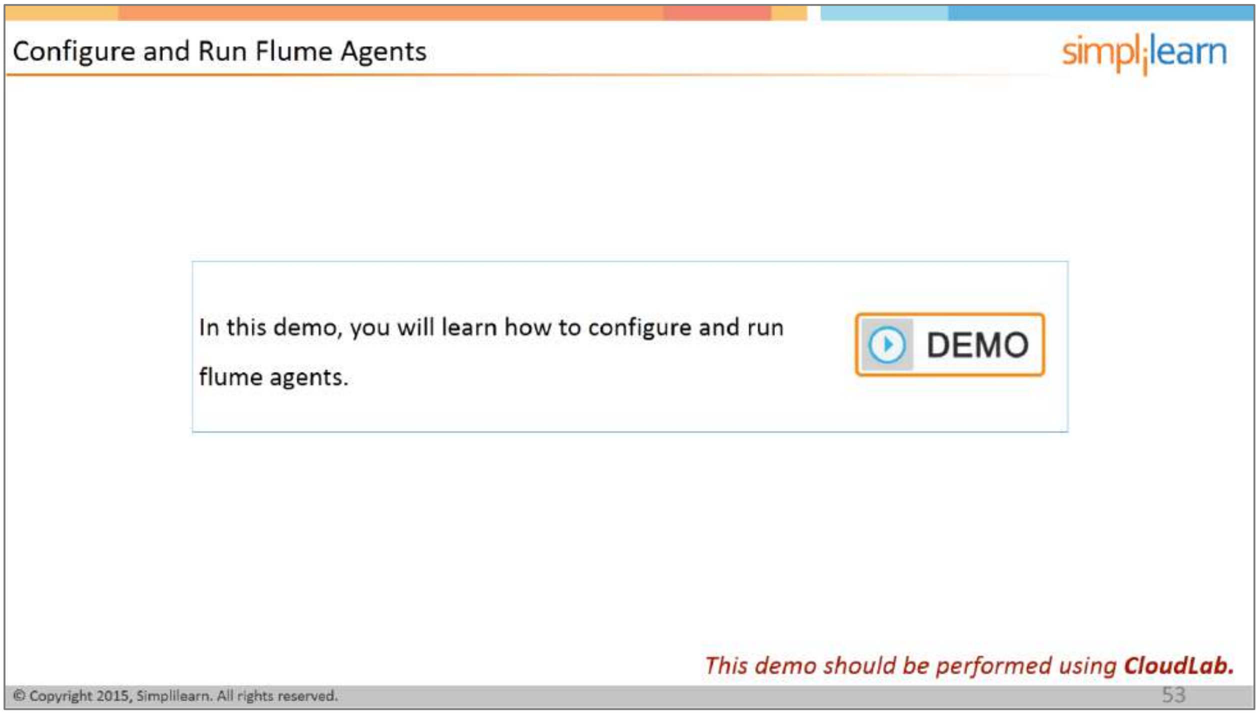
Flume can be used for a variety of use cases.

It can be used for collecting logs from nodes in a Hadoop cluster; for collecting logs from services such as httpd and mail; and for process monitoring. Further, Flume has also been widely used for collecting impressions from custom applications for an advertisement network like Meebo.



Business Scenario

As part of the Nutri Worldwide, Inc. web infrastructure, logs are being maintained on web servers. Tim wants to put these logs in HDFS for analyzing click stream data and processing analytics on the data. Tim intends to use Flume to constantly ingest the web log data to Hadoop.



Configure and Run Flume Agents

Let us summarize the steps performed:

1. Once flume is installed, go to the Flume configuration directory.
2. In this case, it /etc/flume/conf.
3. List the contents using the l-s command.
4. Notice the flume.conf file.
5. Open the flume.conf file for editing in vi editor.
6. Enter the configuration parameters including source, sink, and channel.
7. Enter the source web log and HDFS sink path. Open log4j.properties for editing.
8. Configure the parameters that are log directory and log message type.
9. Now invoke the flume agent. You will see log messages appearing to confirm flume is launched successfully.
10. In another terminal, simulate the creation of web log data.
11. Use a python script to create random web log data which can be ingested in HDFS. The output will be created by a weblog and flume monitors its directory.
12. List the log file to verify if it is created.
13. Now, check the HDFS sink to verify whether the data is inserted successfully.
14. List the contents of /flume/ events.
15. If the HDFS directory shows the data, it means the flume agent has run successfully.

Spot the Error

Jenny has created a table called export_data to store information on export. She needed to export this data to MySql, however, she encountered an error. Identify the code she is entering incorrectly.

Spot the incorrect code.

```
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] # ls export*
export_data
[ hadoop@simplilearn ] # cat export_data
1,Laptop
2,Desktop[ hadoop@simplilearn ] #

[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] # hadoop fs -copyFromLocal export_data export_data
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] # sqoop export --copyFromjdbc:mysql://localhost/test --table exported_table --export-dir expo
rt_data
```

 55

Spot the Error

The incorrect code is:

```
[ hadoop@simplilearn ] # sqoop export --copyFromjdbc:mysql://localhost/test --table exported_table --export-dir expo
rt_data
```

The correct syntax is:

```
[ hadoop@simplilearn ] # sqoop export --copyFromjdbc:mysql://localhost/test --table exported_table --export-dir expo
rt_data
```

 56

simplilearn

QUIZ 1

Which of the following options is not a ZooKeeper entity?

a. Master

b. Leader

c. Follower

d. Observer

 58

simplilearn

QUIZ 2

Which of the following features of the ZooKeeper is used for getting notifications of all followers and observers to the leaders?

a. Znode

b. Watch

c. Triggers

d. Sessions

 60

simplilearn

QUIZ 3

What does Sqoop offer?

a. Capability to import the database to the Hadoop System

b. Capability to export the database from Hadoop System

c. Capability to alter the database and commit the database to the Hadoop System

d. Both a and b

 62

simplilearn

QUIZ 4

Which of the following components is essential to perform transfer operations in Sqoop?

a. Connectors

b. Database compatibility

c. Tables

d. Hive

 64

simplilearn

QUIZ 5

Mike, the Data Analyst, is proficient with SQL. He would like to run an ad hoc analysis on data in HDFS cluster. Which of the following is a data warehousing software built on top of Apache Hadoop that defines a simple SQL-like query language well-suited for Mike's task?

a. ZooKeeper

b. HBase

c. Sqoop

d. Hive

 66

simplilearn

QUIZ 6

You need to import a portion of a relational database every day as files to HDFS and generate Java classes to interact with your imported data. Which of the following tools should you use to accomplish this?

a. Pig

b. Hive

c. Sqoop

d. Both a and b

 68

simplilearn

QUIZ 7

Which of the following tools is used by a Developer to transfer MySQL databases to and from HDFS?

a. Pig

b. Hive

c. Sqoop

d. ZooKeeper

simplilearn

QUIZ 8

Which type of scalability is supported by Apache Flume?

a. Horizontal Scalability

b. Vertical Scalability

c. Diagonal Scalability

d. No Scalability

QUIZ 9

Which of the following options is NOT a Flume entity?

a. Agent

b. Collector

c. Sink

d. MapReduce

 74

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which of the following options is not a ZooKeeper entity?	The Master entity does not exist in ZooKeeper.
2	Which of the following features of the ZooKeeper is used for getting notifications of all followers and observers to the leaders?	The watch feature is used for getting notifications of all followers and observers to the leaders.
3	What does Sqoop offer?	Sqoop offers the capability to import and export a database to or from the Hadoop system.
4	Which of the following components is essential to perform transfer operations in Sqoop?	Sqoop requires connectors to perform transfer operations.
5	Mike, the Data Analyst, is proficient with SQL. He would like to run an ad hoc analysis on data in HDFS cluster. Which of the following is a data warehousing software built on top of Apache Hadoop that defines a simple SQL-like query language well-suited for Mike's task?	Hive can be used for performing import and export operations of data. Mike can perform his SQL actions in SQL DB and then perform an import operation of the DB in HDFS.
6	You need to import a portion of a relational database every day as files to HDFS and generate Java classes to Interact with your imported data. Which of the following tools should you use to accomplish this?	Hive will be useful to perform CRUD operations, whereas Sqoop is required to import/export the database to/from Hadoop.
7	Which of the following tools is used by a Developer to transfer MySQL databases to and from HDFS?	Sqoop component is used by the developer to transfer MySQL databases to and from HDFS.
8	Which type of scalability is supported by Apache Flume?	Apache Flume supports horizontal scalability.
	Which of the following options is NOT a Flume entity?	MapReduce does not come under the Flume entities.

Case Study—ZooKeeper

simplilearn

Scenario | Analysis | Solution

XY Networks provides network security support to many organizations. It has set up a cluster of servers, and created an in-house application that monitors these clusters. It faces multiple problems in coordinating these servers, and there are numerous partial failures and race conditions. Users complain the application's behavior to be unpredictable. They found the handling of distributed coordination in their software is complicated, and will take 200 Man-Days of effort to fix it. Instead, they want to explore the ZooKeeper tool, which is a part of Apache Foundation, and is a library of recipes for distributed process coordination.

 77

Case Study—ZooKeeper

simplilearn

Scenario | Analysis | Solution

The IT team researched on Apache ZooKeeper. It has very expressive APIs and well-tested mechanism for handling distributed process coordination. Some of the recipes provided by ZooKeeper are:

1. Handling partial failures
2. Leader selection
3. Avoiding deadlocks
4. Avoiding race conditions

 78

Case Study—ZooKeeper

Scenario | Analysis | **Solution**

Perform the following steps to explore the nodes and watches in ZooKeeper:

1. Open a ZooKeeper session on one node.
2. Create an ephemeral znode.
3. Create few sequential ephemeral znodes under the node.
4. List the znodes.
5. Retrieve the znode data.
6. Open another ZooKeeper session.
7. Create a watch on the ephemeral nodes from the second session.
8. Delete one of the nodes in the first session, and view the alerts in second session,
9. Quit the first session, and view the alerts in second session.

 79

Case Study—Sqoop

simplilearn

Scenario | Analysis | Solution

XY Invest provides investment advice to high net worth individuals, and maintains stock market data for many exchanges. It has stock market data files that are critical for analysis and monitoring. It chose to implement Hadoop for storing stock market data files, and decided to use Hive for analysis on Hadoop for the company data scientists' ease of use. Its current data is in RDBMS, which is used as a transaction database, and has to be transferred to Hadoop and Hive. The company's IT team has heard that Sqoop is a popular tool for ingesting RDBMS data into Hadoop and Hive.

 81

Case Study—Sqoop

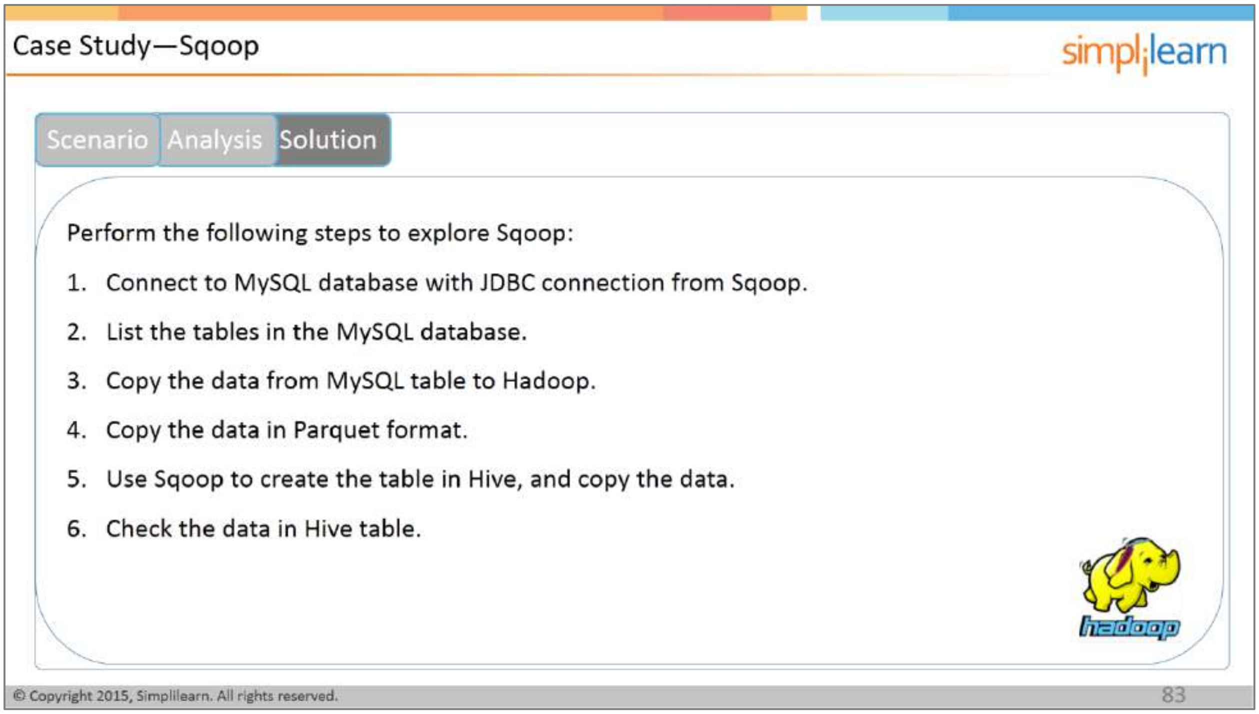
simplilearn

Scenario | Analysis | Solution

The IT team does research on Sqoop, and finds that it is efficient in copying large amounts of data from RDBMS into Hadoop. Sqoop is also effective in importing back aggregated data from Hadoop to RDBMS for other analysis. Advantages of using Sqoop for data ingestion are:

1. It can connect to different types of databases.
2. It can read the table schema, and convert data types into Hadoop and Hive.
3. It can create tables in Hive from database schema.
4. It uses Hadoop map-only jobs for transferring data, and uses 4 mappers by default.
5. It uses the primary key of the table for splitting data for the mappers.
6. It also generates Java-class file for the table with getter and setter methods.
7. It is capable of copying data in popular Parquet format.

 82


Case Study—Sqoop
simplilearn
Scenario
Analysis
Solution
Perform the following steps to explore Sqoop:
1. Connect to MySQL database with JDBC connection from Sqoop.
2. List the tables in the MySQL database.
3. Copy the data from MySQL table to Hadoop.
4. Copy the data in Parquet format.
5. Use Sqoop to create the table in Hive, and copy the data.
6. Check the data in Hive table.
hadoop
© Copyright 2015, Simplilearn. All rights reserved.
83

Case Study—Flume

Scenario | Analysis | Solution

XY Networks provides network security support to many organizations. It has set up a cluster of servers, and created an in-house application that monitors these clusters. It collects System Log files from numerous customers in real-time, and stores these files in Hadoop for analytics processing. A highly scalable and fault-tolerant system is required for gathering these log files. The company's IT team has heard that Flume and Sqoop are popular tools for data ingestion into Hadoop. Since their data is unstructured log files, Flume is the tool of choice for their business need.

 85

Case Study—Flume

Scenario | Analysis | Solution

The IT team does research on Flume, and finds that it has a source-sink architecture with loosely connected sources and sinks. Channels are used to receive data from source to sinks. The advantages of using Flume for data ingestion are:

1. It can process millions of log messages per minute.
2. It is horizontally scalable, and can add new machines to cluster to increase the capacity.
3. Provides real-time processing of log files.
4. It is highly fault-tolerant. If the sink is down, messages will accumulate in the channel and will be delivered once the sink functions.

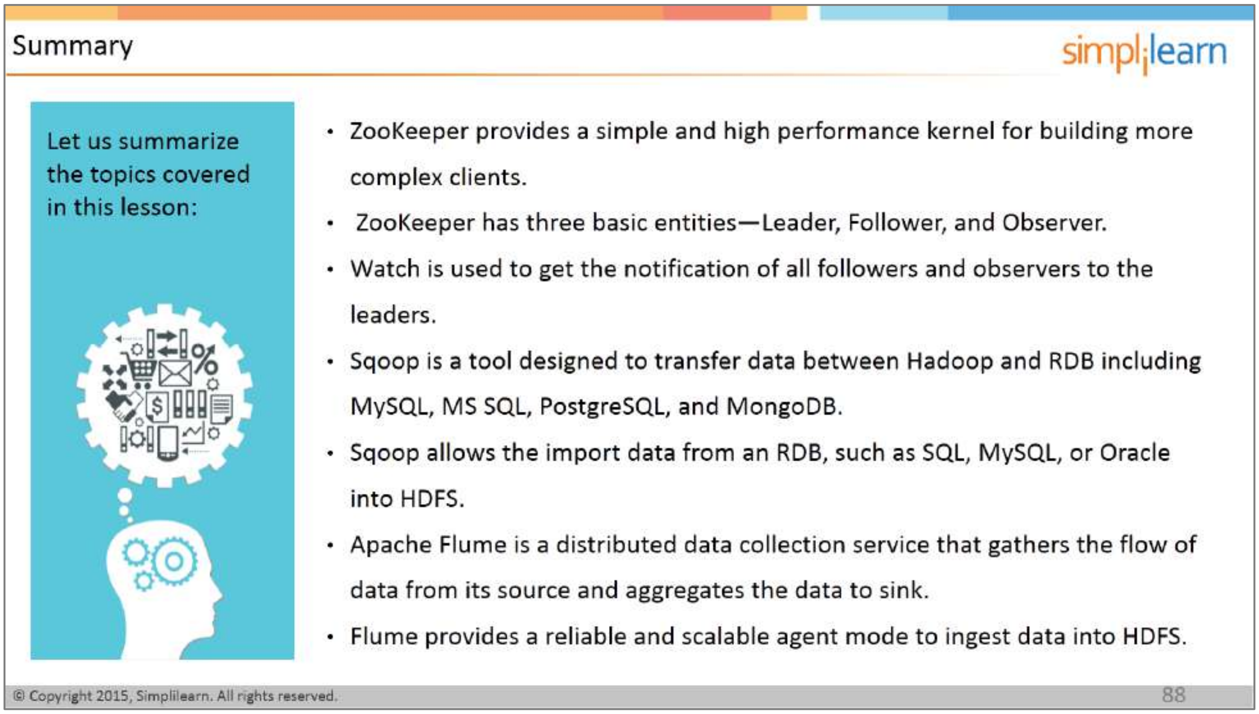
 86

Case Study—Flume

Scenario | Analysis | **Solution**

Perform the following steps to explore Flume:

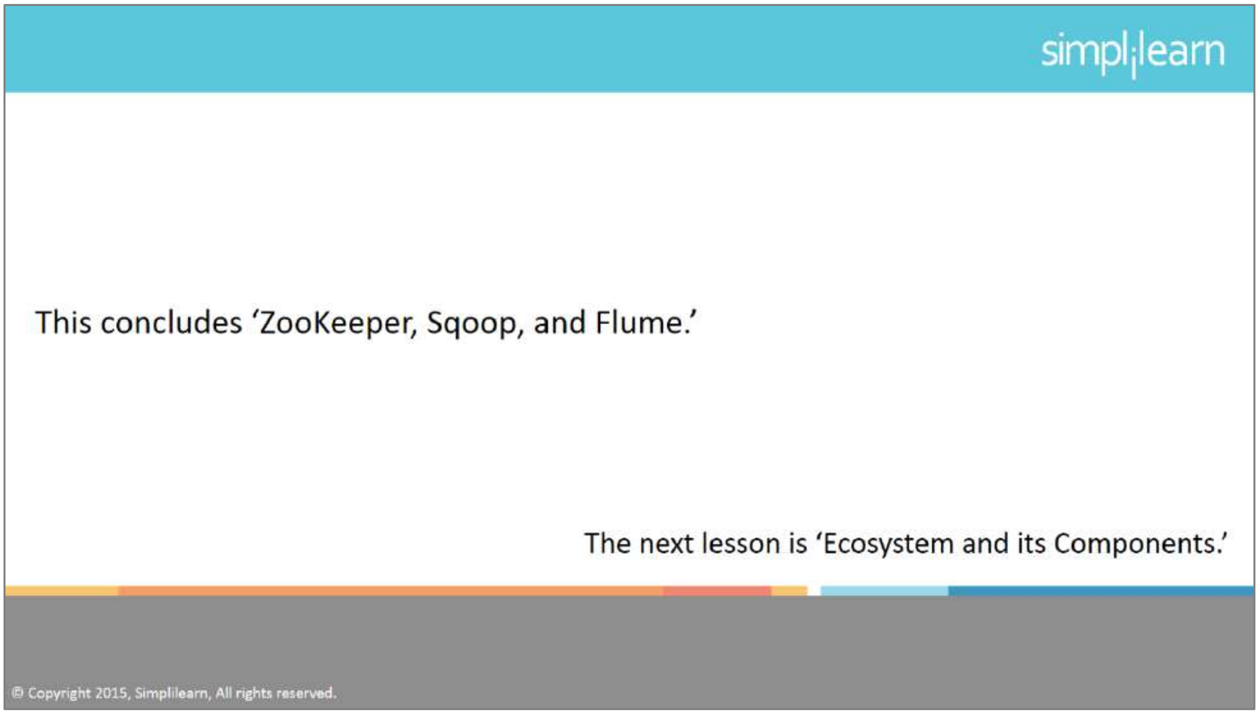
1. Create a Properties file that defines source, sink, and channel.
2. Make the source to extract data from a log file and sink and store it into HDFS.
3. Run Flume with this Properties file.
4. Add data to the log file.
5. Check if the data is copied to HDFS by Flume.
6. As more data is added to log file, Flume picks up the data and adds to HDFS.



Summary

Let us summarize the topics covered in this lesson:

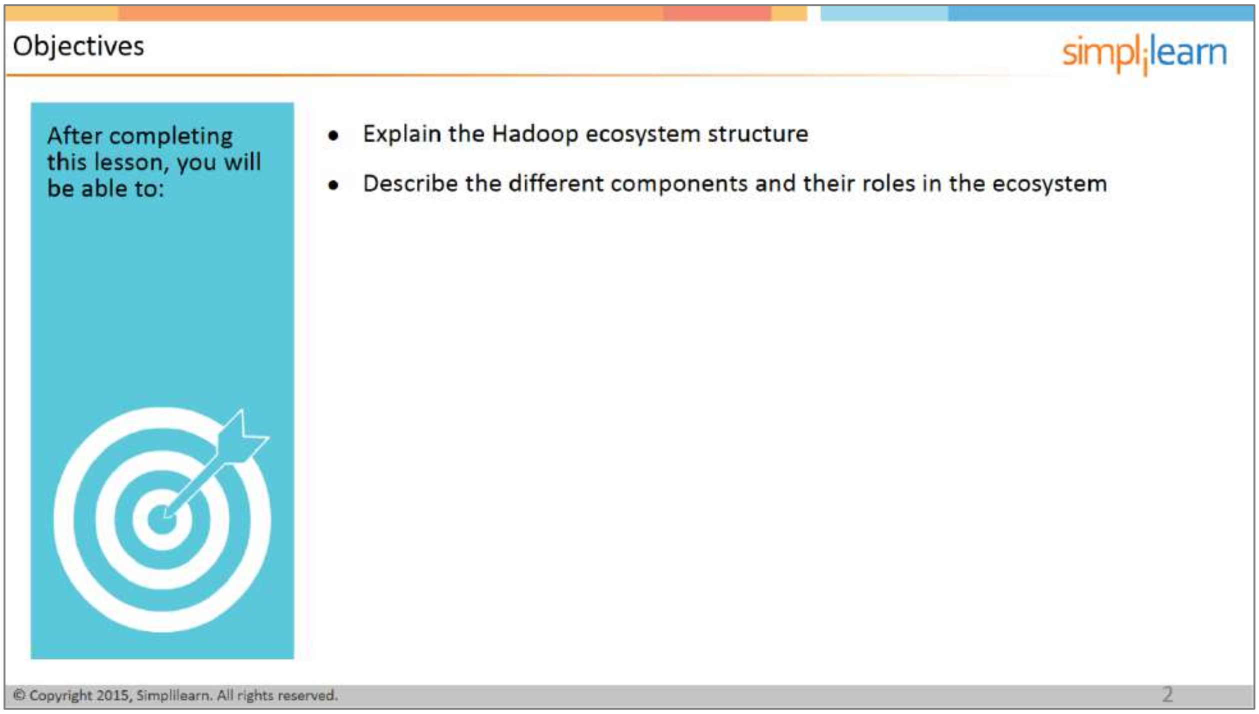
- ZooKeeper provides a simple and high performance kernel for building more complex clients.
- ZooKeeper has three basic entities—Leader, Follower, and Observer.
- Watch is used to get the notification of all followers and observers to the leaders.
- Sqoop is a tool designed to transfer data between Hadoop and RDB including MySQL, MS SQL, PostgreSQL, and MongoDB.
- Sqoop allows the import data from an RDB, such as SQL, MySQL, or Oracle into HDFS.
- Apache Flume is a distributed data collection service that gathers the flow of data from its source and aggregates the data to sink.
- Flume provides a reliable and scalable agent mode to ingest data into HDFS.



Conclusion

This concludes the lesson 'ZooKeeper, Sqoop, and Flume.' The next lesson will cover 'Ecosystem and its Components.'

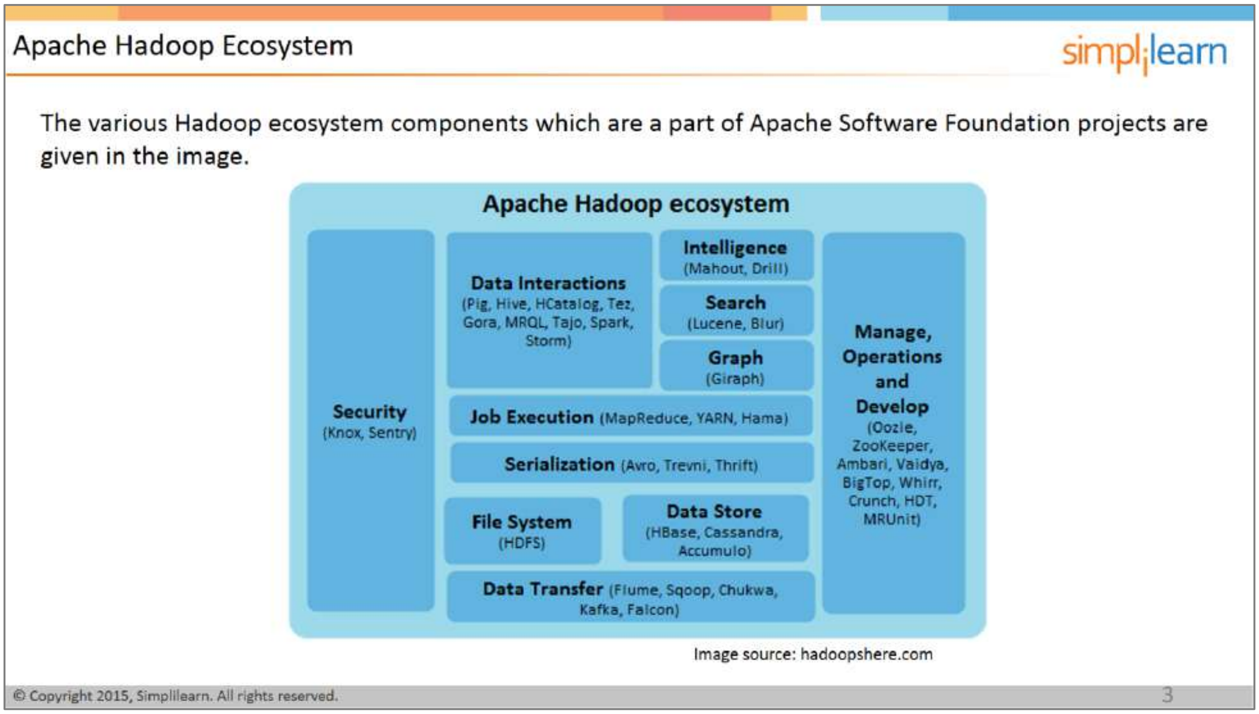
Lesson 11—Ecosystem and its Components



Objectives

After completing this lesson, you will be able to:

- Explain the structure of the Hadoop ecosystem
- Describe the different components and their roles in the ecosystem



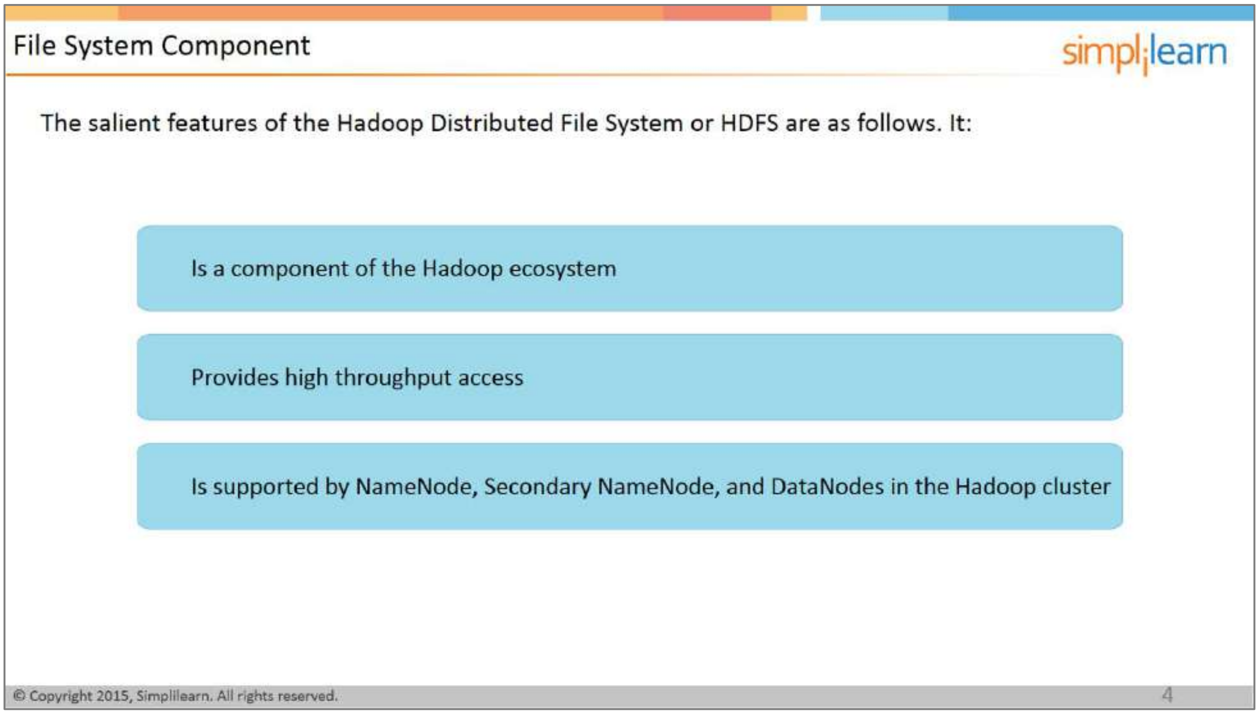
Apache Hadoop Ecosystem

The image displays the various Hadoop ecosystem components which are a part of Apache Software Foundation projects. Note that there are other commercial and open source offerings apart from the Apache projects.

Hadoop ecosystem components have been categorized as follows:

- File system
- Data store
- Serialization
- Job execution
- Work management
- Development
- Operations
- Security

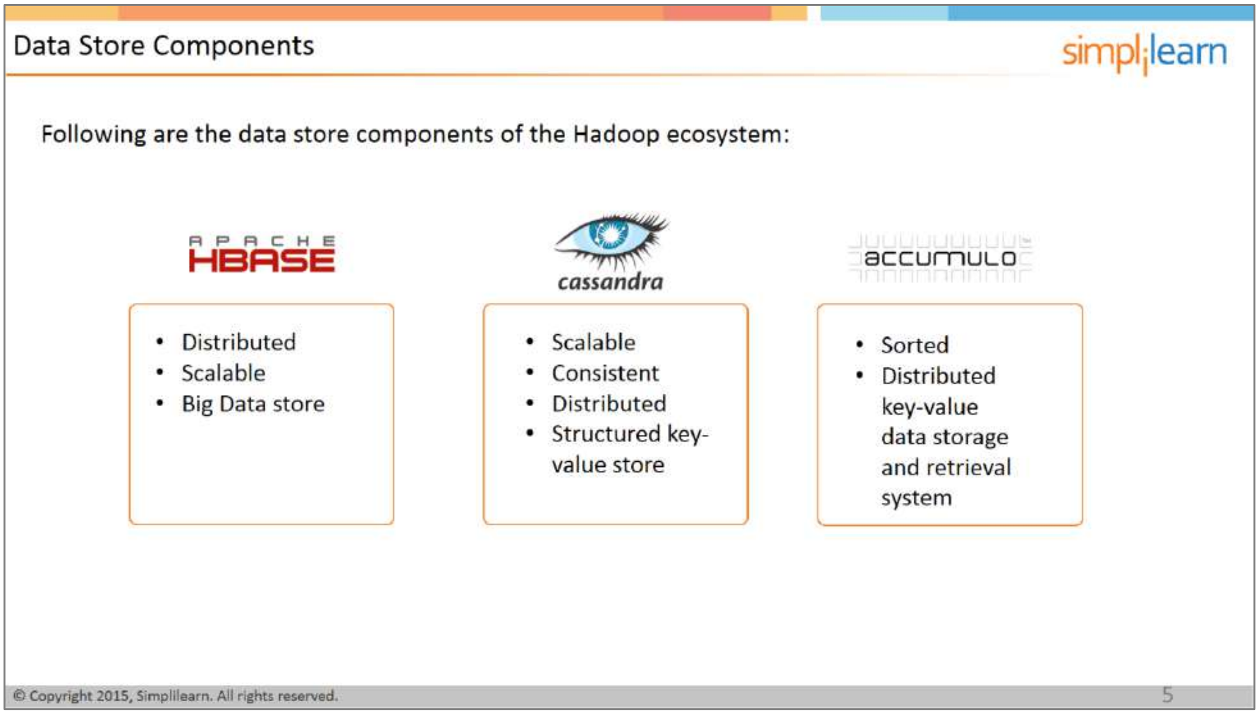
- Data transfer
- Data interactions
- Analytics and intelligence
- Search
- Graph processing



File System Component

The salient features of the Hadoop Distributed File System or HDFS are as follows.

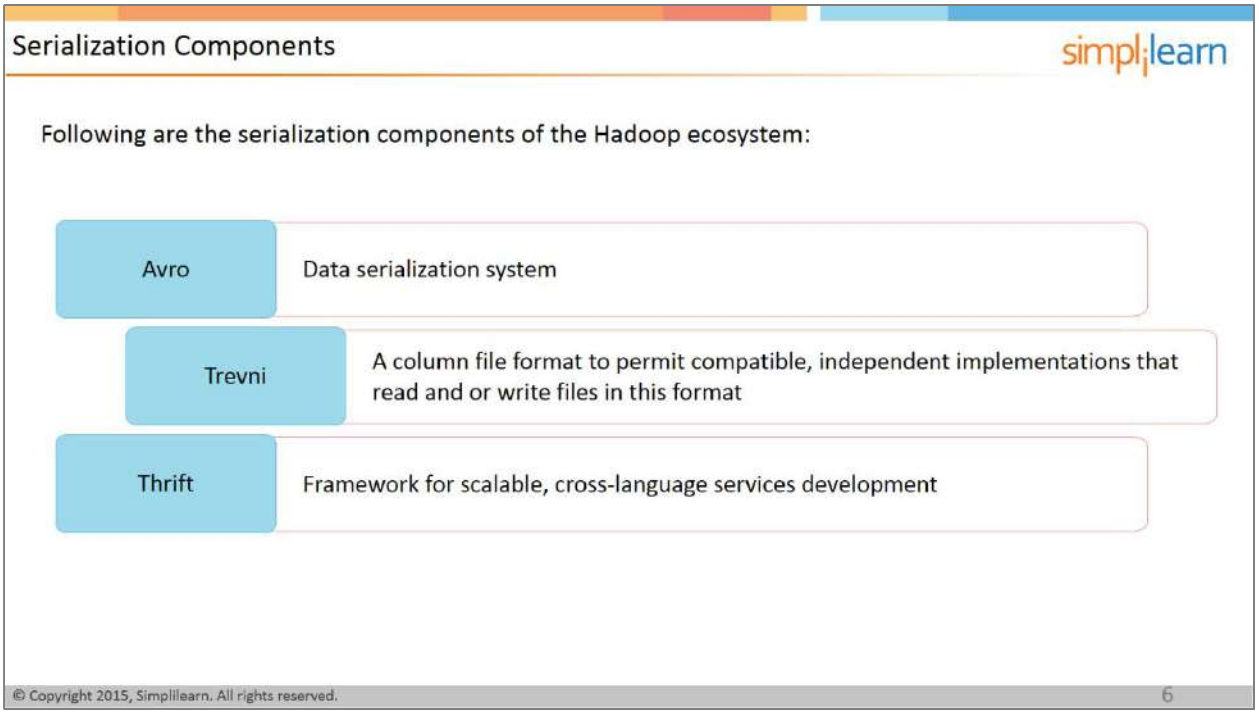
It is the component of the Hadoop ecosystem which provides high throughput access and is supported by NameNode, Secondary NameNode, and DataNodes in the Hadoop cluster.



Data Store Components

The data store components are comprised of HBase, Cassandra, and Accumulo.

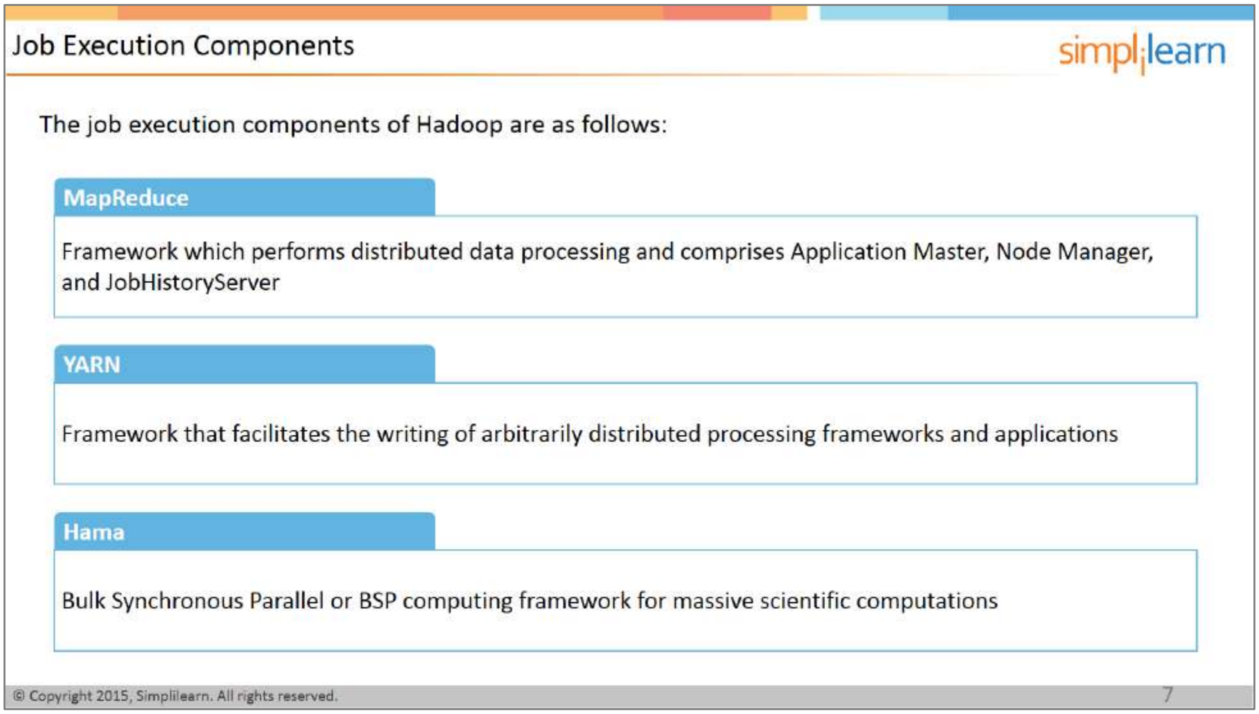
- HBase is a distributed, scalable, and big data store.
- Cassandra is a highly scalable, consistent, distributed, and structured key-value store.
- Accumulo is a sorted, distributed key-value data storage and retrieval system.



Serialization Components

The serialization components are Avro, Trevni, and Thrift.

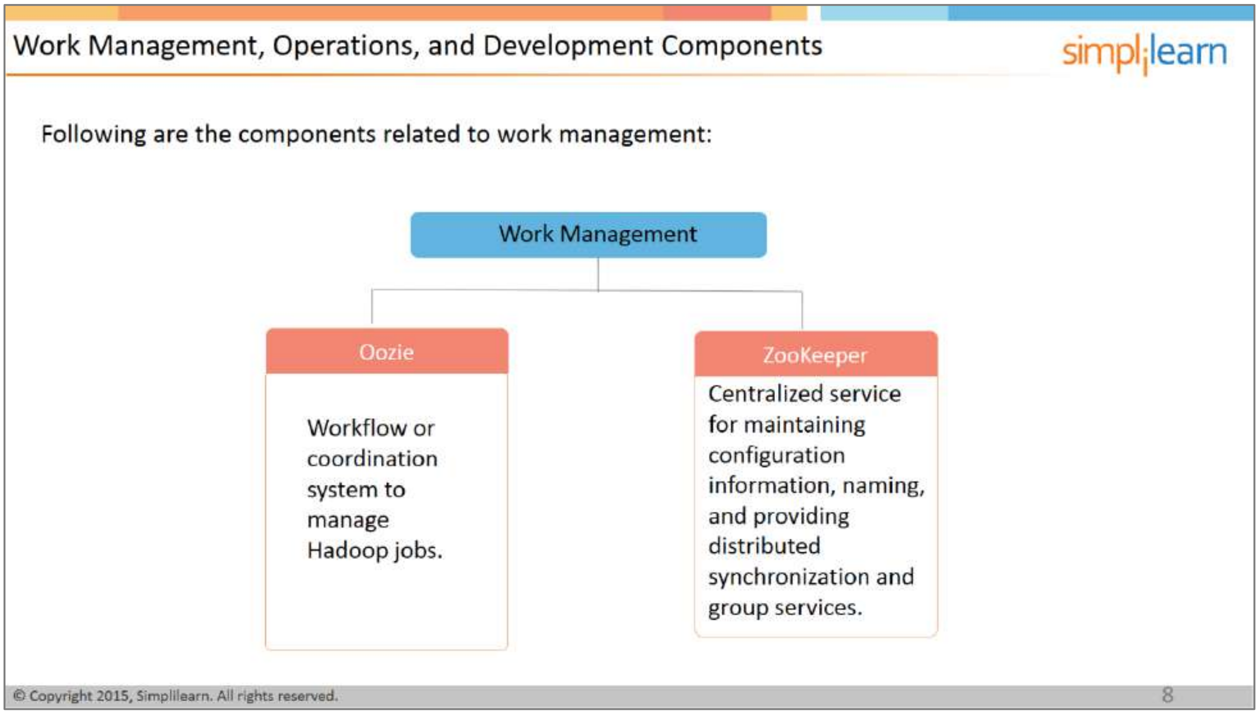
- Avro is a data serialization system.
- Trevni is a column file format used to permit compatible, independent implementations that read and /or write files in this format.
- Thrift is a framework for scalable, cross-language services development.



Job Execution Components

The job execution components include MapReduce, YARN, and Hama. The description of each component are as follows:

- MapReduce is a framework for performing distributed data processing. It comprises Application Master, Node Manager, and JobHistoryServer.
- YARN is a framework that facilitates the writing of arbitrarily distributed processing frameworks and applications.
- Hama is a Bulk Synchronous Parallel or BSP computing framework for massive scientific computations such as matrix, graph, and network algorithms.

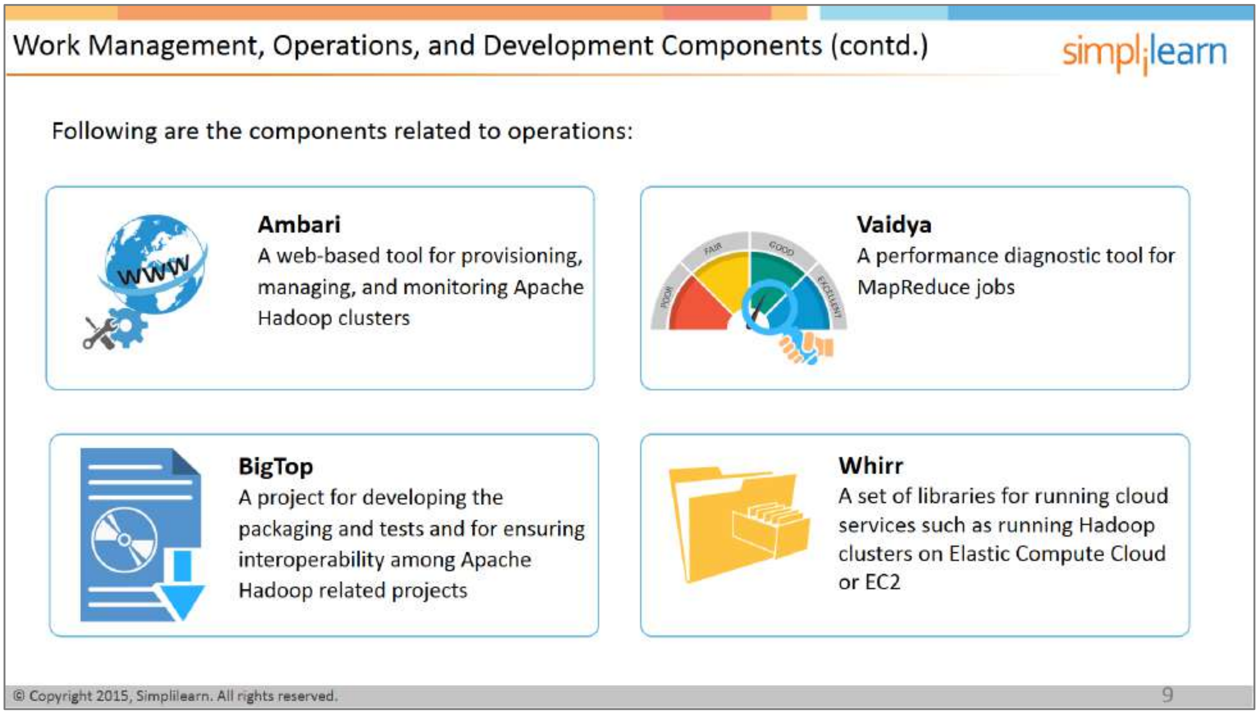


Work Management, Operations, and Development

There are a number of components related to work management, operations, and development.

The work management components of the Hadoop ecosystem are Oozie and ZooKeeper.

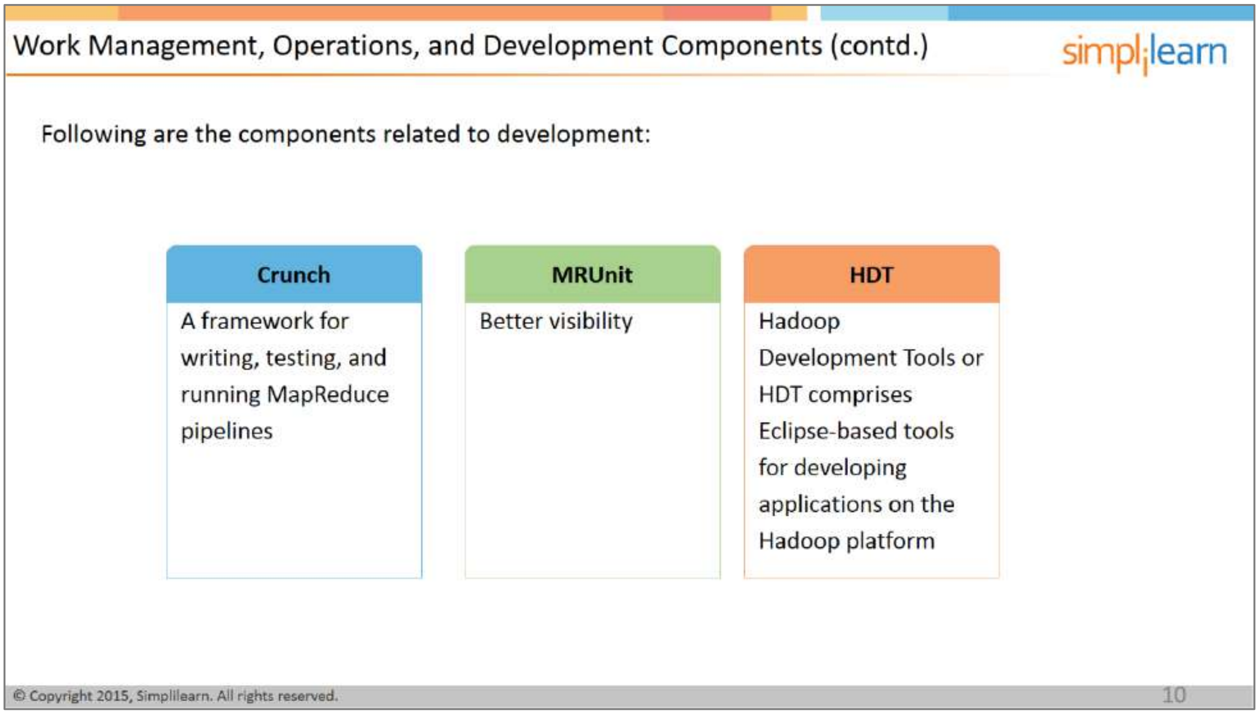
- Oozie is a workflow or coordination system that helps to manage Hadoop jobs.
- Zookeeper is a centralized service for maintaining configuration information and naming. It provides distributed synchronization and group services.



Work Management, Operations, and Development

The operation components of the Hadoop ecosystem are Ambari, Vaidya, BigTop, and Whirr.

- Ambari is a web-based tool used for provisioning, managing, and monitoring Apache Hadoop clusters.
- Vaidya is a performance diagnostic tool for MapReduce jobs.
- BigTop is a project for the development of packaging and tests. It also ensures interoperability among Apache Hadoop related projects.
- Whirr constitutes a set of libraries for running cloud services like running Hadoop clusters on Elastic Compute Cloud or EC2.



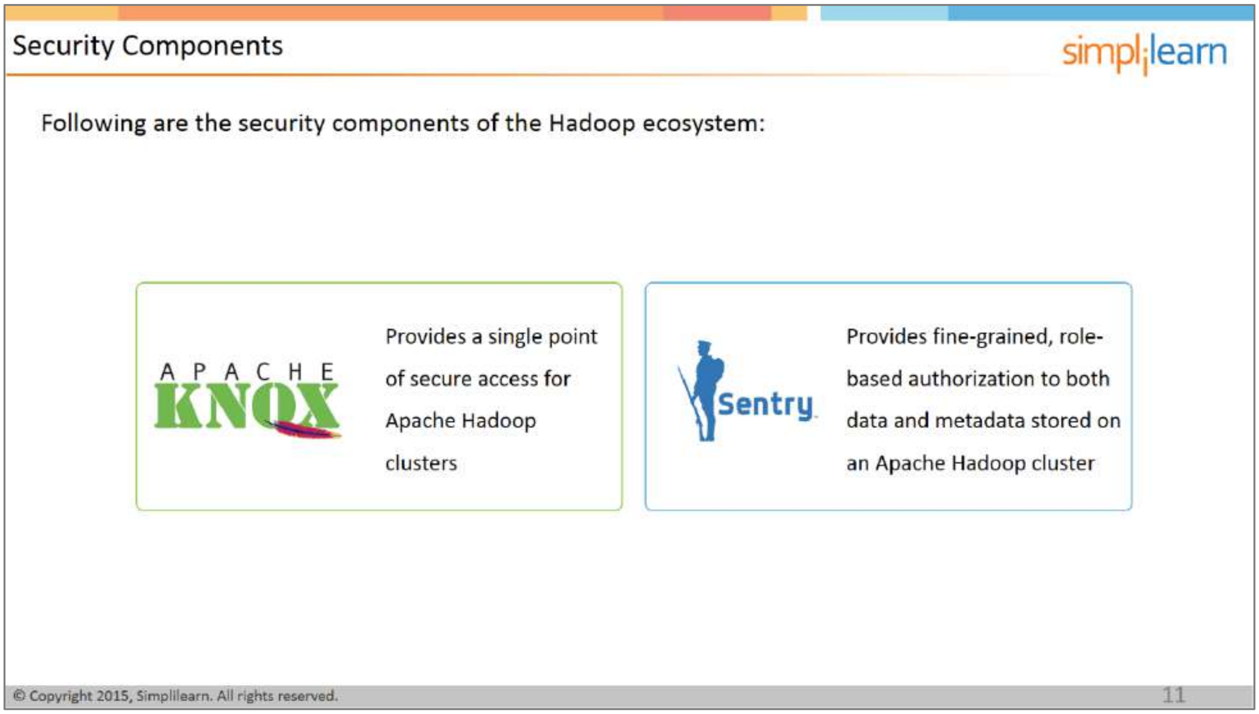
Work Management, Operations, and Development

The development components of the Hadoop ecosystem are Crunch, MRUnit, and Hadoop Development Tools or HDT.

Let's understand each development component in detail.

- Crunch is a framework for writing, testing, and running MapReduce pipelines.
- MRUnit is a Java library that helps developers unit test Apache Hadoop MapReduce jobs.
- Hadoop Development Tools or HDT comprises Eclipse-based tools for developing applications on the Hadoop platform.

Note that some of these tools may not be under active development, or they may not be supported currently by the open source community.

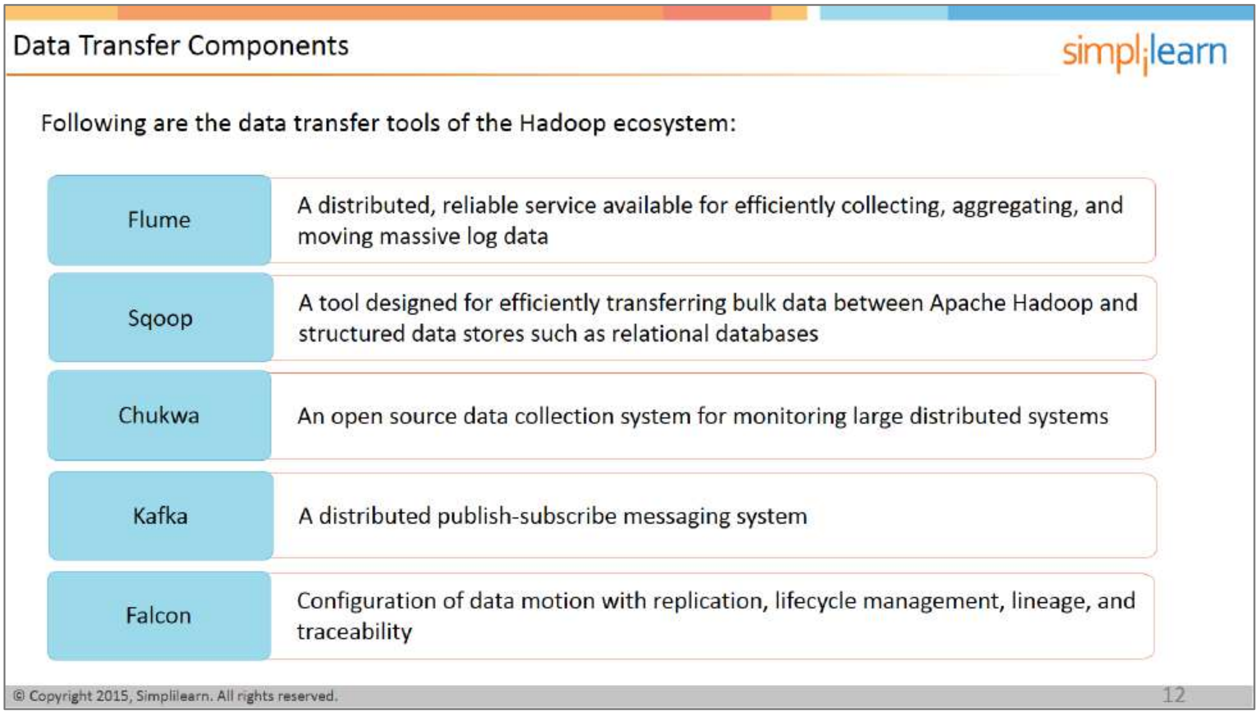


Security Components

The following information focuses on the security components, which are Knox and Sentry.

- Knox is a system that provides a single point of secure access for Apache Hadoop clusters.
- Sentry is a system for providing fine-grained, role-based authorization to both data and metadata stored on an Apache Hadoop cluster.

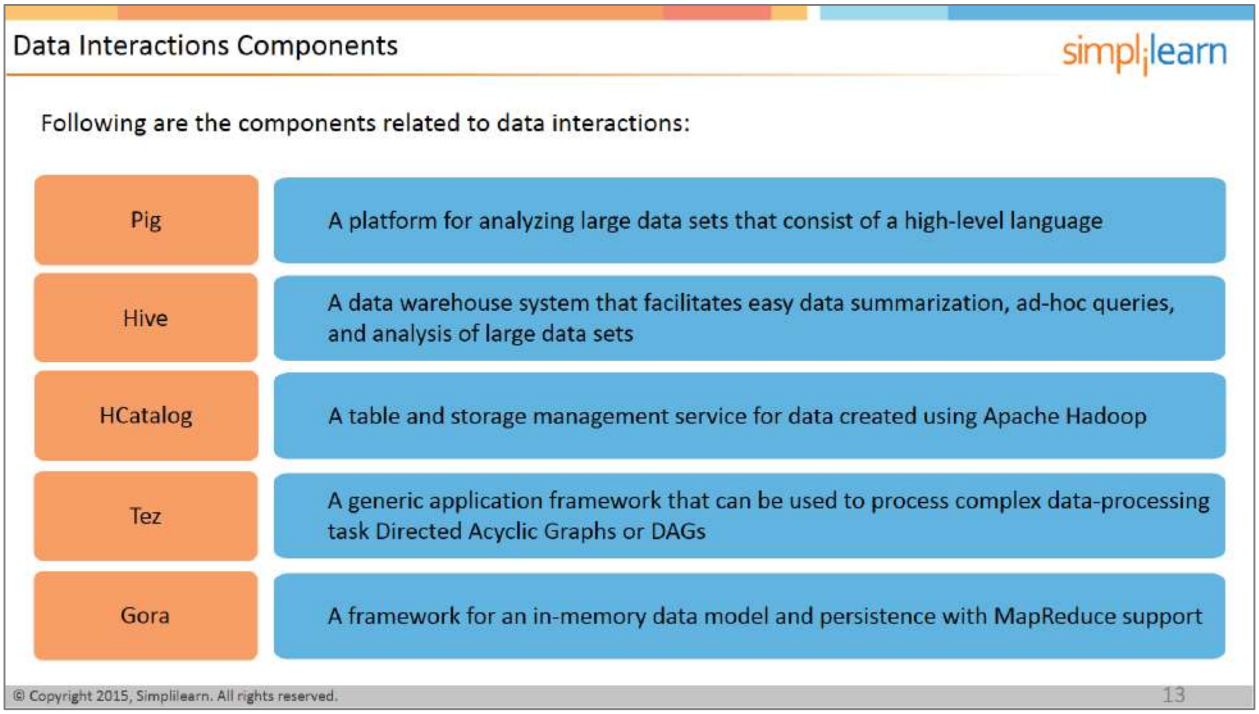
Both these projects are currently in the initial release phase.



Data Transfer Components

The data transfer components are Flume, Sqoop, Chukwa, Kafka, and Falcon.

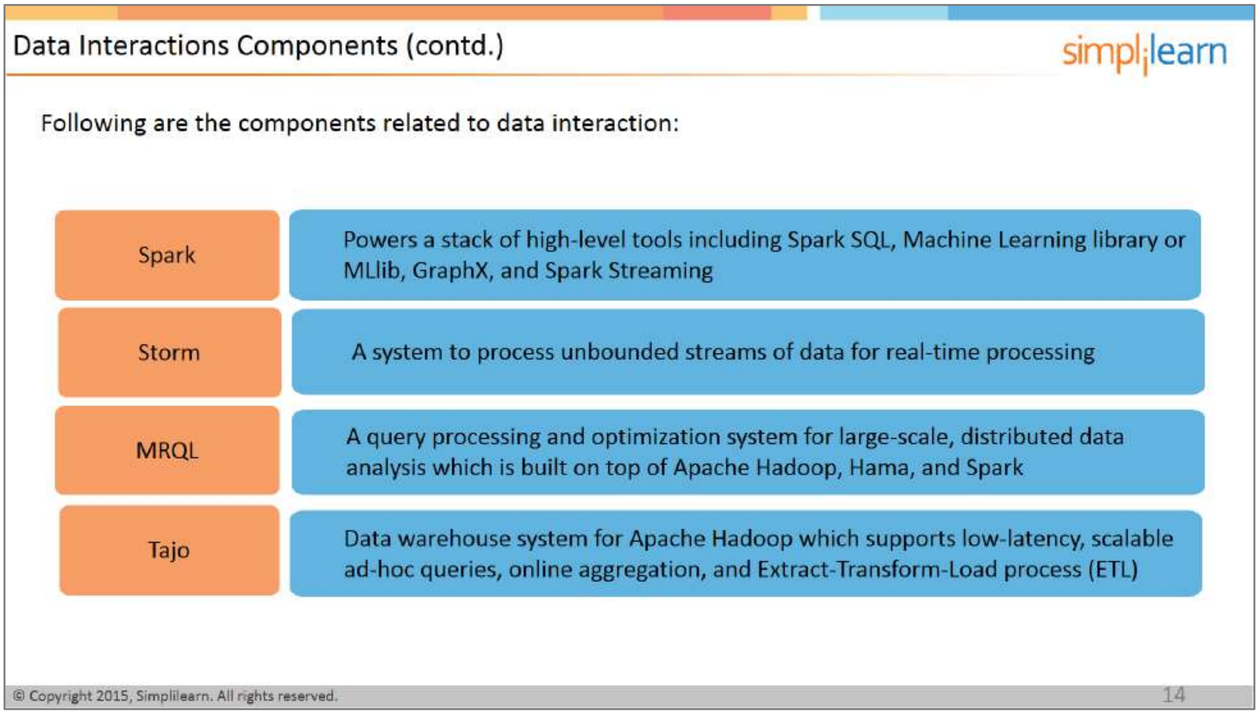
- Flume is a distributed, reliable service available for efficiently collecting, aggregating, and moving large amounts of log data.
- Sqoop is designed for efficiently transferring bulk data between Apache Hadoop and structured data stores such as relational databases.
- Chukwa is an open source data collection system for monitoring large distributed systems.
- Kafka is a distributed publish-subscribe messaging system.
- Falcon is a configuration of data-motion with replication, lifecycle management, lineage, and traceability.



Data Interactions Components

The components related to data interactions are:

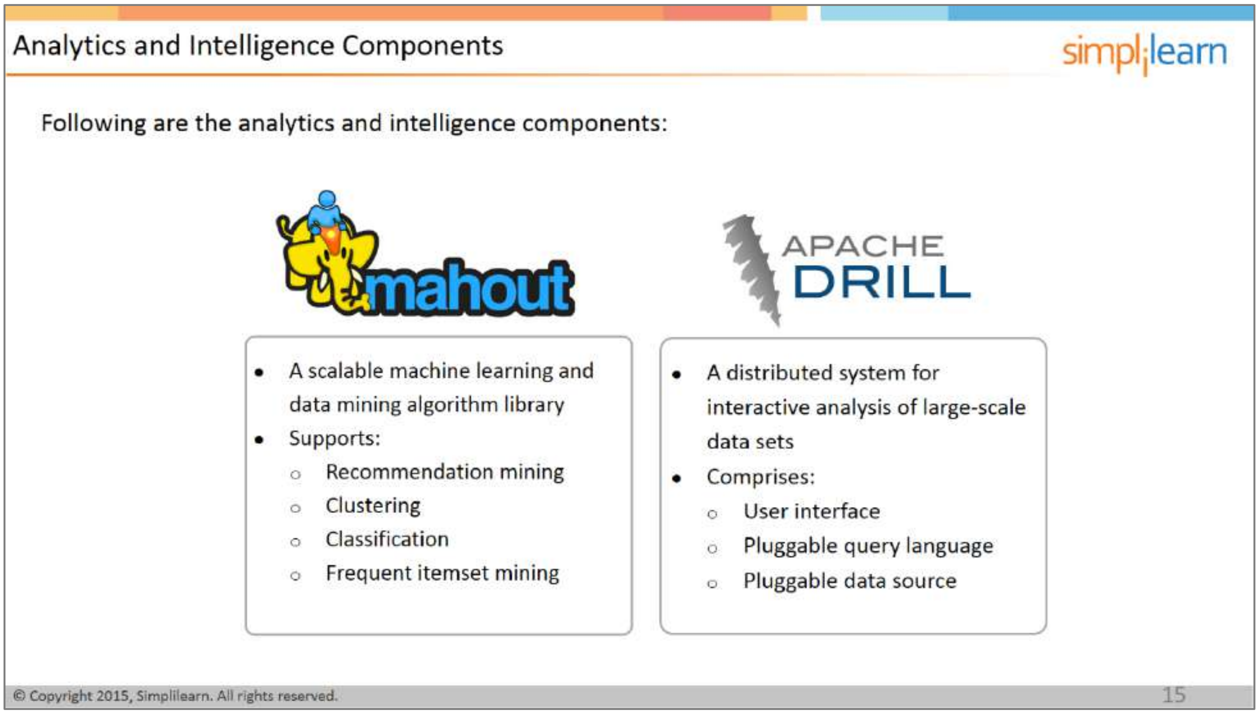
- Pig is a platform for analyzing large data sets that consist of a high-level language for expressing data analysis programs coupled with infrastructure for evaluating these programs.
- Hive is a data warehouse system that facilitates easy data summarization, ad-hoc queries, and analysis of large data sets stored in Hadoop compatible file systems.
- HCatalog is a table and storage management service for data created using Apache Hadoop.
- Tez is a generic application framework that can be used to process complex data-processing task DAGs or Directed Acyclic Graphs. It runs natively on Apache Hadoop YARN. It provides SQL queries on top of Hadoop.
- Gora is a framework for an in-memory data model and persistence with MapReduce support.



Data Interactions Components (contd.)

The additional components related to Data Interactions are:

- Spark powers a stack of high-level tools including Spark SQL, MLlib, GraphX, and Spark Streaming. MLib is for machine learning.
- Storm is a system to process unbounded streams of data for real-time processing.
- MRQL is a query processing and optimization system for large-scale, distributed data analysis built on top of Apache Hadoop, Hama, and Spark.
- Tajo is a data warehouse system for Apache Hadoop. It supports low-latency, scalable ad-hoc queries, online aggregation, and extract-transform-load process or ETL.

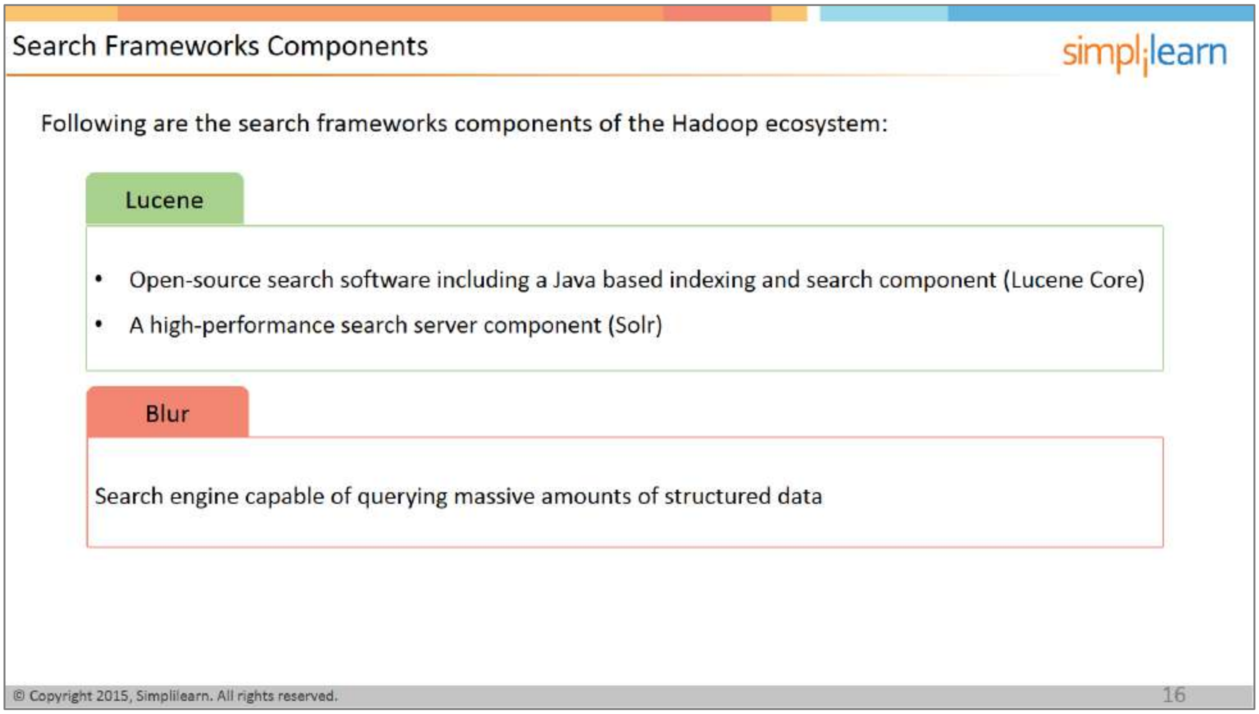


Analytics and Intelligence Components

The components comprising of analytics and intelligence in the Hadoop ecosystem are Mahout and Drill. Let us look at each one of them in detail.

- Mahout is a scalable machine learning and data mining algorithm library. It supports recommendation mining, clustering, classification, and frequent itemset mining.

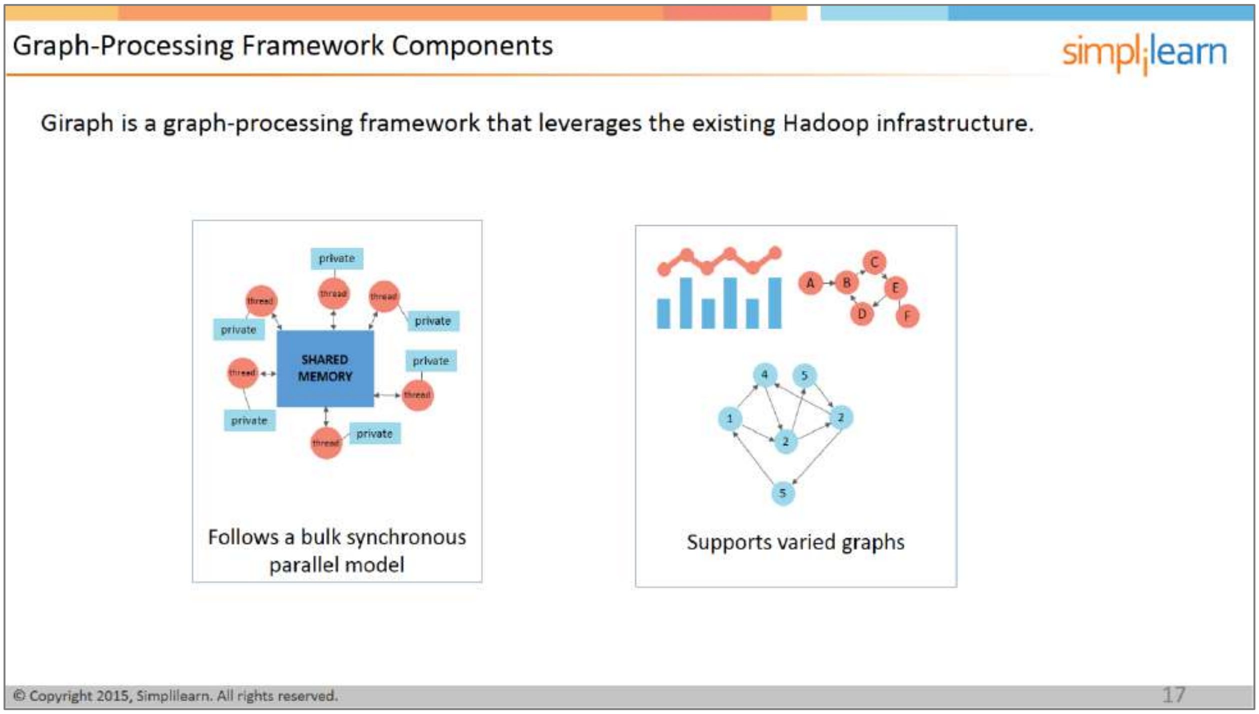
- Drill is a distributed system for interactive analysis of large-scale data sets. It is composed of user interfaces such as CLI and REST, pluggable query language, and pluggable data source. CLI stands for Command Line Interface, and REST stands for Representational State Transfer.



Search Framework Components

Lucene and Blur are two search frameworks.

- Lucene is an open-source search software. It includes a Java based indexing and search component, Lucene Core, and a high-performance search server component, Solr.
- Blur is a search engine capable of querying massive amounts of structured data at incredible speeds in a cloud computing environment.

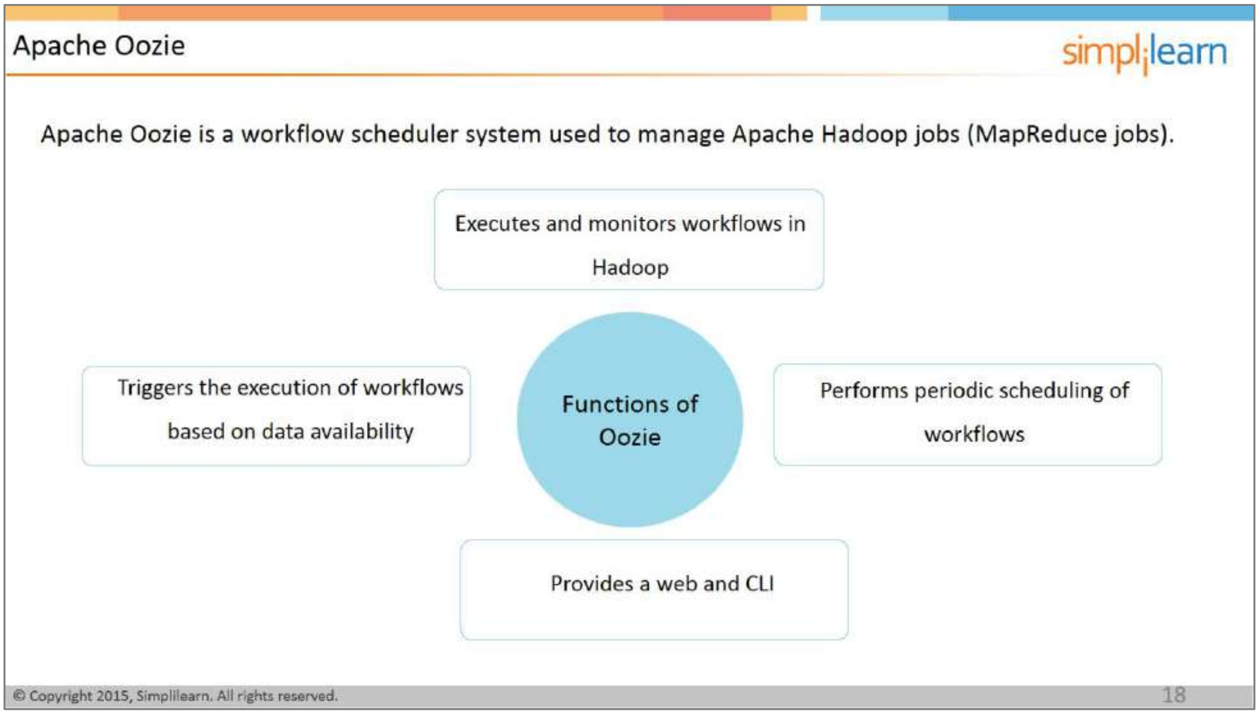


Graph-Processing Framework Components

Next we will discuss the graph-processing framework of the Hadoop ecosystem.

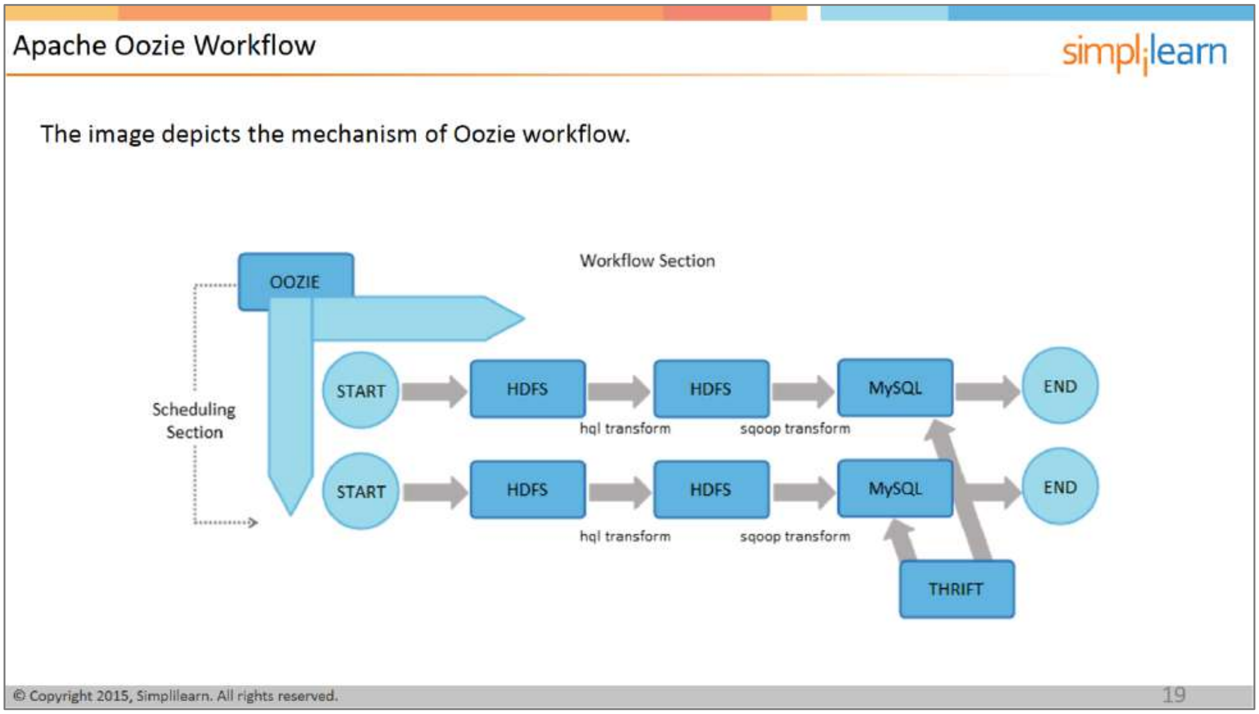
Giraph is a graph-processing framework that leverages the existing Hadoop infrastructure.

- It follows the bulk synchronous parallel model to run large-scale algorithms.
- It supports direct, undirect, weighted, unweighted, and multi graphs.



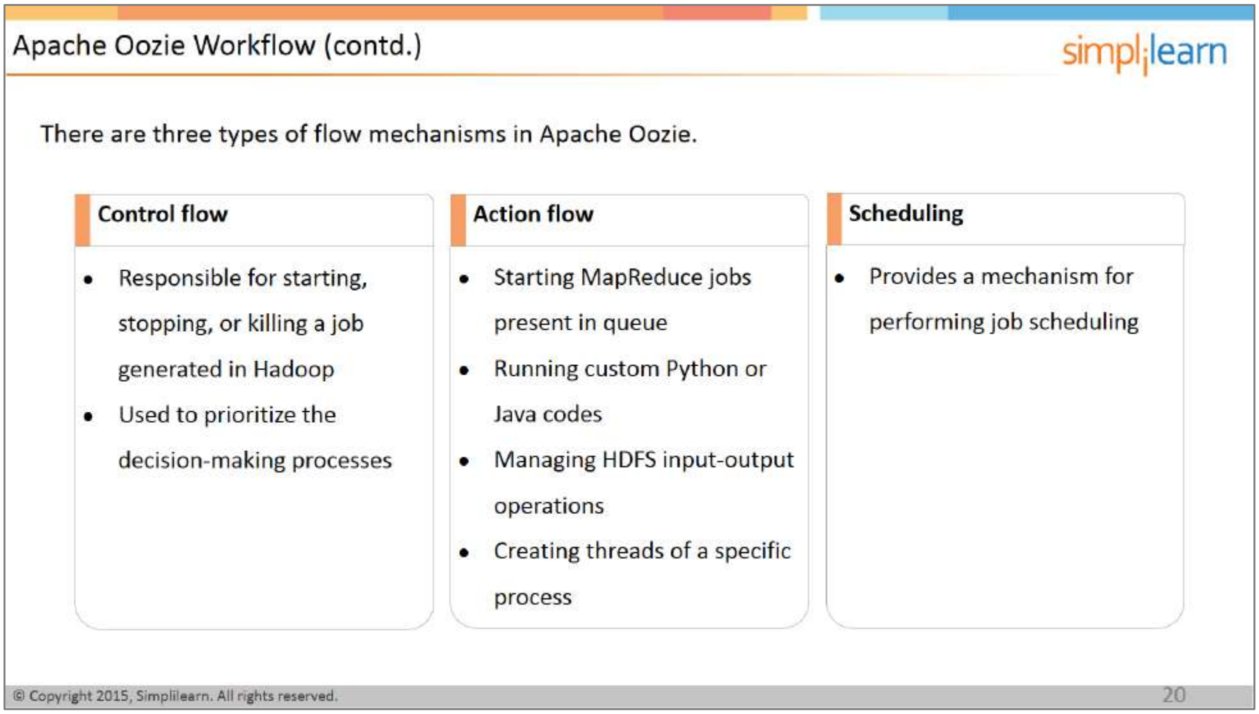
Apache Oozie

Apache Oozie is a workflow scheduler system used to manage Hadoop MapReduce jobs. It executes and monitors workflows in Hadoop. The workflow scheduler provides an option to its users to prioritize jobs based on their requirements. It also performs periodic scheduling of workflows. Furthermore, Oozie possesses the capability to trigger the execution of workflows based on data availability. It also provides a web and CLI.



Apache Oozie Workflow

The image depicts the basic mechanism of Oozie workflow. The x-axis represents the workflow section, and the y-axis displays the scheduling section. The structure is systematic so that the user can keep track of the flow. For instance, the first flow in which data is extracted from HDFS using Hive Query Language (HQL) and then stored in another location of HDFS. This data is then converted to structured data and stored in MySQL using Sqoop. This enables a user to create a systematic workflow for any type of Hadoop or Hadoop ecosystem related jobs.

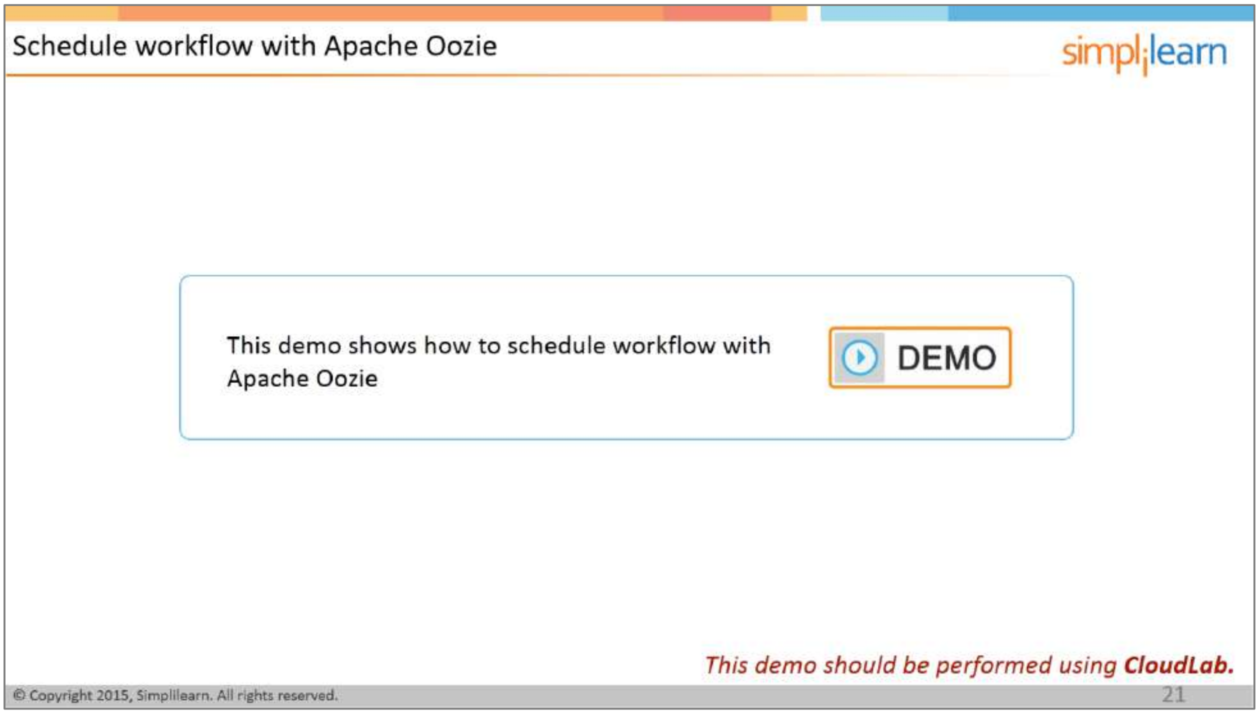


Apache Oozie Workflow (contd.)

Oozie has three types of flow mechanisms: control flow, action flow, and scheduling. The control flow is responsible for starting, stopping, or killing a job generated in Hadoop. It is also used for decision-making with respect to priority.

The action flow is responsible for starting the Hadoop MapReduce jobs that are present in the queue. It can also run custom Python or Java code and manage HDFS input-output operations. It is also used to create threads of a specific process.

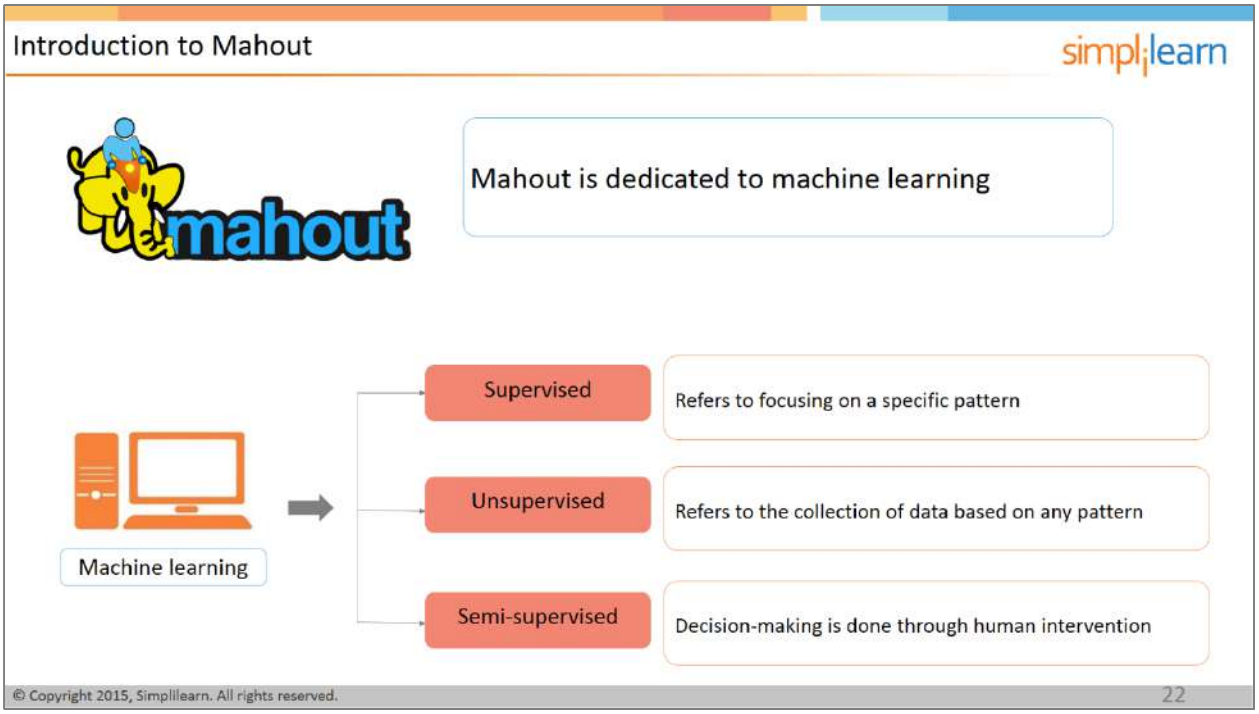
Scheduling pertains to providing a mechanism to schedule jobs.



Schedule workflow with Apache Oozie

Let us summarize the steps performed:

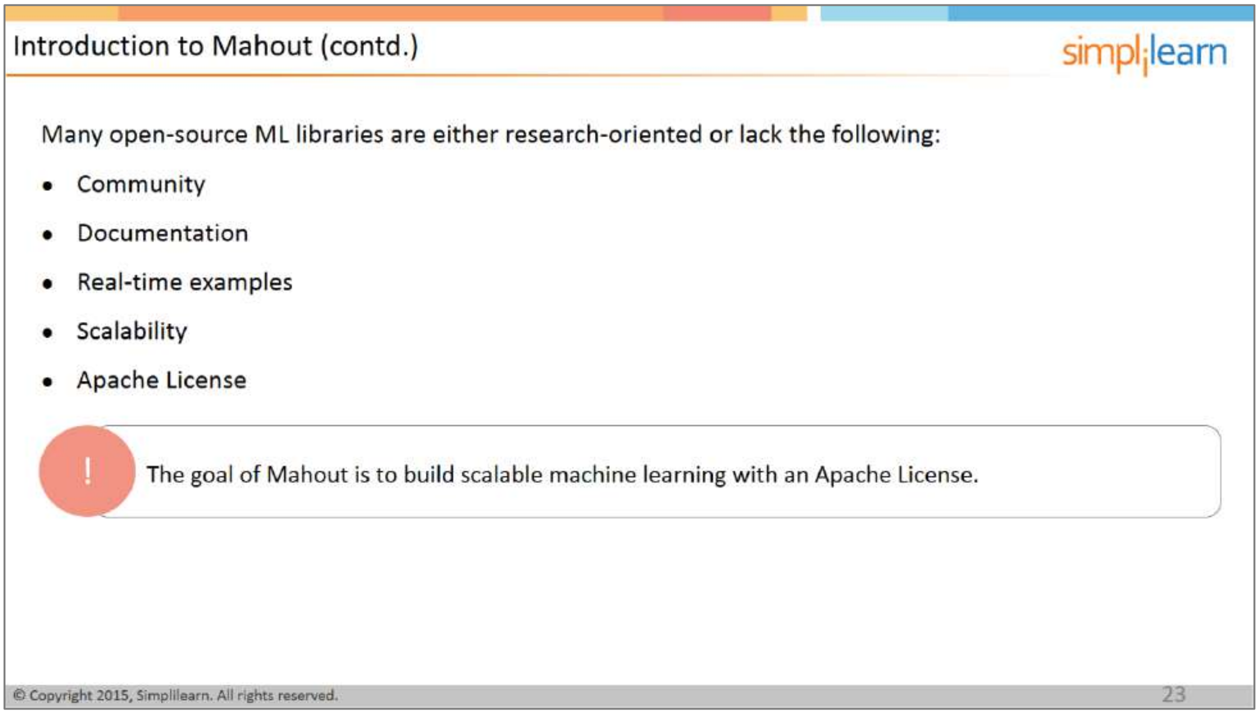
1. Change the path to the directory where Oozie is installed.
2. Extract the Oozie examples file.
3. To extract, use tar xvz command and extract in a directory named examples
4. Copy the examples directory to user's home directory in HDFS
5. Verify by using the command Hadoop fs –ls.
6. Launch an Oozie job now to run a MapReduce example job.
7. Notice the parameters after Oozie job command. Specify the Oozie URL. For instance, here we mention the IP address followed by 11000 port.
8. Oozie port may vary in your installation.
9. Notice a job ID is returned.
10. Let's check the status of this Oozie job now.
11. After the Oozie job command and URL mention info parameter and the job ID.
12. You will see that job information is returned.
13. To view Oozie status in Web Interface hit the URL mentioning IP address followed by Oozie port.
14. Double-click any job
15. Click Job DAG to view the job graph.



Introduction to Mahout

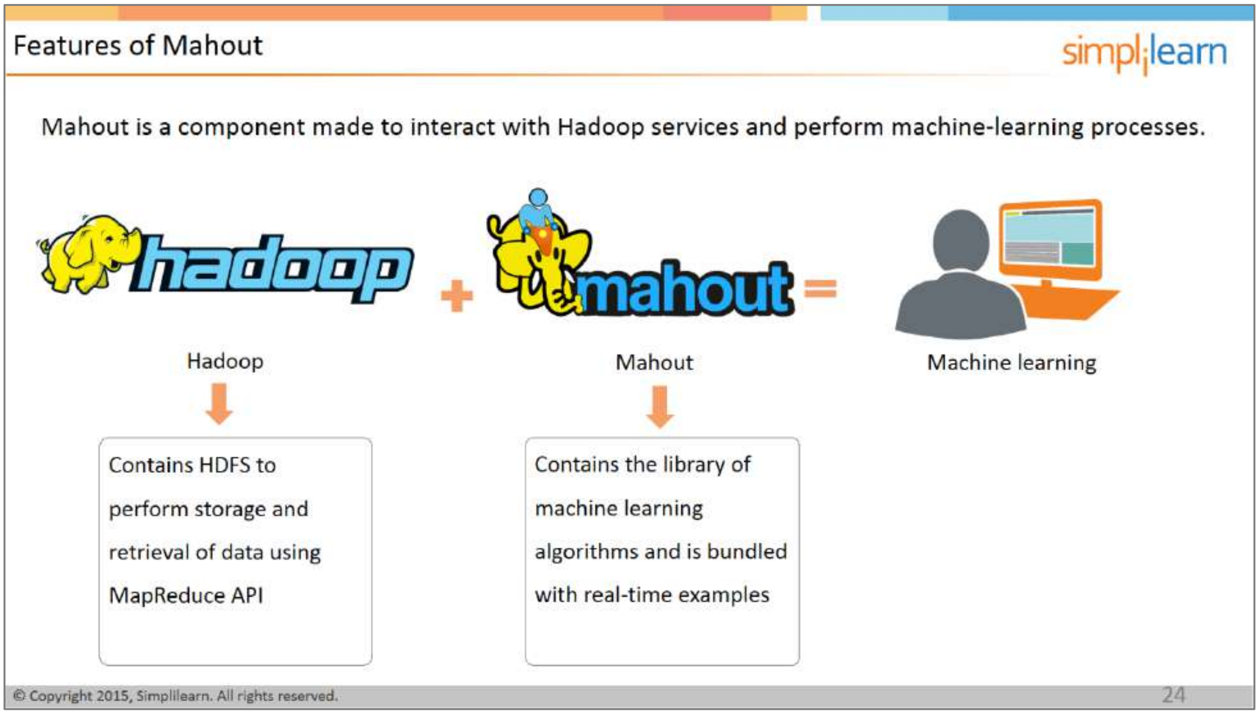
Mahout is an ecosystem component that is dedicated to machine learning. The machine learning process can be done in three modes: supervised, unsupervised and semi-supervised modes.

- The supervised mode refers to focusing on a specific pattern.
- The unsupervised mode refers to the collection of data based on any data pattern.
- The semi-supervised mode is similar to the supervised mode, with the difference being that decision-making is done through human intervention.



Introduction to Mahout (contd.)

Machine learning is a challenge because many open-source ML libraries are either research-oriented or they lack community, documentation, real-time examples, scalability, and an Apache License. The goal of Mahout is to build scalable machine learning libraries with an Apache License.



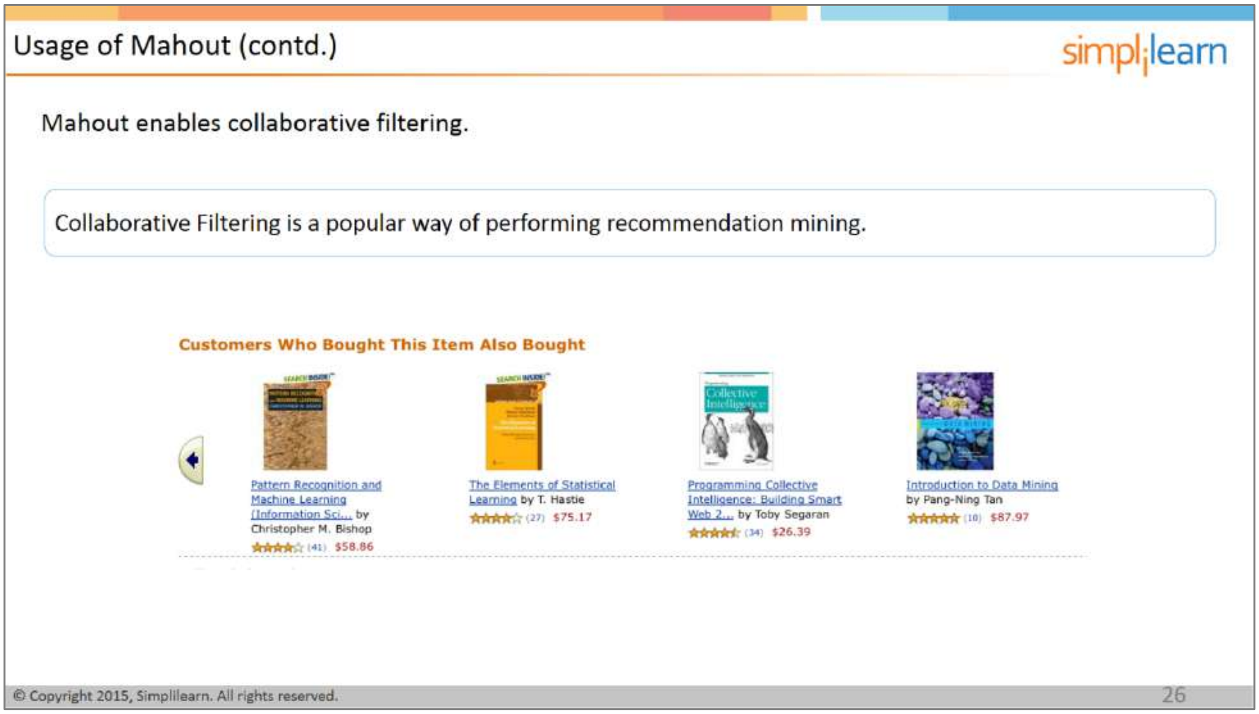
Features of Mahout

Mahout is a component made to interact with Hadoop services and perform machine-learning processes. Hadoop contains HDFS to perform storage and retrieval of data using MapReduce API, which provides scalability and fault tolerance in processing a job. Mahout contains the library of machine learning algorithms. It is bundled with real-time examples of these algorithms.

Usage of Mahout

Mahout helps in clustering, which is one of the most popular techniques of machine learning.

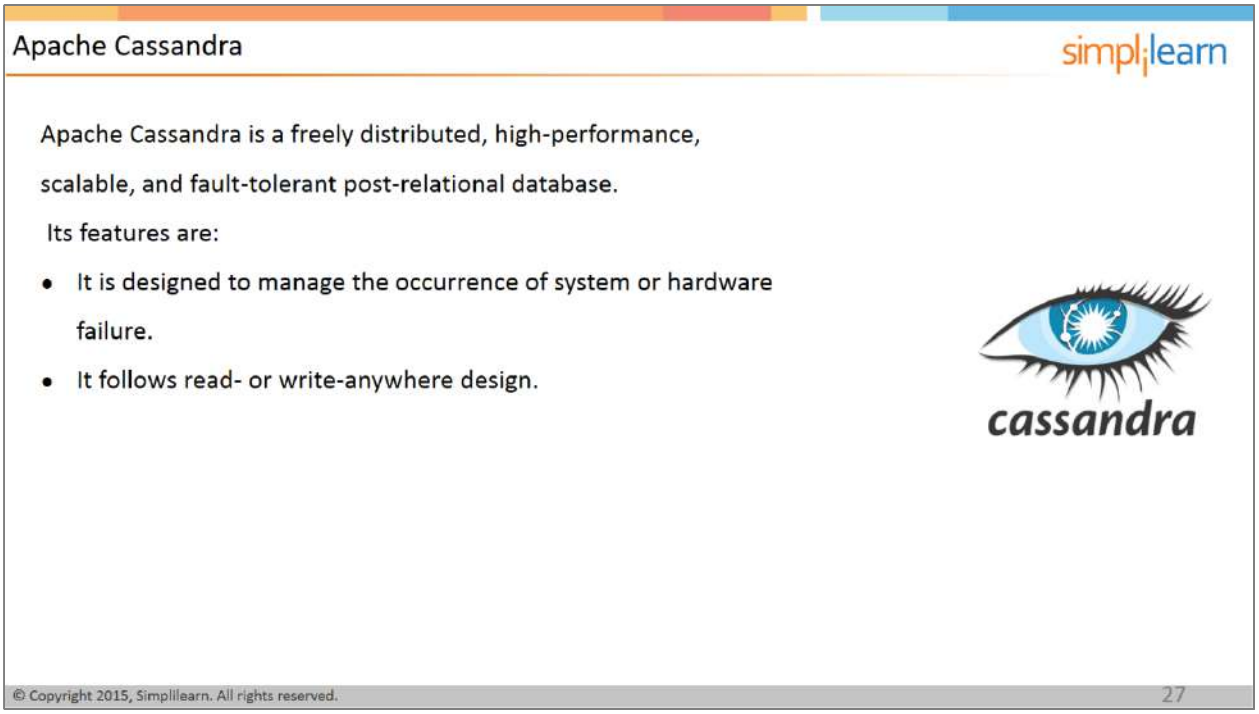
Clustering allows the system to group numerous entities into separate clusters or groups based on certain characteristics or features of the entities. One of the best examples of clustering is seen in Google News section.



Usage of Mahout (contd.)

Another concept of machine learning is Collaborative Filtering. Mahout enables Collaborative Filtering.

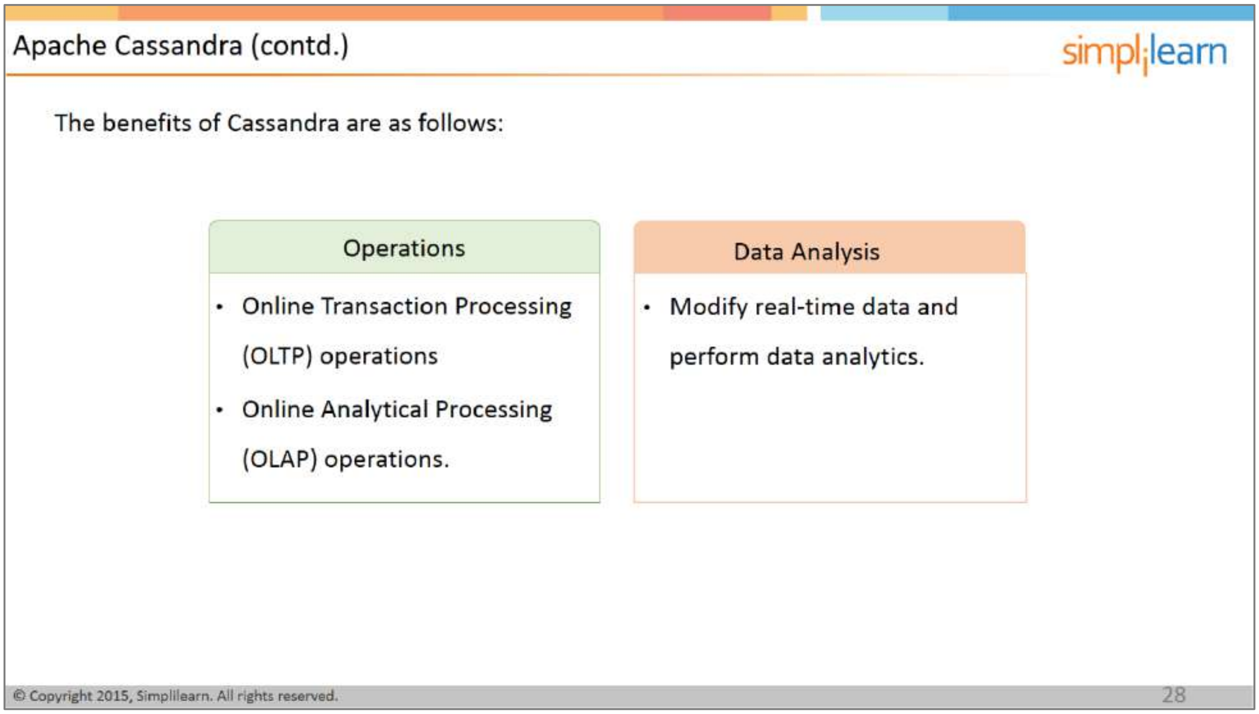
Collaborative Filtering is used where data mining is performed for recommendation engines. Examples of Collaborative Filtering can be found in shopping websites where data is displayed based on the user's browsing experience. One such example is shown on the image.



Apache Cassandra

Apache Cassandra is a freely distributed, high-performance, scalable, and fault-tolerant post-relational database. It possesses the following features:

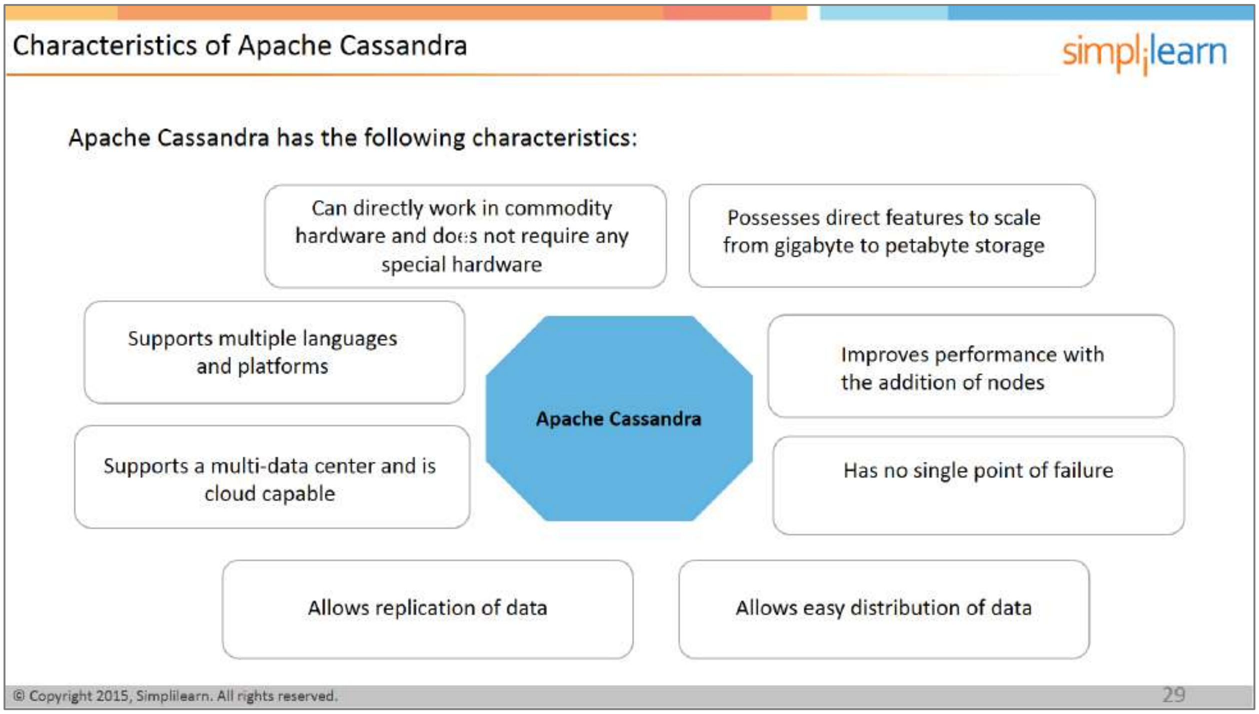
- It is designed with the awareness that system or hardware failures can occur.
- Cassandra follows read- or write-anywhere design, which makes it different from other ecosystem components.



Apache Cassandra (contd.)

The benefits of Cassandra are that:

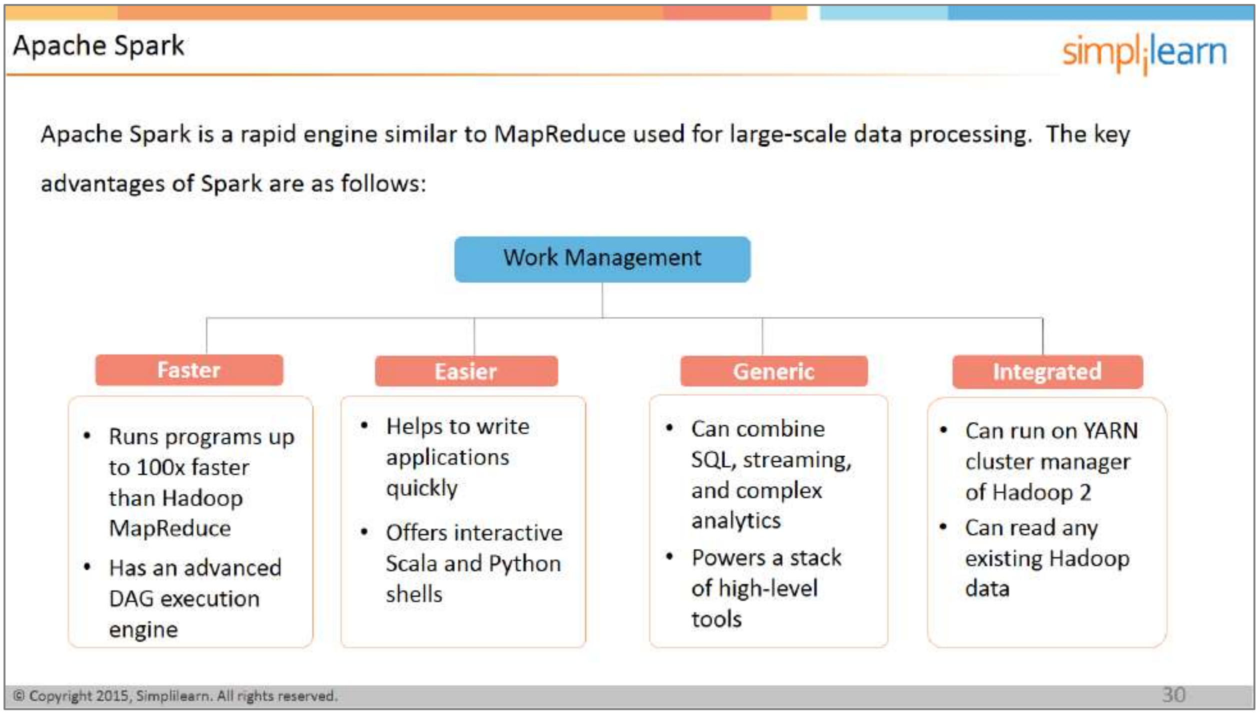
- it performs Online Transaction Processing (OLTP) operations and Online Analytical Processing (OLAP) operations and
- It helps to modify real-time data and perform data analytics.



Characteristics of Apache Cassandra

Apache Cassandra is used for the following reasons:

- It possesses direct features to scale from gigabyte to petabyte storage.
- It improves performance with the addition of nodes.
- It has no single point of failure.
- It allows easy distribution of data
- It allows replication of data.
- It supports a multi-data center and is cloud capable.
- It supports multiple languages and platforms.
- It can directly work in commodity hardware and does not require any special hardware.



Apache Spark

Apache Spark is a rapid engine similar to MapReduce that is used for large-scale data processing.

Following are the key advantages of Spark:

First, spark is faster.

- Spark claims to run programs up to 100 times faster than Hadoop MapReduce in memory or ten times faster on disk.
- Spark has an advanced DAG execution engine that supports cyclic data flow and in-memory computing.

Second, Spark is easier.

- It offers support to write applications quickly in Java, Scala, or Python.
- It offers interactive Scala and Python shells.

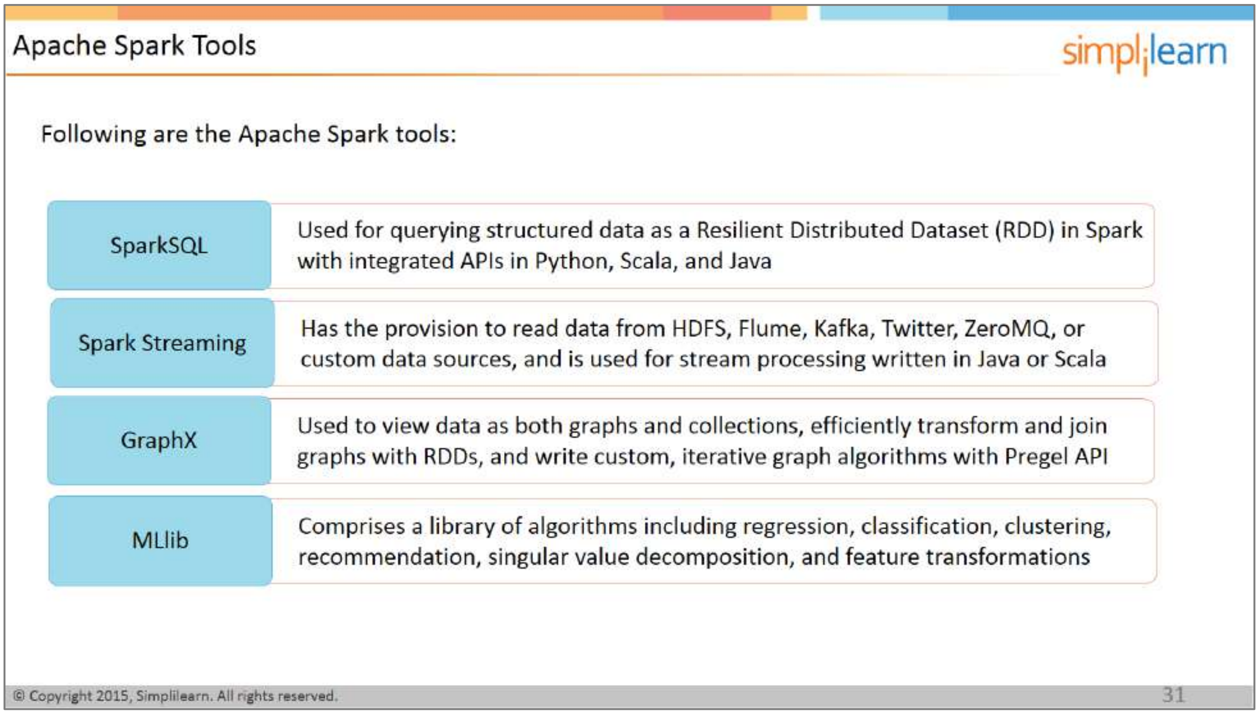
Third, Spark is generic.

- It can combine SQL, streaming, and complex analytics.

- It powers a stack of high-level tools including Spark SQL, MLlib for machine learning, GraphX, and Spark Streaming.

Fourth, Spark is integrated with Hadoop:

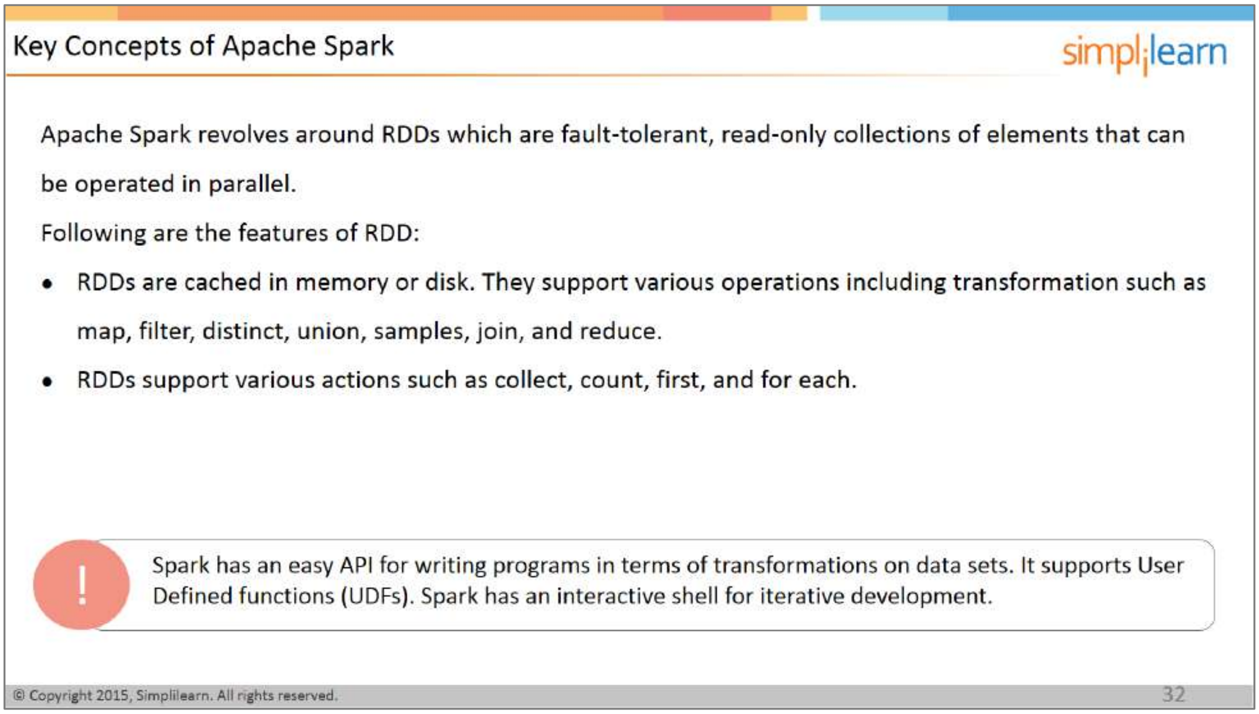
Spark can run on YARN cluster manager of Hadoop 2, and it can read any existing Hadoop data.



Apache Spark Tools

The key tools of Apache Spark are as follows:

- Spark SQL is used for querying structured data as a Resilient Distributed Data set (RDD) in Spark with integrated APIs in Python, Scala, and Java.
- Spark Streaming has the provision to read data from HDFS, Flume, Kafka, Twitter, ZeroMQ, or custom data sources, and it is used for stream processing written in Java or Scala.
- Graph X is used to view the data as both graphs and collections, to transform and join graphs with RDDs efficiently. It is also used to write custom iterative graph algorithms with the Pregel API.
- MLlib is comprised of a library of algorithms including regression, classification, clustering, recommendation, singular value decomposition, and feature transformations.



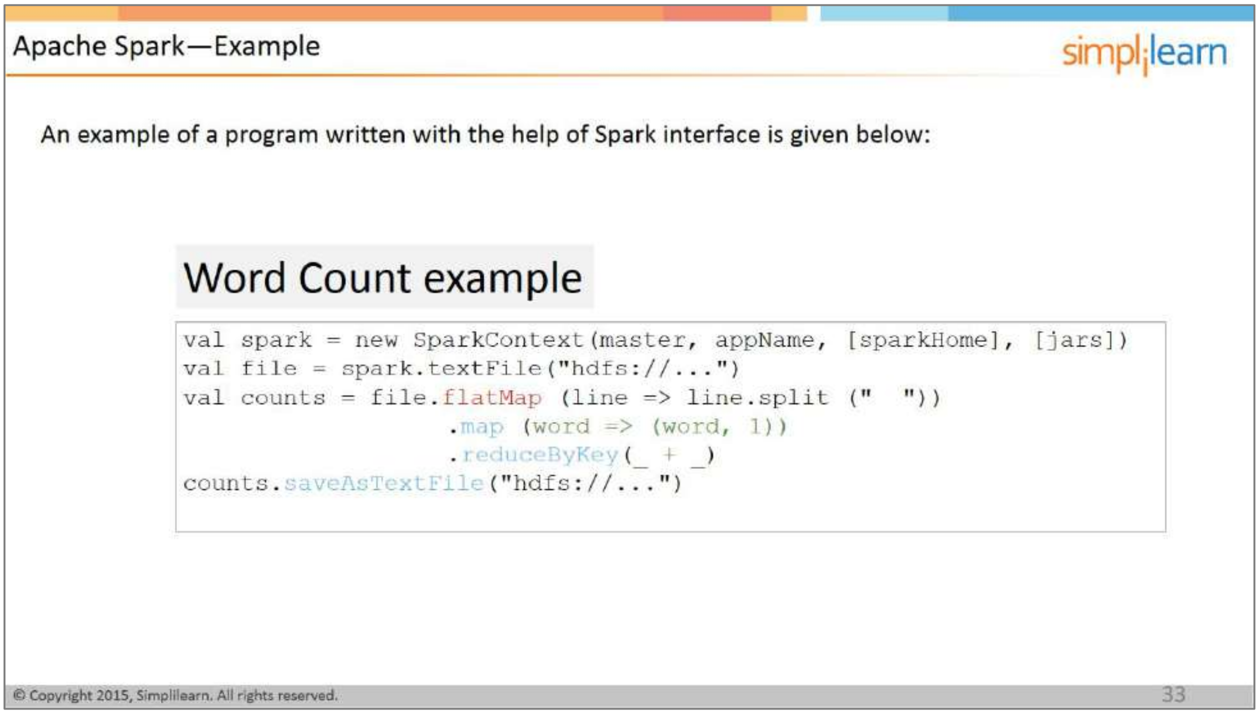
Key Concepts Related to Apache Spark

Apache Spark revolves around Resilient Distributed Data sets (RDD) which are fault-tolerant, read-only collections of elements that can be operated in parallel.

Following are the features of RDDs:

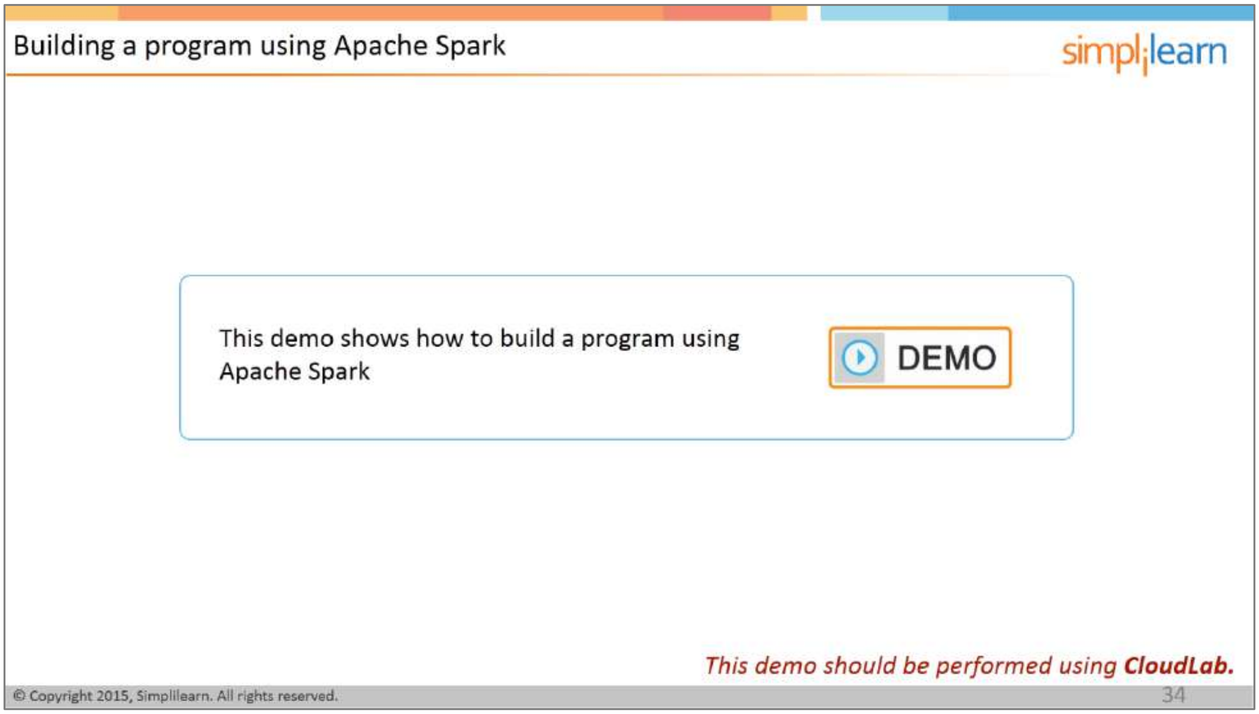
- RDDs are cached in memory or disk. They support various operations including transformation such as map, filter, distinct, union, samples, join, and reduce.
- RDDs also support actions such as collect, count first for each, and others.

Spark has an easy API for writing programs in terms of transformations on data sets. It supports User Defined Functions (r UDFs). Spark has an interactive shell for iterative development.



Apache Spark—Example

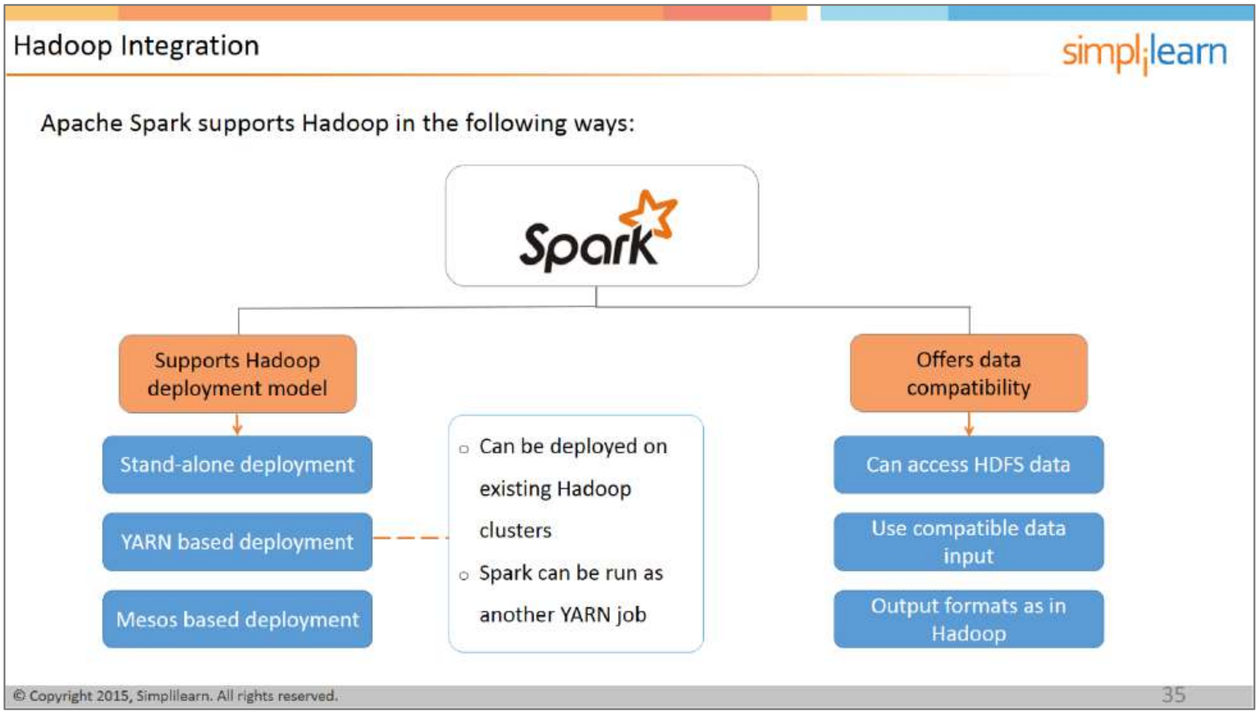
An example of a program written with the help of Spark interface is shown here.



Building a program using Apache Spark

Let us summarize the steps performed:

1. Let us look at how to run a Spark job in Spark shell. If Spark is already installed in your machine, type spark shell on command prompt.
2. You will notice scala prompt once Spark Shell is launched.
3. We will run a Word Count example here in Spark. Assuming an e-book by the name '4300.txt' already exist in your HDFS, type the command shown.
4. Here 'sc' is the Spark context and we have just created a RDD by the name file.
5. In the command shown, we are now applying map and reduce transformation in the 'file' RDD.
6. Use the command counts.count() to count the total number of words in the e-book.
7. Notice the result showing total number of words in the file.
8. Let's now print word wise count.
9. Notice the results in key value pairs.
10. Here we are able to run a word count job using simple Spark Scala program while leveraging Spark API.



Hadoop Integration

Apache Spark supports the Hadoop deployment model. Some of the deployment models include stand-alone deployment, YARN-based deployment, and Mesos-based deployment. Spark can be deployed on existing Hadoop clusters, and it can also run as simply another YARN job.

Spark offers data compatibility to Hadoop. It can access HDFS data and use compatible data input and output formats as in Hadoop. Spark leverages localized data processing.

Spot the Error

Carol was assigned the task of writing a program to count the number of words in an e-book. After finding the word count, the output should be printed. Carol was unable to achieve the result. Help her identify the error.

Spot the incorrect code.

```
[ hadoop@simplilearn ] #
[ hadoop@simplilearn ] # spark-shell
Spark context available as sc.
15/10/19 22:55:22 INFO SparkILoop: Created sql context (with Hive support)..
SQL context available as sqlContext.

scala> val file = sc.textFile("hdfs://127.0.0.1:9000/sdata/4300.txt")

scala> counts.toArray().foreach println
```


Spot the Error

The incorrect code is: `scala> counts.toArray().foreach println`

The correct syntax is: `scala> counts.toArray().foreach(println)`

QUIZ 1

Which of the following options can be used as Hadoop dedicated databases for OLTP and OLAP?

a. Hive

b. HBase

c. Mahout

d. Cassandra

 40

QUIZ 2

Which of the following distributions is dedicated to machine learning?

a. Hive

b. HBase

c. Mahout

d. Cassandra

 42

QUIZ 3

Which of the following options is NOT a Chukwa entity?

a. Agent

b. Collector

c. Sink

d. MapReduce

 44

QUIZ 4

What type of scalability is supported by Apache Flume?

a. Horizontal scalability

b. Vertical scalability

c. Diagonal scalability

d. No scalability

 46

simplilearn

QUIZ 5

Which of the following issues can be addressed by MapReduce v2 (MRv2/YARN)?

a. A single point of failure in the NameNode

b. Improving utilization of resources

c. HDFS latency

d. Standardize on a single MapReduce API

48

simplilearn

QUIZ 6

Identify the MapReduce v2 (MRv2/YARN) daemon responsible for managing resources required by the NameNode for performing a specific job.

a. ResourceManager

b. NodeManager

c. ApplicationMaster

d. TaskTracker

50

QUIZ 7

Which of the following options is included in the workflows in Oozie?

a. Sequence of MapReduce and Pig

b. Sequence of MapReduce jobs only

c. Sequence of MapReduce and Pig jobs

d. Alternative repetition of MapReduce jobs until a desired answer or state is reached

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which of the following options can be used as Hadoop dedicated databases for OLTP and OLAP?	Apache Cassandra can be used for OLAP and OLTP operations on Hadoop.
2	Which of the following distributions is dedicated to machine learning?	Apache Mahout is used for machine learning.
3	Which of the following options is NOT a Chukwa entity?	MapReduce does not come under the Chukwa entities. It is a Hadoop entity.
4	What type of scalability is supported by Apache Flume?	Apache Flume supports diagonal scalability.
5	Which of the following issues can be addressed by MapReduce v2 (MRv2/YARN)?	YARN helps in improving the utilization of resources.
6	Identify the MapReduce v2 (MRv2/YARN) daemon responsible for managing resources required by the NameNode for performing a specific job.	ResourceManager is responsible for managing resources required by the NameNode for performing a specific job.
7	Which of the following options is included in the workflows in Oozie?	Sequences of MapReduce and Pig can be combined with other actions including forks, decision points, and path joins.

Case Study

Scenario | Analysis | Solution

XY Networks provides network security support to many organizations. It has set up a cluster of servers, and created an in-house application that monitors these clusters. It collects System Log files from numerous customers in real-time, and stores these files in Hadoop for analytics processing. While Hadoop is good for storage of large files and analytics, a faster response is required for some analytics. The company's IT team has heard that Spark provides much faster access to Big Data than Hadoop.

Case Study

simplilearn

Scenario | Analysis | Solution

The IT team does research on Spark, and finds that it is more expressive than Hadoop. It provides more functions than MapReduce. They found that Spark is 10 times faster than Hadoop in log processing. They also found that Spark provides SQL interface that can read Hive tables faster. Spark also has machine learning libraries that can be used for analytics.

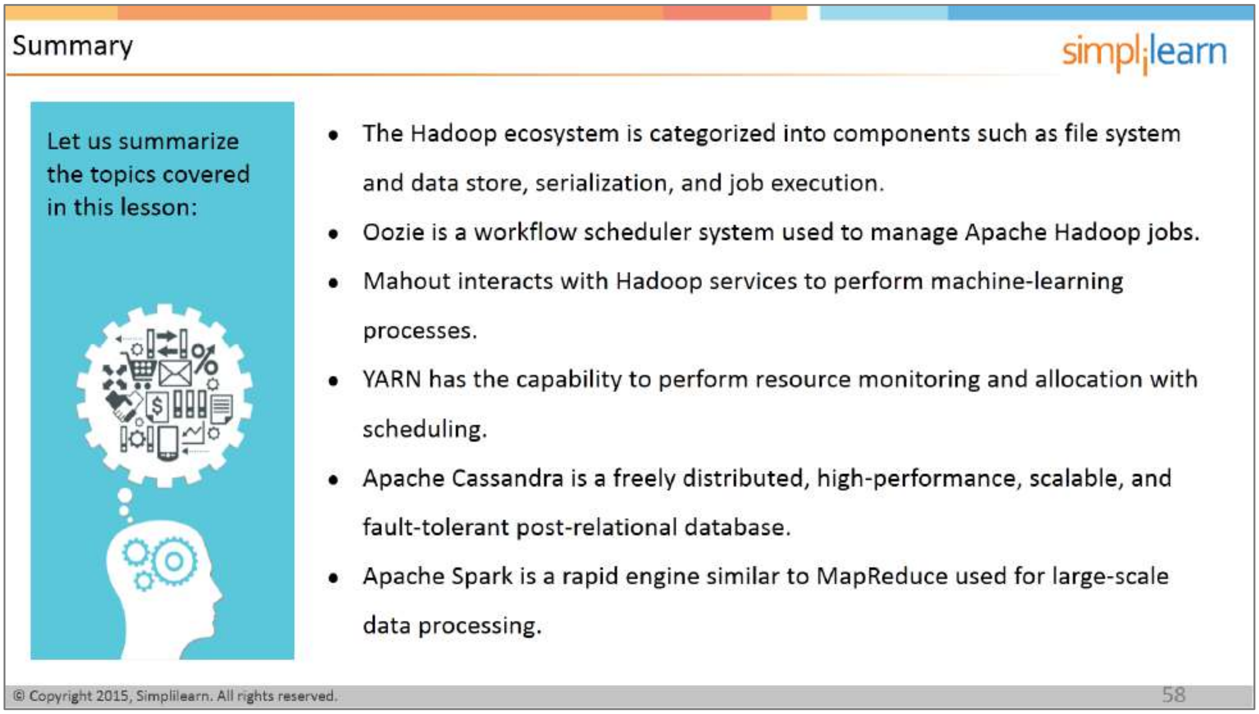
1. It can use HDFS for storage.
2. It is horizontally scalable. They can add new machines to the cluster any time to increase the capacity.
3. Provides SQL interface, and connects to Hive metastore.
4. Provides machine learning and graph processing libraries.

Case Study

Scenario | Analysis | Solution

Perform the following steps to explore Spark:

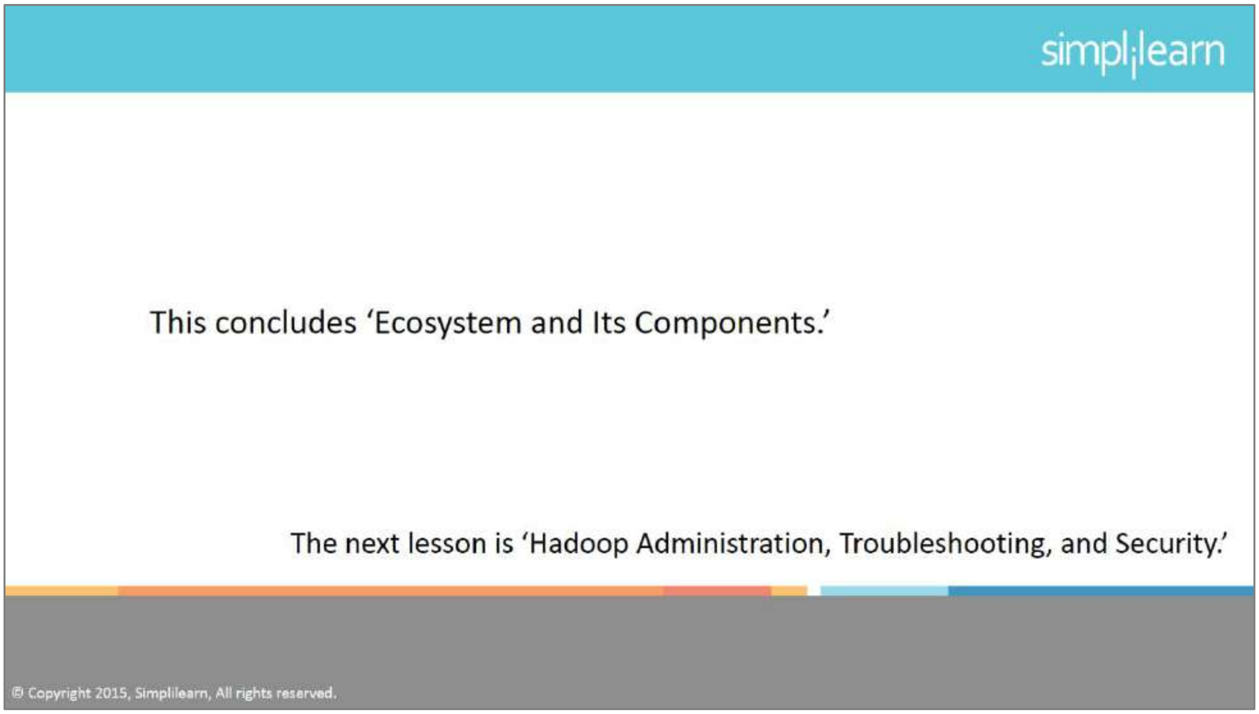
1. Read the log files from HDFS.
2. Filter the required log types.
3. Aggregate the count for each log type.
4. Store the results back to HDFS.
5. Check the web interface for monitoring.



Summary

Let us summarize the topics covered in this lesson:

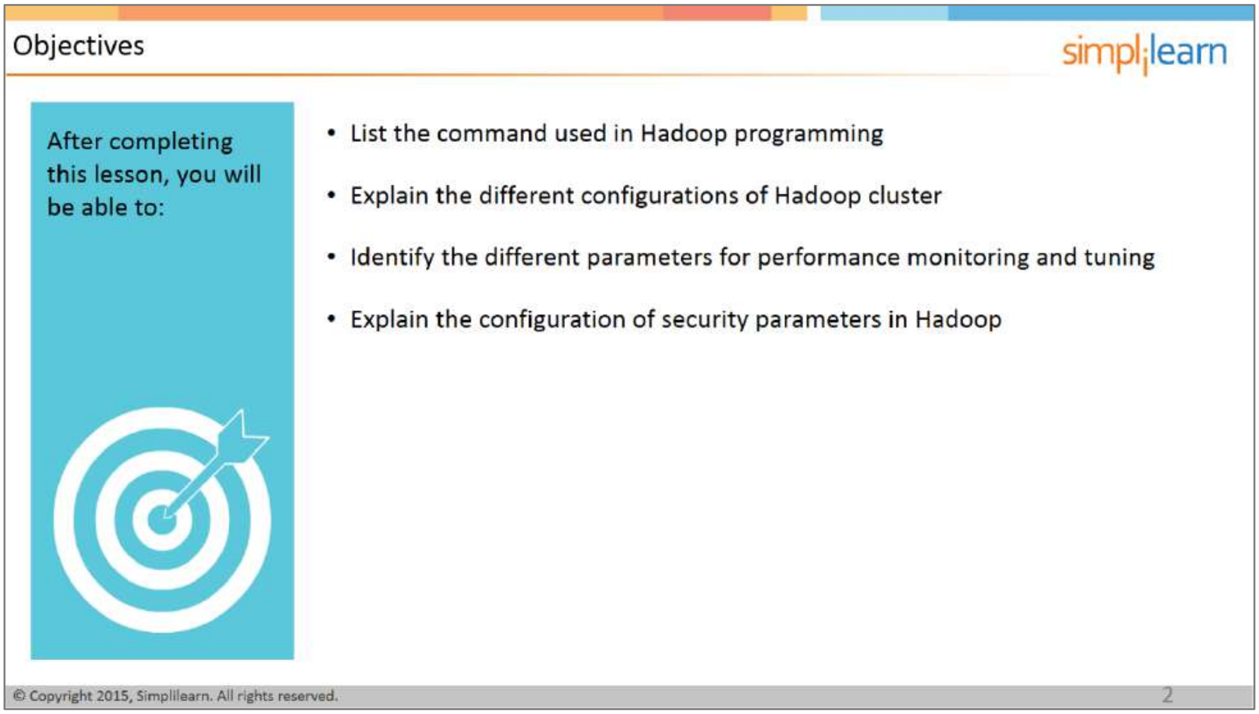
- The Hadoop ecosystem is categorized into components such as file system and data store, serialization, and job execution.
- Oozie is a workflow scheduler system used to manage Apache Hadoop jobs (MapReduce jobs).
- Mahout interacts with Hadoop services to perform machine-learning processes.
- YARN possesses the capability to perform resource monitoring and allocation with scheduling.
- Apache Cassandra is a freely distributed, high-performance, scalable, and fault-tolerant, post-relational database.
- Apache Spark is a rapid engine similar to MapReduce used for large-scale data processing.



Conclusion

This concludes 'Hadoop Ecosystem and its Components.' In the next lesson, we will focus on 'Hadoop Administration, Troubleshooting, and Security.'

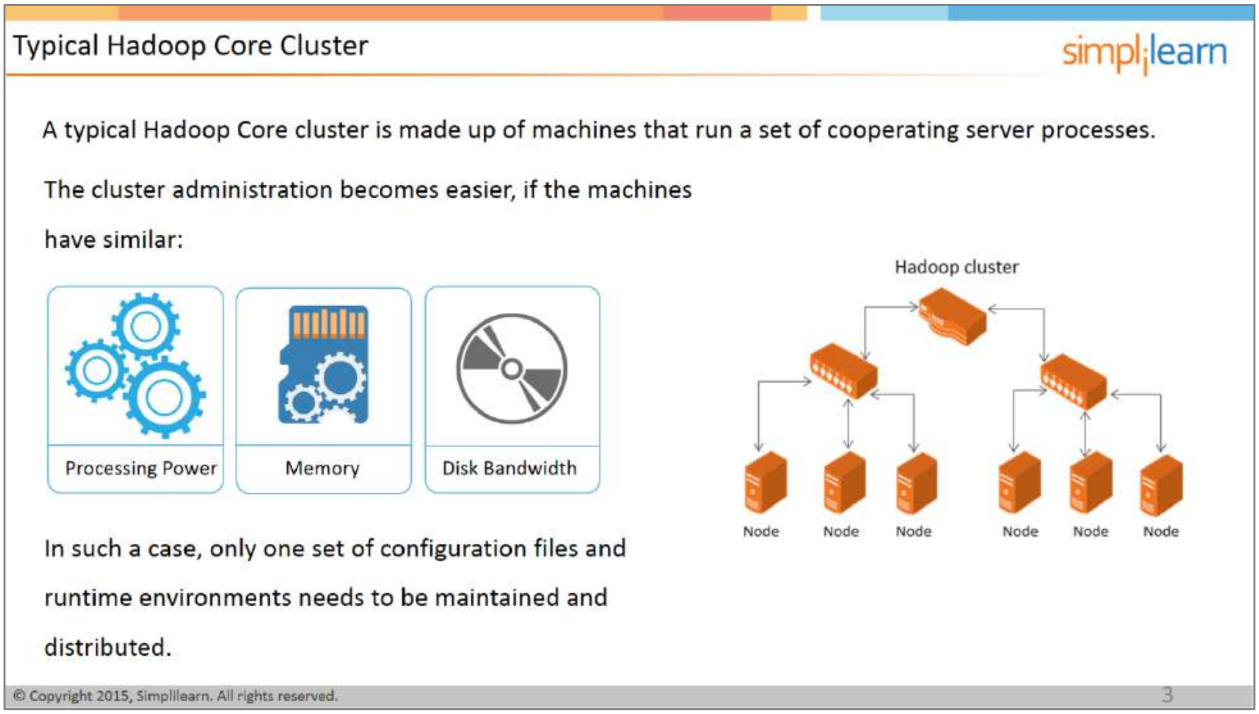
Lesson 12—Hadoop Administration, Troubleshooting, and Security



Objectives

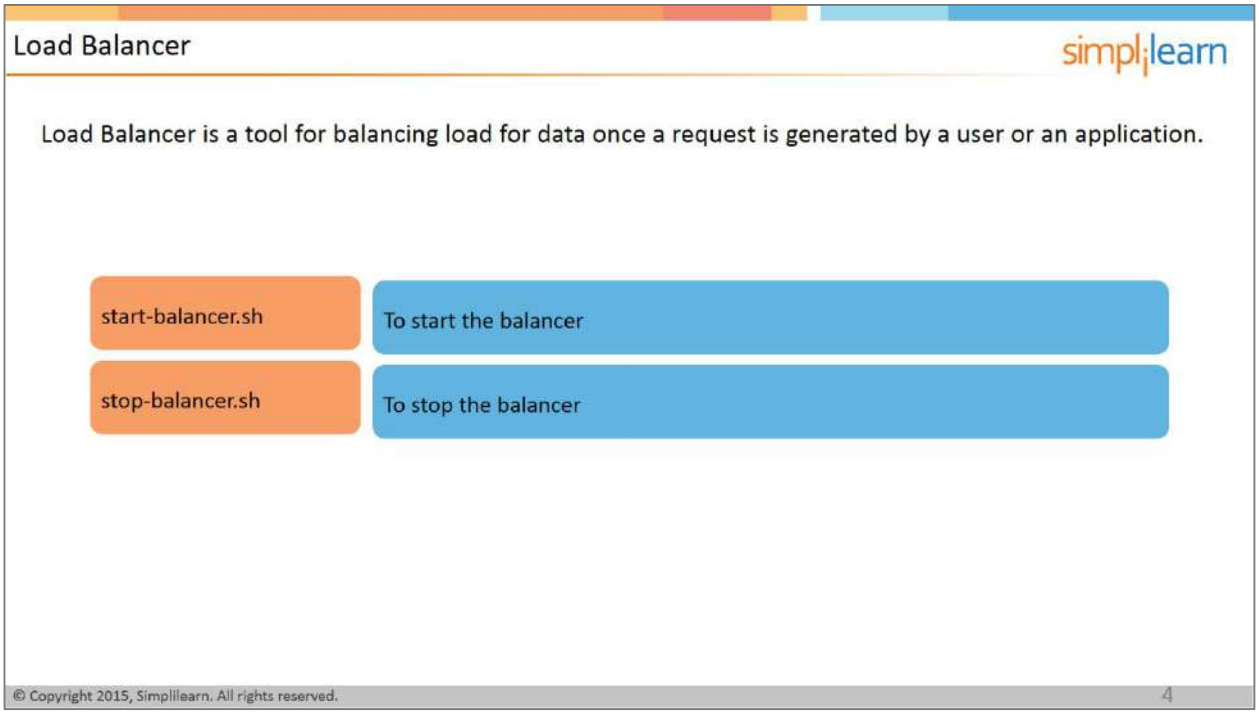
After completing this lesson, you will be able to:

- List the commands used in Hadoop programming
- Explain the different configurations of Hadoop cluster
- Identify the different parameters for performance monitoring and tuning
- Explain the configuration of security parameters in Hadoop



Typical Hadoop Core Cluster

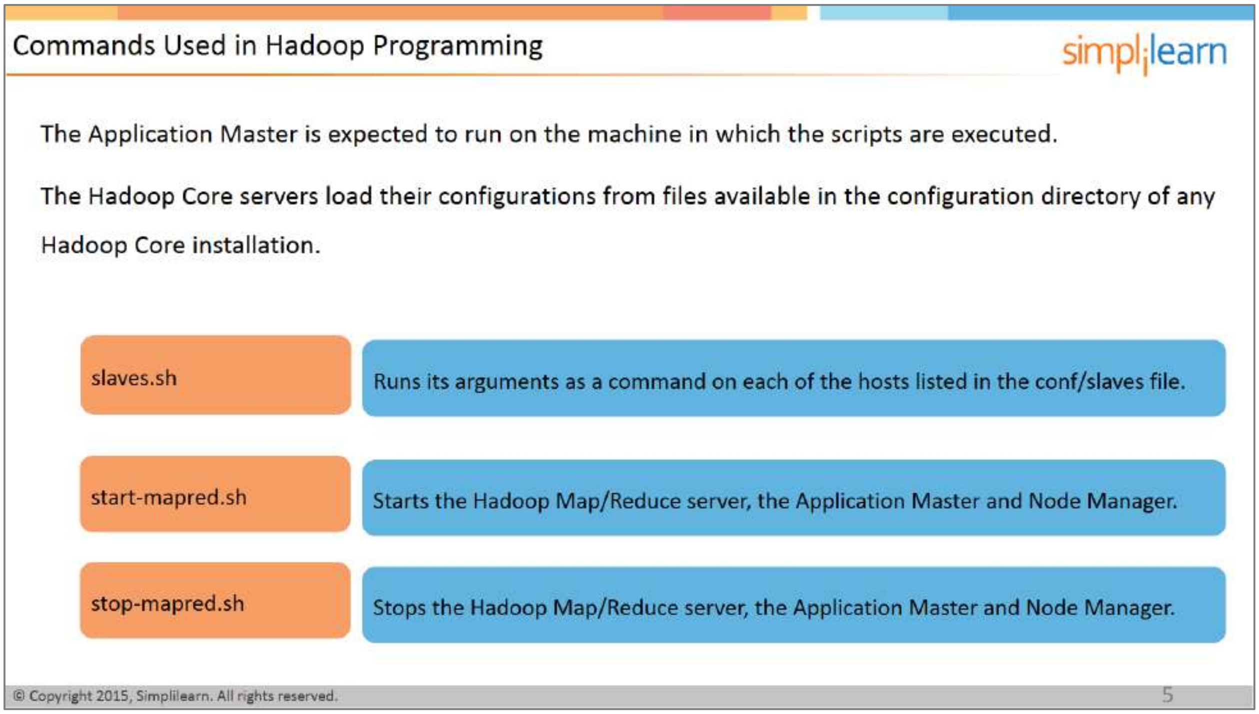
A typical Hadoop Core cluster is composed of machines that run a set of cooperating server processes. Machines in the cluster are not required to be homogeneous. If the machines have similar processing power, memory, and disk bandwidth, the cluster administration becomes easier. In such a case, only one set of configuration files and runtime environments needs to be maintained and distributed.



Load Balancer

The Hadoop has to balance the huge data requested by a user or an application. This balancing of data load is performed using Load balancer tool.

Use ‘start-balancer.sh’ to start the balancer and ‘stop-balancer.sh’ to stop the balancer.



Commands Used in Hadoop Programming

The Application Master is expected to run on the machine in which the scripts are executed.

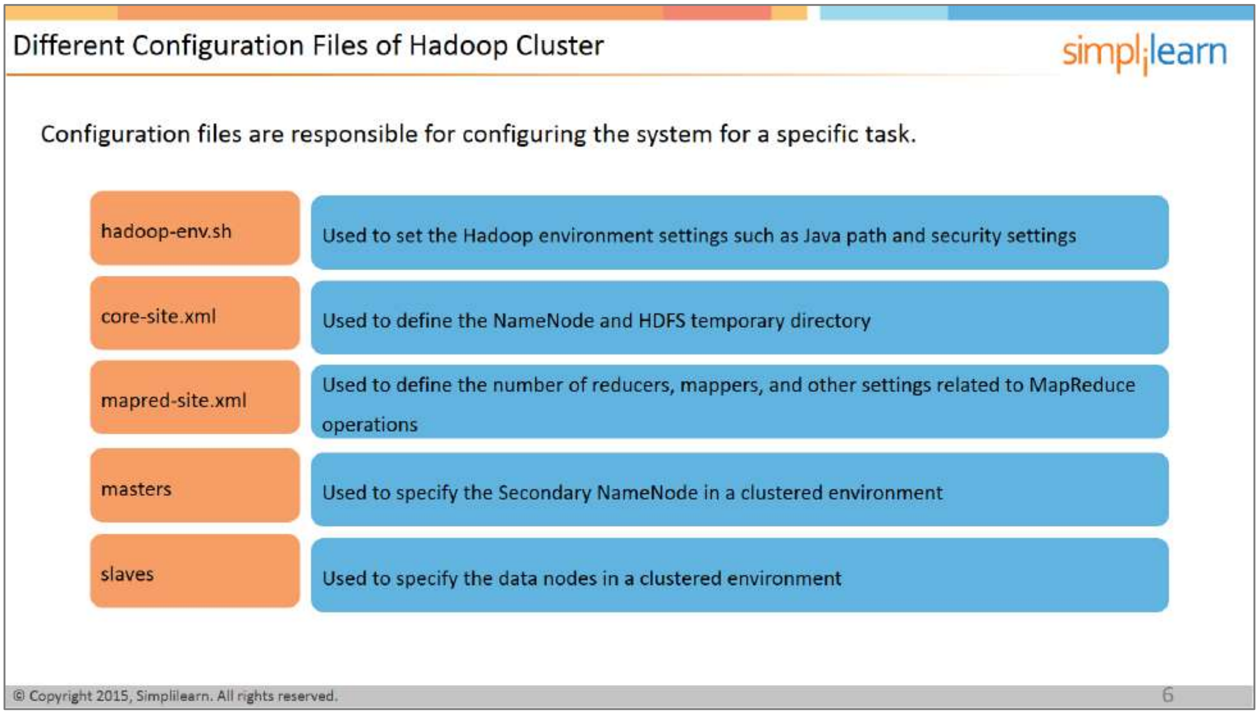
The Hadoop Core servers load their configurations from files available in the configuration directory of any Hadoop Core installation.

Let’s discuss some of the commands used in Hadoop programming.

slaves.sh runs its arguments on each of the hosts listed in the conf/slaves file.

start-mapred.sh starts the Hadoop MapReduce server, the Application Master, and Node Manager.

stop-mapred.sh stops the Hadoop MapReduce server, the Application Master, and Node Manager.

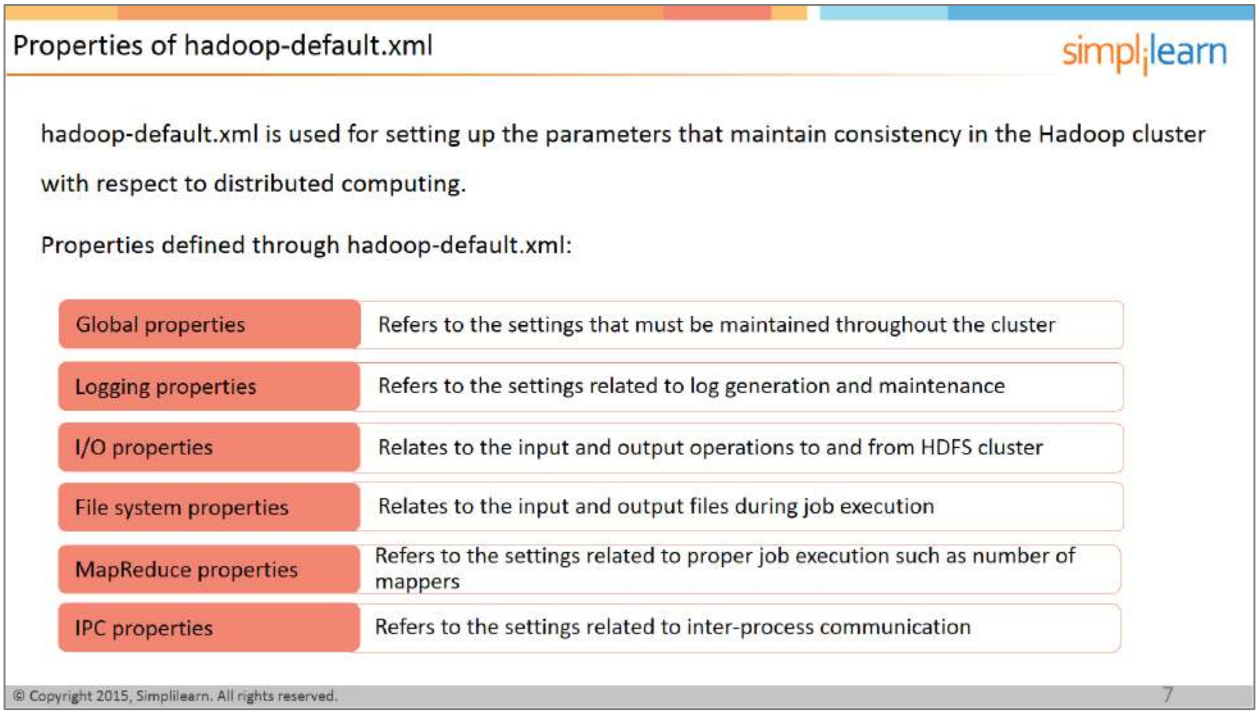


Different Configuration Files of a Hadoop Cluster

Configuration files are responsible for configuring the system for a specific task.

Following are the configuration files of a Hadoop cluster:

- hadoop-env.sh sets Hadoop environment settings such as Java path and security settings
- core-site.xml defines the NameNode and the HDFS temporary directory
- mapred-site.xml defines the number of reducers, mappers, and other settings related to MapReduce operations
- masters specifies the Secondary NameNode in a clustered environment
- slaves specifies the DataNodes in a clustered environment



Properties of Hadoop-default.xml

hadoop-default.xml is used for setting up the parameters that maintain consistency in the Hadoop cluster with respect to distributed computing.

Following are the properties defined through hadoop-default.xml:

- Global,
- Logging,
- I/O,
- File system,
- MapReduce, and
- IPC properties.

Global properties refer to the settings that must be maintained throughout the cluster.

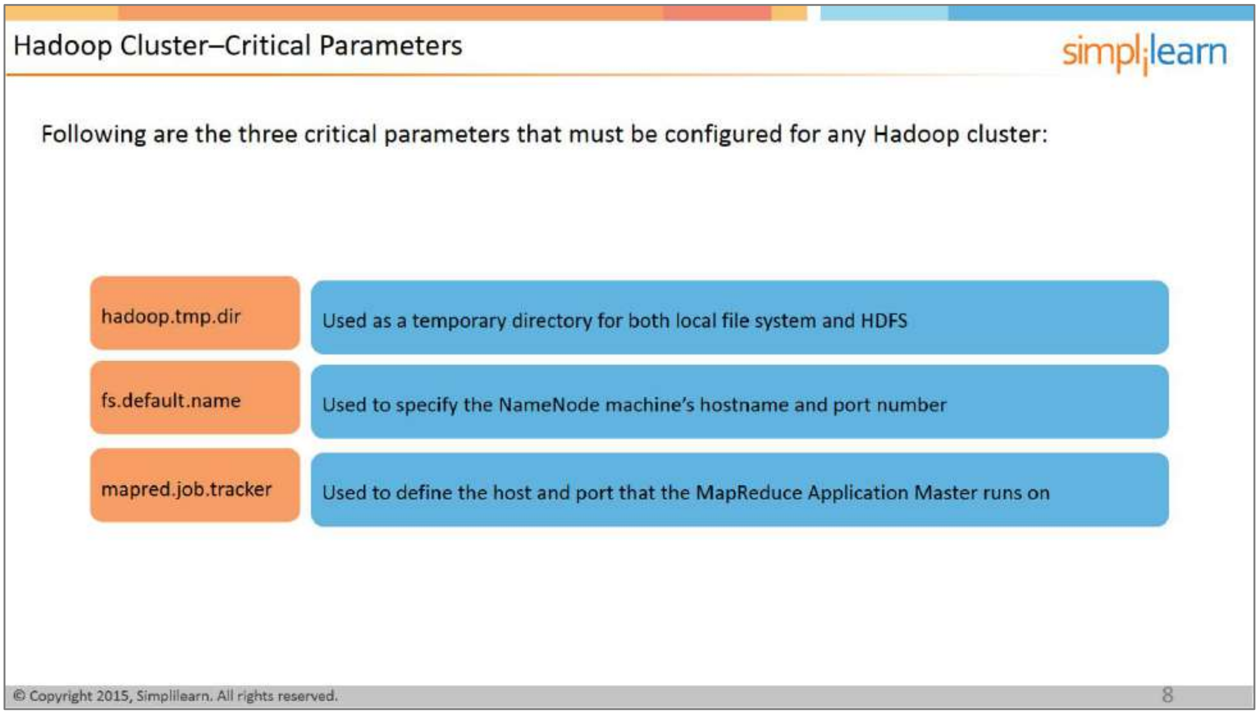
Logging properties refer to the settings related to log generation and maintenance.

I/O properties relate to the input and output operations to and from an HDFS cluster.

File system properties relate to the input and output files during job execution.

MapReduce properties refer to the settings related to proper job execution such as the number of mappers.

IPC properties refer to the settings related to inter-process communication.



Hadoop Cluster– Critical Parameters

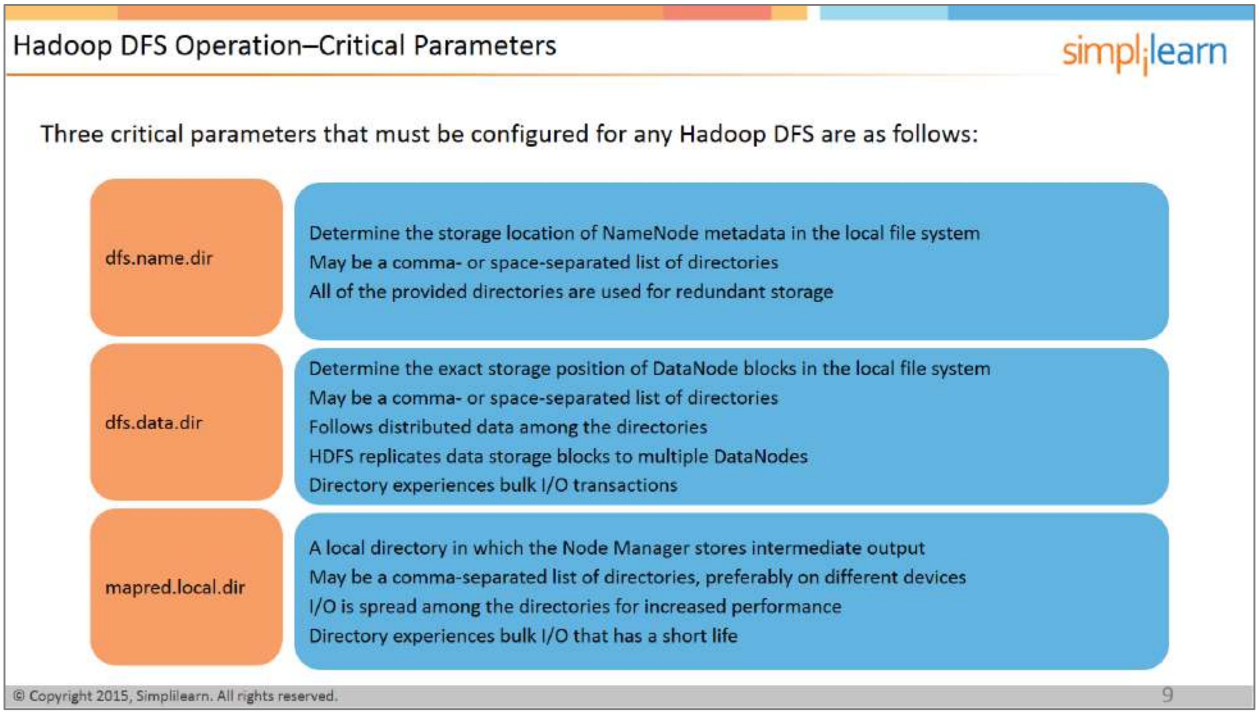
Let's now look at the critical parameters that must be configured for any Hadoop cluster and DFS operation.

The three critical parameters that must be configured for any Hadoop cluster are as follows.

The hadoop.tmp.dir parameter is used as a temporary directory for both local file system and HDFS.

The fs.default.name parameter is used to specify the NameNode machine's hostname and port number.

The mapred.job.tracker parameter is used to define the host and port on which MapReduce Application Master runs.



Hadoop DFS Operation– Critical Parameters

The three critical parameters to be configured for any Hadoop DFS operation are as follows.

dfs.name.dir determines the storage location of NameNode metadata in the local file system. It may be a comma- or space-separated list of directories. All the provided directories are used for redundant storage.

dfs.data.dir determines the exact storage position of DataNode blocks in the local file system. It may be a comma- or space- separated list of directories. It follows distributed data among the directories. HDFS replicates data storage blocks to multiple DataNodes. This directory experiences bulk I/O transactions.

mapred.local.dir is a local directory in which the Node Manager stores intermediate output. It may be a comma-separated list of directories, preferably on different devices. The I/O is spread among the directories for increased performance. The directory experiences bulk short life I/O.

Port Numbers for Individual Hadoop Services

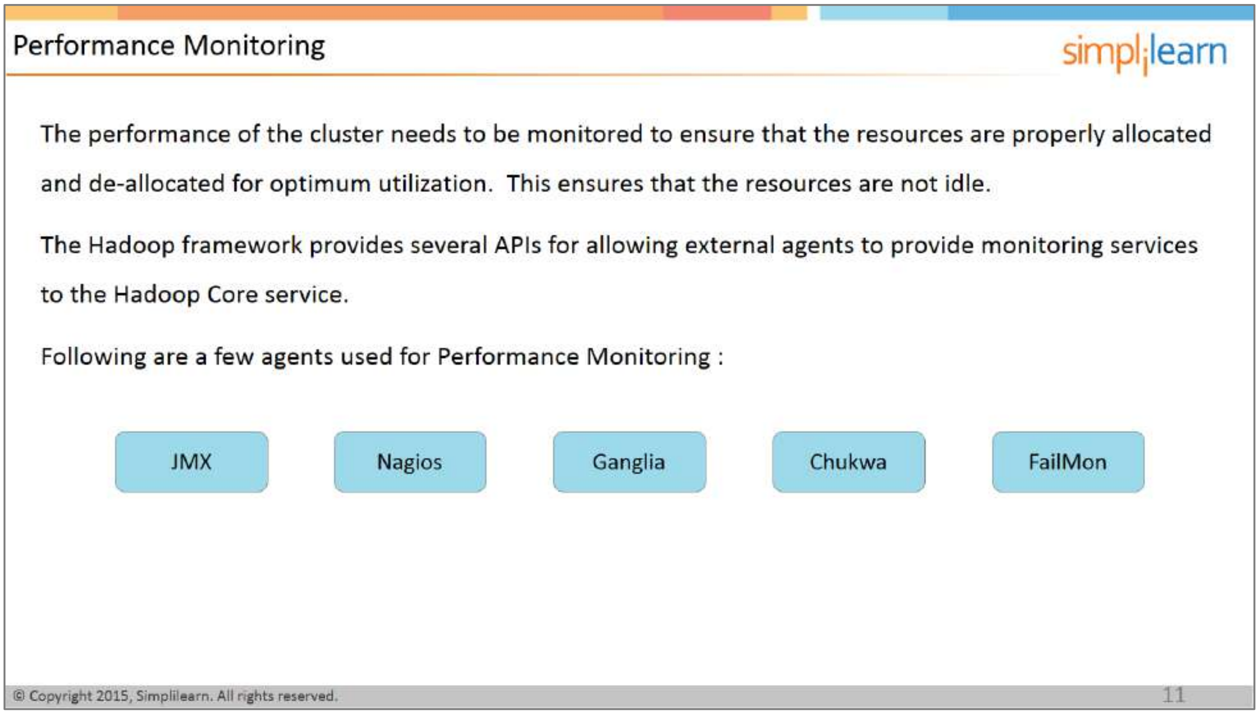
Port numbers for individual Hadoop services can be classified as follows:

Port Number	Name of the Parameter	Explanation for the Parameter
50030	mapred.job.tracker.http.address	Application Master administrative web GUI
50070	dfs.http.address	Node Manager administrative web GUI
50010	dfs.datanode.address	DataNode control port
50020	dfs.datanode.ipc.address	DataNode IPC port, used for block transfer
50060	mapred.task.tracker.http	address Per Node Manager web interface
50075	dfs.datanode.http	address Per DataNode web interface
50090	dfs.secondary.http	address Per Secondary NameNode web interface
50470	dfs.https	address NameNode web GUI via HTTPS
50475	dfs.datanode.https	address Per DataNode web GUI via HTTPS

Port Numbers for Individual Hadoop Services

The table shows the individual port numbers for specific services that can be accessed via the NameNode IP.

Please note that these ports may vary in different commercial distributions.

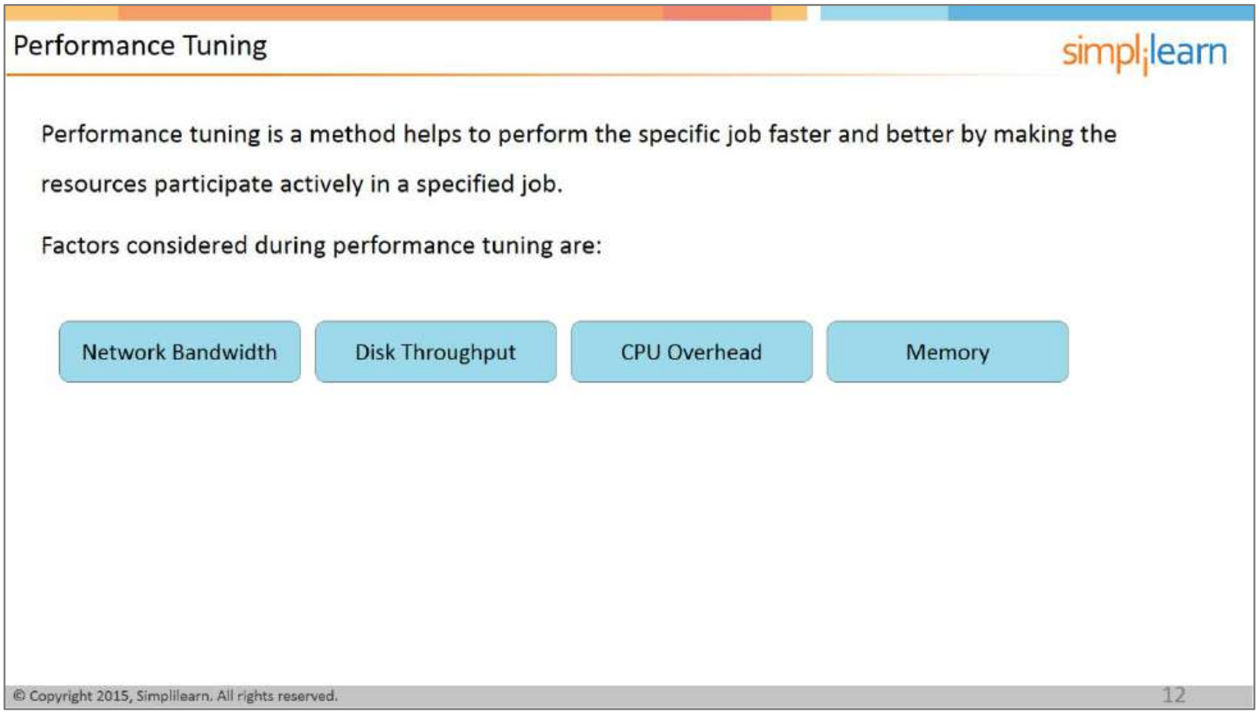


Performance Monitoring

The performance of a cluster needs to be monitored to ensure that the resources are properly allocated and de-allocated for optimum utilization. This ensures that the resources are not idle.

The Hadoop framework provides several APIs for allowing external agents to provide monitoring services to the Hadoop Core service.

Some agents used for Performance Monitoring are JMX, Nagios, Ganglia, Chukwa, and FailMon.



Performance Tuning

Performance tuning is a method that helps perform the specific job faster and better by making the resources participate actively in a specified job. The factors considered during Performance Tuning are network bandwidth, disk throughput, CPU overhead, and memory.

Parameters of Performance Tuning

Performance Tuning is done using the following parameters:

Parameters	Function
dfs.datanode.handler.count	Handles the number of server threads for the DataNode
dfs.datanode.du.reserved	Reserves space in bytes per volume
dfs.replication	Sets the replication factor
fs.checkpoint.dir	Stores the temporary images and merges them in need of a job in the local file system of the DFS Secondary NameNode.
mapred.local.dir.minspacestart	Limits the job tasks for execution if the space is relatively less
dfs.block.size	Changes the block size; default is 64MB
dfs.name.edits.dir	Determines the exact storage position of the DFS NameNode transaction (edits) file in the local file system

 13

Parameters of Performance Tuning

Performance Tuning is done using the following parameters.

dfs.datanode.handler.count handles the number of server threads for the DataNode.

dfs.datanode.du.reserved reserves space in bytes per volume.

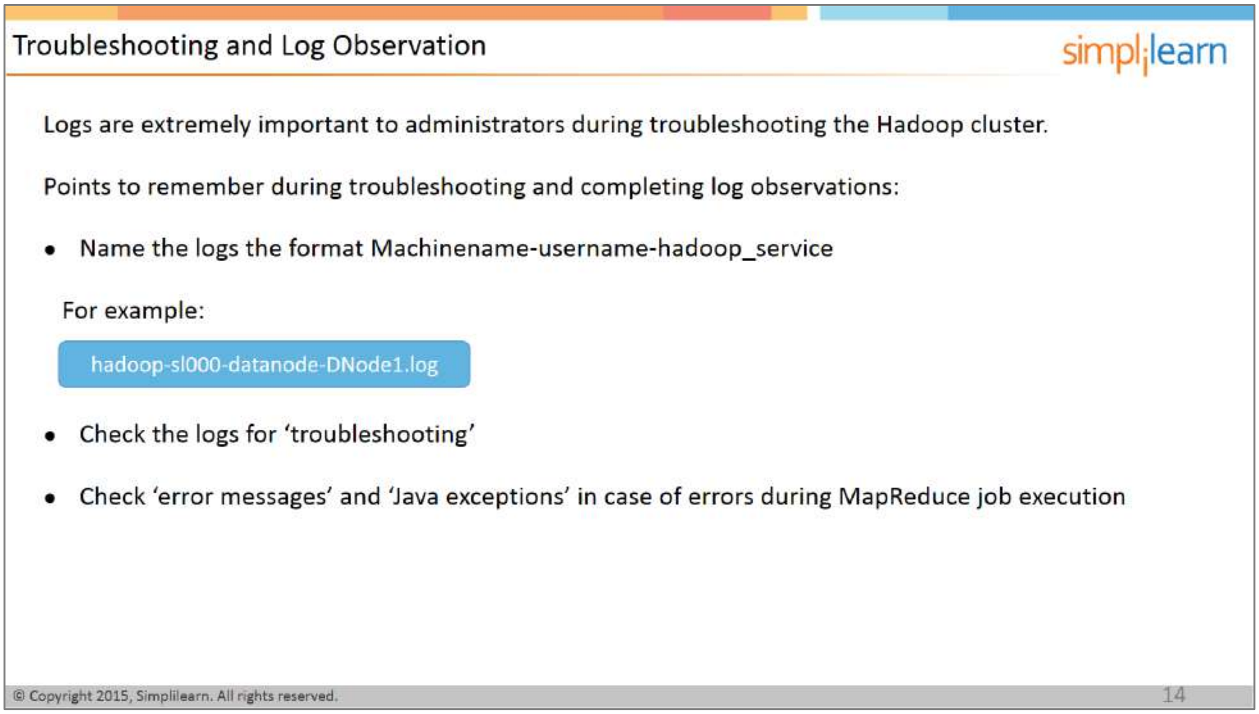
dfs.replication sets the replication factor.

fs.checkpoint.dir is the default replication factor, which stores the temporary images and merges them in need of a job in the local file system of the DFS Secondary NameNode.

mapred.local.dir.minspacestart limits the job tasks for execution if the space is relatively less.

dfs.block.size changes the block size; the default is 64MB.

dfs.name.edits.dir determines the exact storage position of the DFS NameNode transaction or edits file in the local file system.

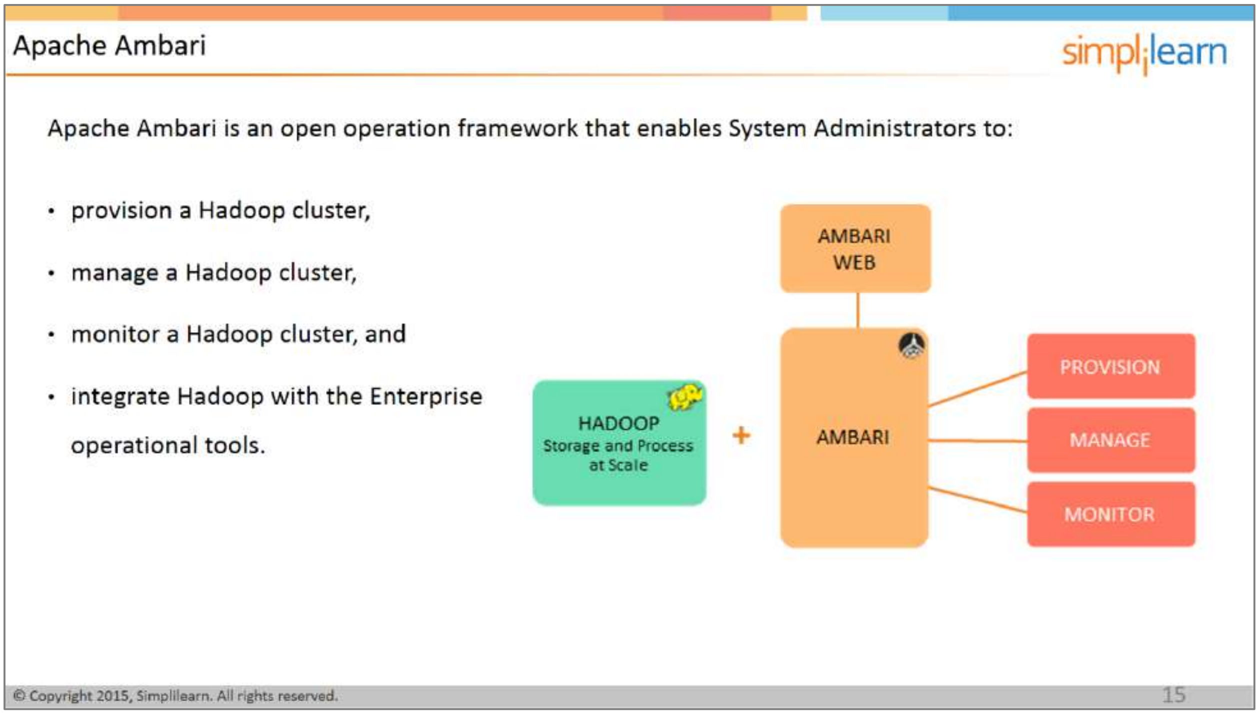


Troubleshooting and Log Observation

Logs are important to Administrators during troubleshooting the Hadoop cluster.

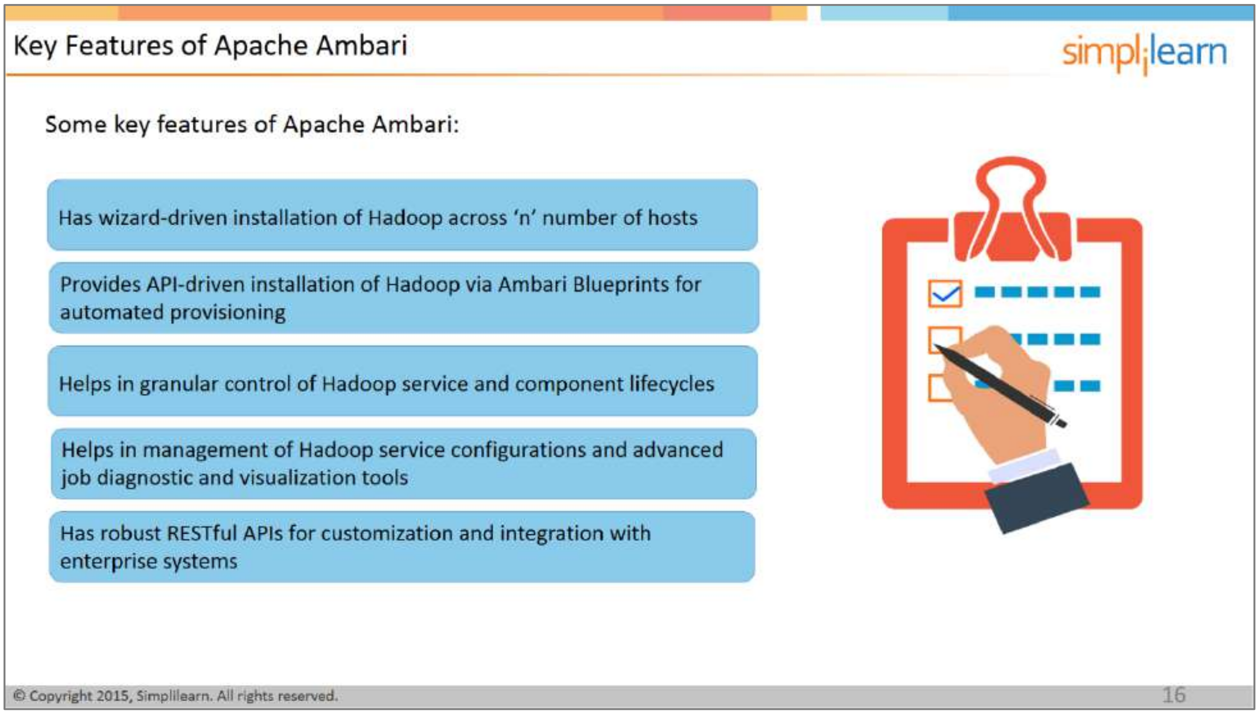
Remember the following points during troubleshooting and completing log observations:

- Name the logs in the Machinename-username-hadoop_service format. An example is hadoop-sl 000-datanode-DNode1.log.
- Logs are always checked for troubleshooting.
- Check the Java exceptions and error messages in case of errors during MapReduce job execution.



Apache Ambari

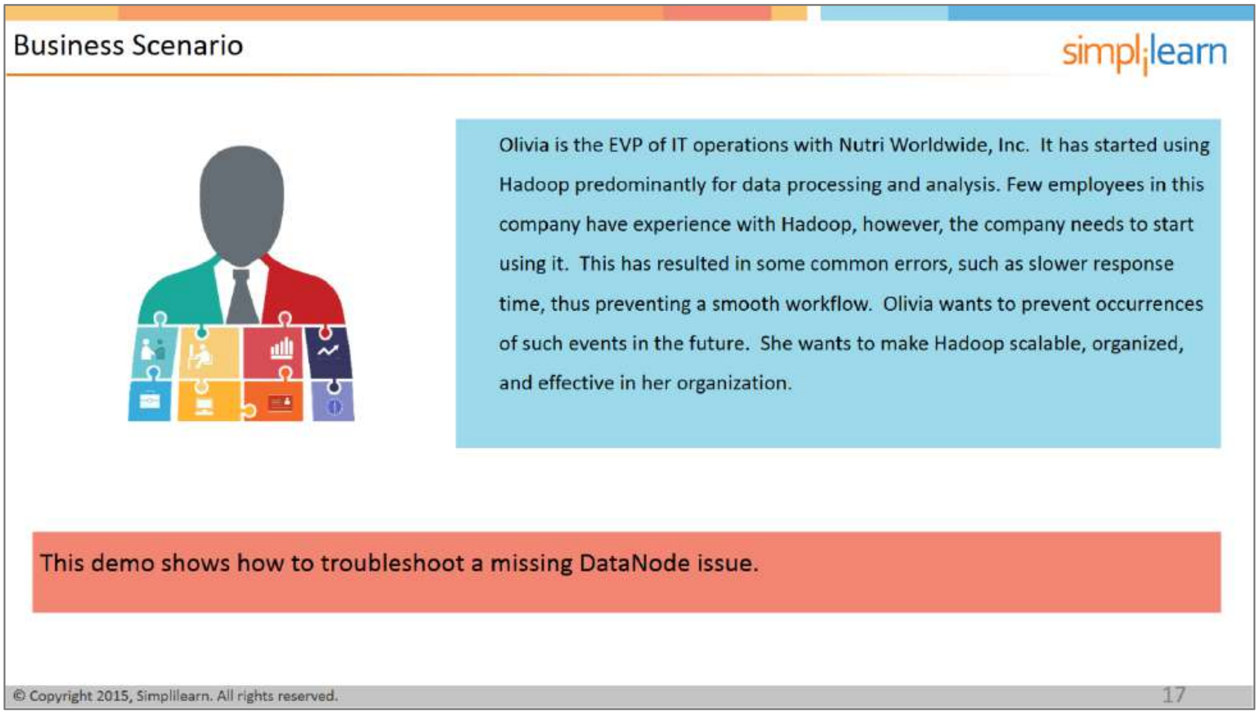
Apache Ambari is an open operation framework that enables System Administrators to provision, manage, and monitor a Hadoop cluster, as well as integrate Hadoop with the Enterprise operational tools.



Key Features of Apache Ambari

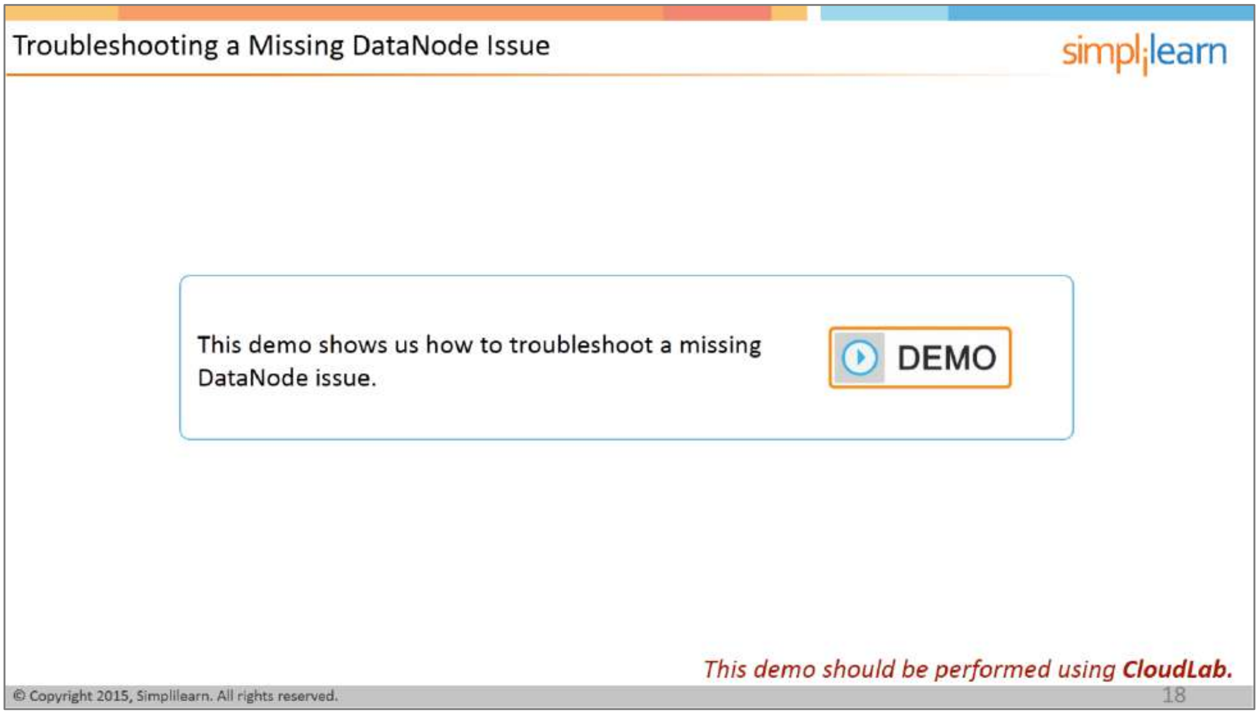
Following are a few key features of Apache Ambari:

- It has wizard-driven installation of Hadoop across 'n' number of hosts
- It provides API-driven installation of Hadoop via Ambari Blueprints for automated provisioning
- It helps in granular control of Hadoop service and component lifecycles
- It helps in management of Hadoop service configurations and advanced job diagnostic and visualization tools
- It has robust RESTful APIs for customization and integration with enterprise systems



Business Scenario

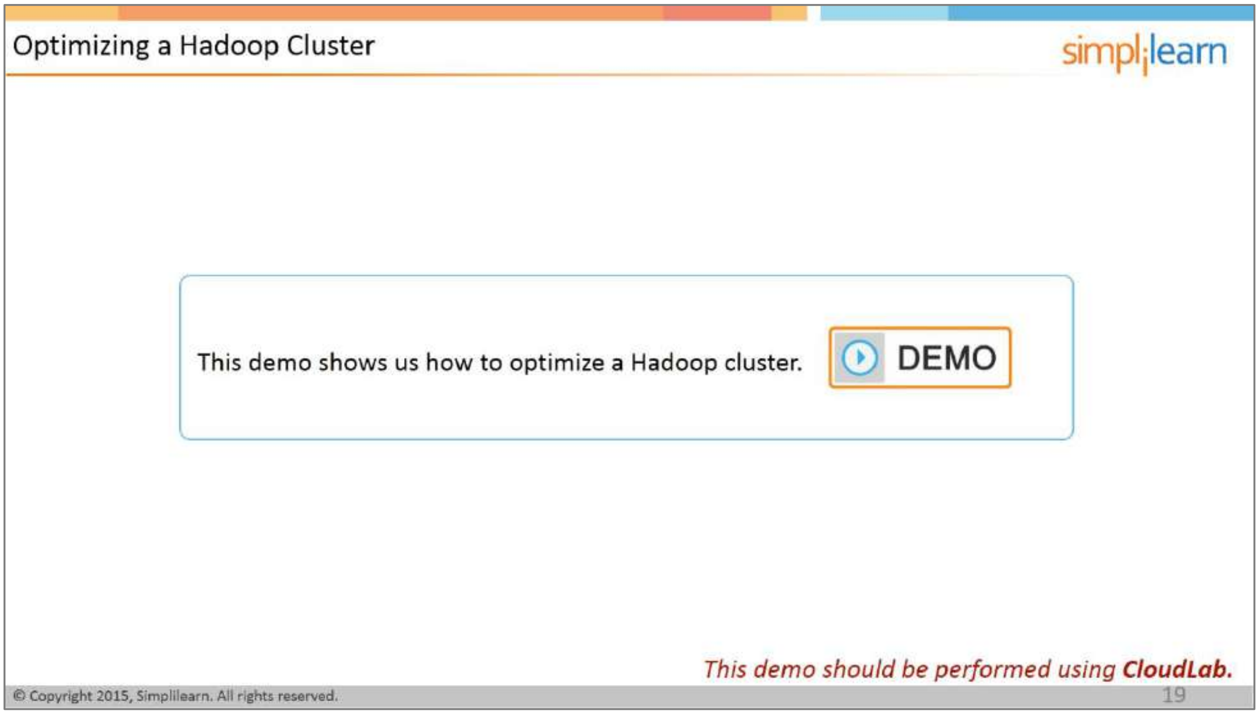
Olivia is the EVP of IT operations with Nutri Worldwide, Inc. It has started using Hadoop predominantly for data processing and analysis. Few employees in this company have experience with Hadoop, however, the company needs to start using it. This has resulted in some common errors, such as slower response time, thus preventing a smooth workflow. Olivia wants to prevent occurrences of such events in the future. She wants to make Hadoop scalable, organized, and effective in her organization.



Troubleshooting a Missing DataNode Issue

Let us summarize the steps performed:

1. First create an issue of missing DataNode to perform troubleshooting. Reformat the NameNode.
2. In the re-format question, type upper case Y and press Enter to continue.
3. Type Clear.
4. The format is successfully performed.
5. Use the command shown to start all the services.
6. All the Hadoop services have successfully started.
7. Use the jps command to check the status of the Hadoop services.
8. Use the command shown to open the log file for DataNode.
9. You will see a bunch of data displayed.
10. This part shows the reason for the DataNode service not starting.
11. Use the command shown to open the location to re-write the namespace for DataNode. Press Enter.
12. Delete the old namespace id and re-write the namespace id as 1861898000. Then, save the file.
13. Use the command shown to stop the service.
14. Use the command shown to start the service now.
15. Let's verify whether the DataNode is active. Type jps and press Enter.
16. You will now see that the DataNode service is successfully restored.

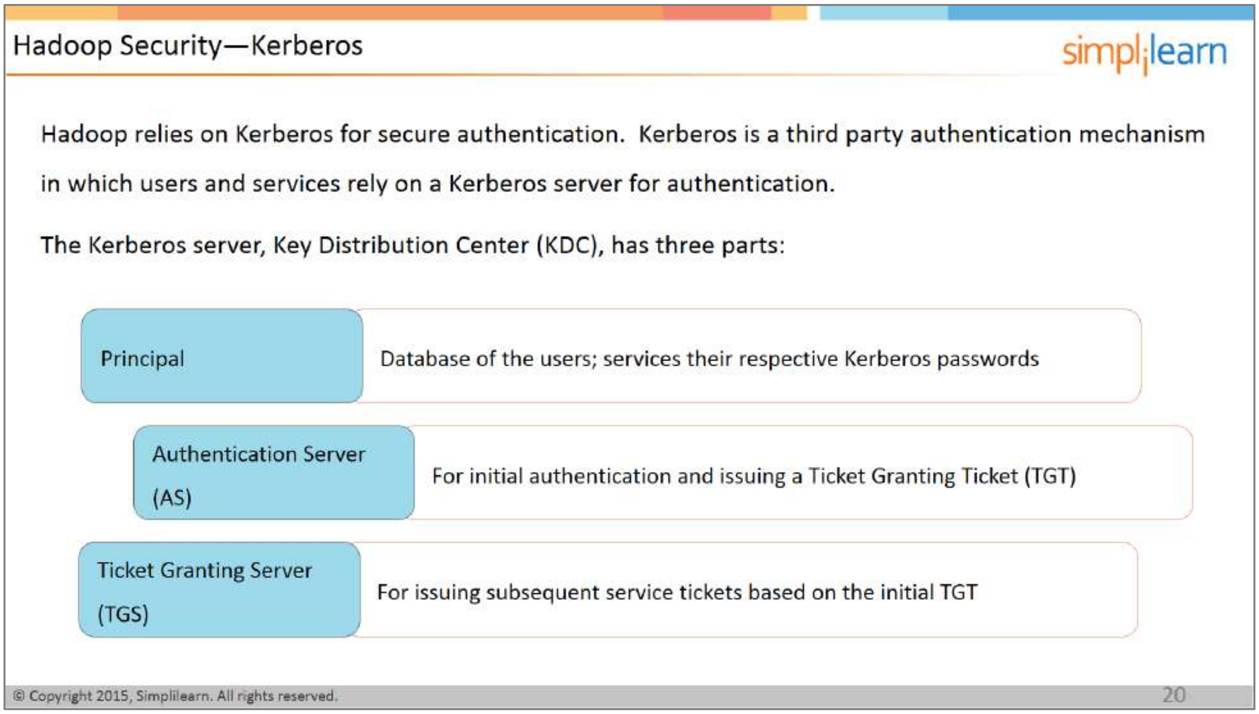


Optimizing a Hadoop Cluster

Let us summarize the steps performed:

1. Create a chunk of data and perform sorting.
2. A map-reduce operation is performed for sorting.
3. Let’s check the generated data in the GUI.
4. Click Browse the file system.
5. Click the data link.
6. Click the demoinput link.
7. Observe that a 500MB data is generated and the default block size is 64MB. Click on Go back to DFS home.
8. Perform sorting operation in this data.
9. You can check the job status in MapReduce GUI.
10. Note the address.
11. Click the Terasort job to see the job status and completion time.
12. Perform some optimization by opening hdfs-site.xml.
13. You need to set the hdfs-site.xml parameters.
14. Set dfs.replication as 2, dfs.block.size as 128MB, dfs.namenode.handler.count as 20, and dfs.datanode.handler.count as 5. Press Enter.
15. Open mapred-site.xml.
16. You need to set mapred-site.xml parameters now. Press Enter once you set the value for all parameters.
17. You need to delete the demo output and input file.

18. The command to delete the demo input file is shown.
19. Stop the hadoop services and start again.
20. The command to start the hadoop service is start-all.sh
21. Press Enter.
22. Ensure that all services are active. This is done using the command jps.
23. Type clear and press Enter.
24. Let's re-create the data using Teragen. The command is shown. Press Enter.
25. The MapReduce operation will start for generating a data file. Press Enter.
26. Let's now perform terasort on the generated data. The command is shown. Press Enter.
27. Let's check the data in GUI.
28. Click the data link.
29. Click the demoinput file.
30. You will observe that the block size is now 128MB.
31. Click the second job to find the job execution time.
32. Thus we have successfully performed the optimization process.



Hadoop Security—Kerberos

So far we have discussed Hadoop configuration and troubleshooting. Let us now discuss Hadoop Security in detail.

Hadoop relies on Kerberos for secure authentication. Kerberos is a third party authentication mechanism in which users and services rely on a Kerberos server for authentication.

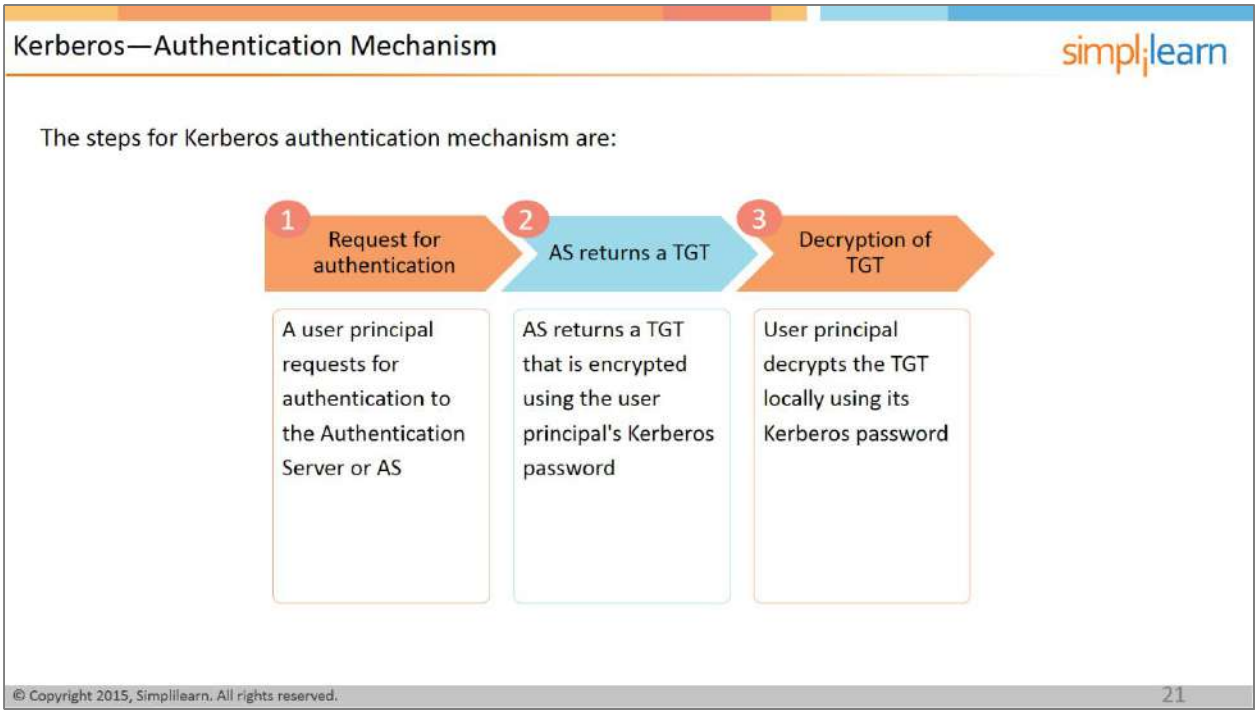
Kerberos server, also known as Key Distribution Center or KDC, has three parts: They are:

1. Principal
2. Authentication Server
3. Ticket Granting Server

Principal is a database of the users, and it services their respective Kerberos passwords.

Authentication Server or AS is for initial authentication and issuing a Ticket Granting Ticket or TGT.

Ticket Granting Server or TGS is for issuing subsequent service tickets based on the initial TGT.



Kerberos—Authentication Mechanism

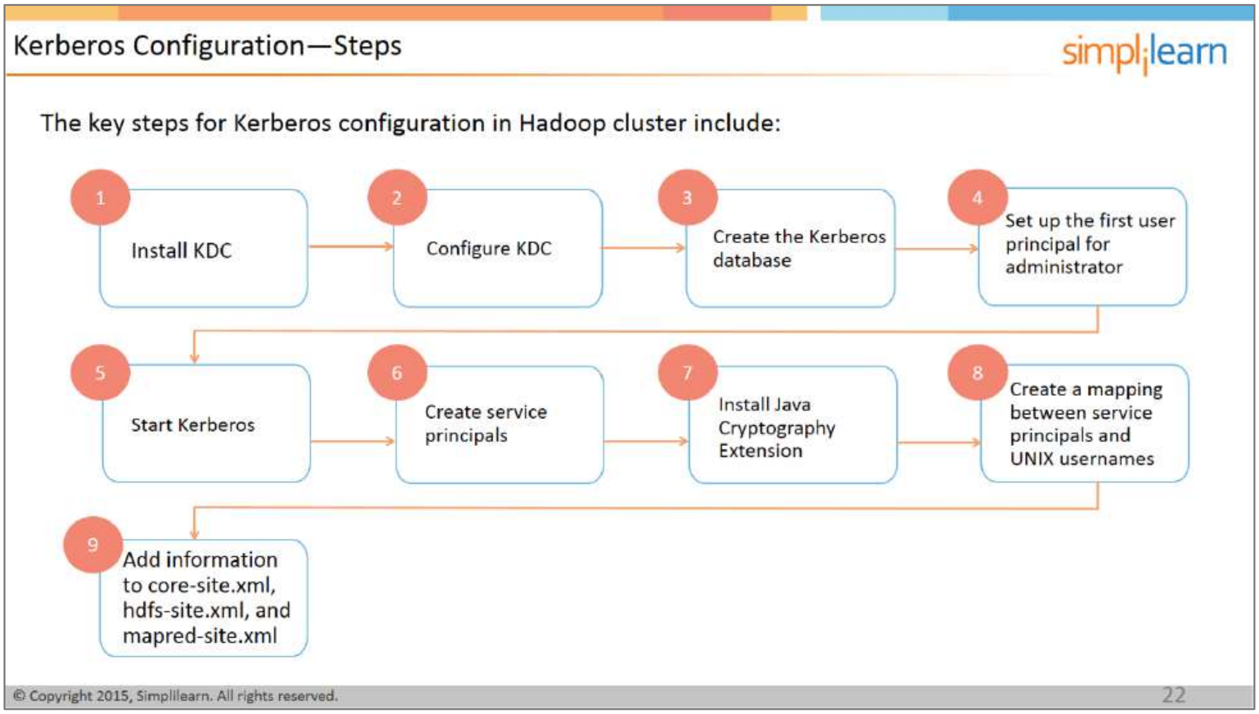
The steps for the Kerberos authentication mechanism are as follows.

Step 1: A user principal requests for authentication to the AS.

Step 2: AS returns a TGT that is encrypted using the user principal's Kerberos password.

Step 3: User principal decrypts the TGT locally using its Kerberos password.

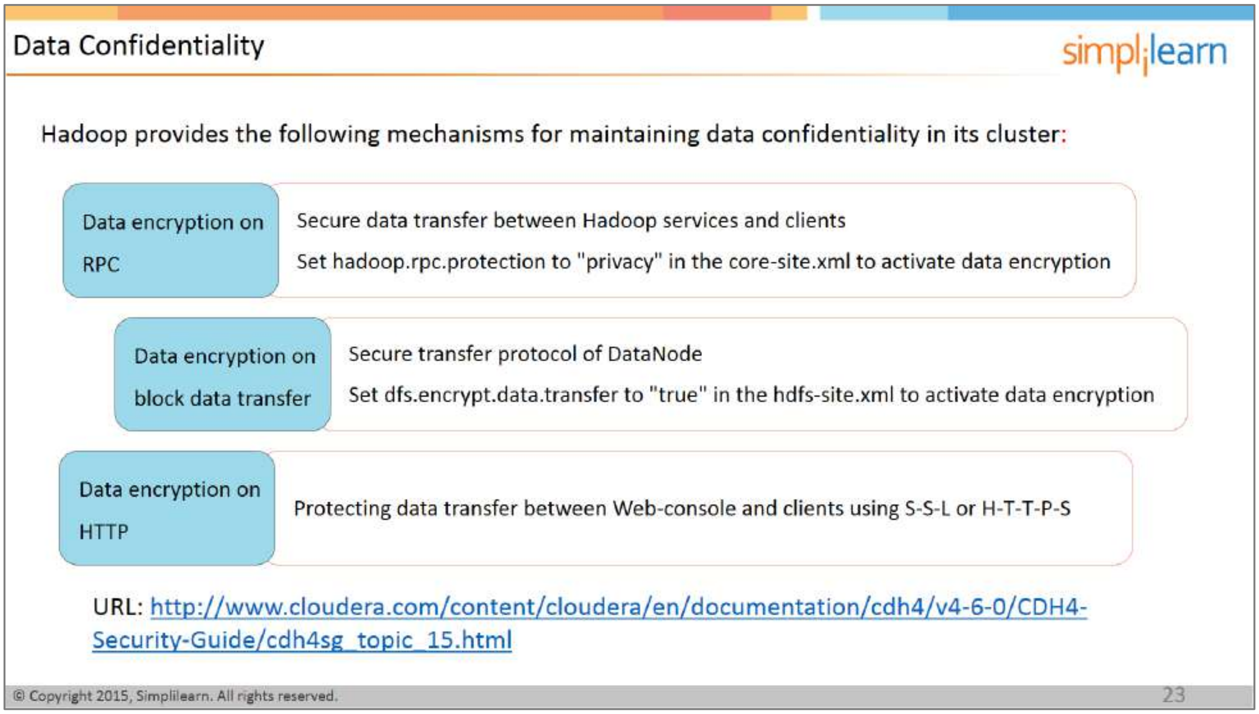
The service principal uses a special file, called a keytab, which contains its authentication credentials to avoid providing a password each time to decrypt the TGT.



Kerberos configuration–Steps

The key steps for Kerberos configuration in Hadoop cluster include:

- Installing the Key Distribution Center or KDC
- Configuring the KDC
- Creating the Kerberos database
- Setting up the first user principal for administrator
- Starting Kerberos
- Creating service principals for NameNode, DataNode, Application Master, and Node Manager
- Installing Java Cryptography Extension or JCE Unlimited Strength Jurisdiction Policy File on all machines
- Creating a mapping between service principals and UNIX usernames
- Adding information to three main service configuration files: core-site.xml, hdfs-site.xml, and mapred-site.xml



Data confidentiality

Hadoop also provides the following mechanisms for maintaining data confidentiality in its cluster.

Data Encryption on RPC implies securing data transfer between Hadoop services and clients. For this, you need to set hadoop.rpc.protection to ‘privacy’ in the core site .xml which will activate data encryption.

Data encryption on block data transfer implies securing transfer protocol of DataNode. To activate this, set dfs.encrypt.data.transfer to ‘true’ in the hdfs-site.xml.

Data encryption on HTTP implies protecting data transfer between Web-console and clients using S-S-L or H-T-T-P-S.

The URL shown refers to distribution security guide on how to activate these mechanisms.

Spot the Error

Brian has recently joined in Nutri Worldwide, Inc. as a Data Analyst Trainee. He was asked to upgrade the hardware of the clusters. He executed the JPS command to check the status of the Hadoop services. He found the DataNode service is missing and planned to troubleshoot the missing DataNode. He needed to read the log files to identify the reason for the issue. However, he was unfamiliar of the Syntax to be used. Assist Brian by identifying the correct syntax to read the log files. *Spot the correct syntax.*

```
hadoop@HadoopPseudoVM:~$ cat /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
hadoop@HadoopPseudoVM:~$ host /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
hadoop@HadoopPseudoVM:~$ /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
hadoop@HadoopPseudoVM:~$ sudo vi /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
```

 25

Spot the Error

The incorrect syntax are:

```
hadoop@HadoopPseudoVM:~$ host /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
```

And

```
hadoop@HadoopPseudoVM:~$ /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
```

And

```
hadoop@HadoopPseudoVM:~$ sudo vi /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
```

The correct syntax is:

```
hadoop@HadoopPseudoVM:~$ cat /usr/local/hadoop/logs/hadoop-hadoop-datanode-HadoopPseudoVM.log
```

 26

QUIZ 1

Which security mechanism is used by Hadoop?

a. ACL

b. Proxy

c. AD

d. Kerberos

QUIZ 2

Which visual monitoring tool provides time based historical performance statistics?

a. JMX

b. Ganglia

c. Chukwa

d. Nagios

simplilearn

QUIZ 3

What are the node types put in slaves file?

a. Application Master and Node Manager

b. NameNode and DataNode

c. Application Master and NameNode

d. Node Manager and DataNode

 32

simplilearn

QUIZ 4

What controls all other parameters?

a. hadoop.tmp.dir

b. mapred.local.dir

c. dfs.data.dir

d. dfs.name.dir

 34

QUIZ 5

What is the difference between a failed task attempt and a killed task attempt?

a. A failed task attempt is a task attempt that threw an unhandled exception. A killed task attempt is one that was terminated by the Application Master.

b. A failed task attempt is a task attempt that did not generate any key-value pairs. A killed task attempt is a task attempt that threw an exception and was thus killed by the execution framework.

c. A failed task attempt is a task attempt that is completed, but with an unexpected status value. A killed task attempt is a duplicate copy of a task attempt that was started as part of speculative execution.

d. A failed task attempt is a task attempt that threw a Runtime Exception (the task fails). A killed task attempt is a task attempt that threw any other type of exception (such as an I/O Exception); the execution framework catches these exceptions and reports them as killed.

QUIZ 6

Which happens if the NameNode crashes?

a. HDFS becomes unavailable until a new NameNode is restored.

b. The Secondary NameNode seamlessly takes over, and there is no service interruption.

c. HDFS becomes unavailable to new MapReduce jobs, but running jobs will continue until completion.

d. HDFS becomes temporarily unavailable until an administrator starts redirecting client requests to the Secondary NameNode.

QUIZ 7

Your cluster has 10 DataNodes, each with a single 1 TB hard drive. You utilize all of your disk capacity for HDFS, reserving none for MapReduce. You implement default replication settings. What is the storage capacity of your Hadoop cluster (assuming no compression)?

a. About 3 TB

b. About 5 TB

c. About 10 TB

d. About 11 TB

ANSWERS:

S.No.	Question	Answer & Explanation
1	Which security mechanism is used by Hadoop?	It is Kerberos.
2	Which visual monitoring tool provides time based historical performance statistics?	Ganglia provides information for hours, days, weeks, or months.
3	What are the node types put in slaves file?	The Node Manager and DataNode are multi-nodes controlled by Application Master and name nodes, so they are slave nodes.
4	What controls all other parameters?	hadoop.tmp.dir defines the prefix of all other directory/path info.
5	What is the difference between a failed task attempt and a killed task attempt?	A failed task attempt is a task attempt that is already completed, but with an unexpected status value, whereas a killed task attempt is a duplicate copy of a task attempt that was started as a part of speculative execution.
6	Which happens if the NameNode crashes?	If NameNode crashes, HDFS becomes unavailable until a new NameNode is restored.
7	Your cluster has 10 DataNodes, each with a single 1 TB hard drive. You utilize all of your disk capacity for HDFS, reserving none for MapReduce. You implement default replication settings. What is the storage capacity of your Hadoop cluster (assuming no compression)?	About 3 TB because the default replication value is three. Therefore, 3 DataNodes will be approximately 3 TB (1 TB multiplied by three).

Case Study

Scenario | Analysis | Solution

XY Networks provides network security support to many organizations. It has system-generated log files that are critical for security analysis and monitoring. These files are growing in size, and the company is running out of storage space. It also uses expensive and obsolete backup mechanism for these files. The company was given an estimate of 5 million dollars to upgrade their storage and backup mechanism. It's IT team suggests that storage costs can be reduced by 90% by using Hadoop. A cluster of more than 100 machines is required to set up and maintain Hadoop and other ecosystem products. The IT team has heard of Ambari, which can help monitor the cluster.

 43

Case Study

Scenario | Analysis | Solution

The IT team does research on Ambari, and finds that it can be used to monitor Hadoop and other ecosystem tools such as Hive, HBase, and Oozie. It also interacts with machine monitoring tools like Nagios and Ganglia, and helps them add new machines or remove some machines for maintenance. It also alerts Administrators in case of a resource outage.

Some advantages of using Ambari are:

1. A single dashboard for all the tools with web-based interface.
2. Provisioning resources and machines from any place.
3. Health check of all servers.
4. Easy to configure.

 44

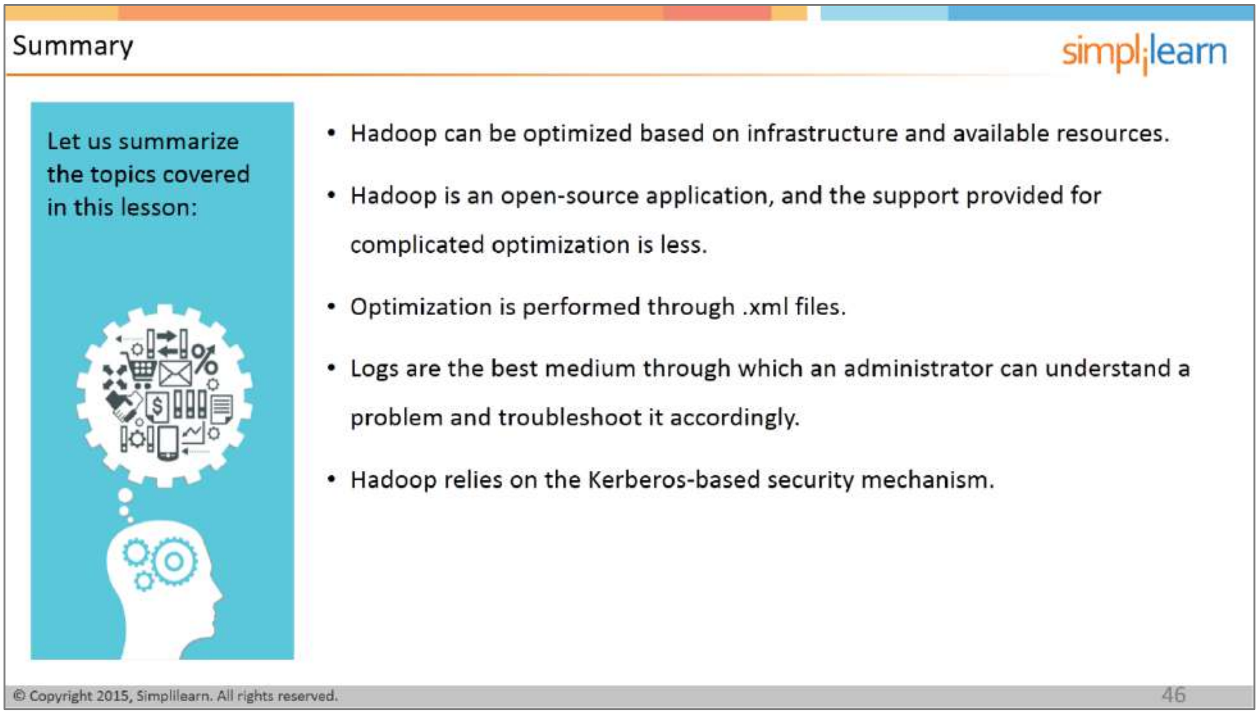
Case Study

Scenario | Analysis | Solution

Perform the following steps to set up a 3-node hadoop cluster with Hadoop, Hive, Pig, HBase, Oozie, Sqoop, Flume, and Spark, and install Ambari to monitor the cluster:

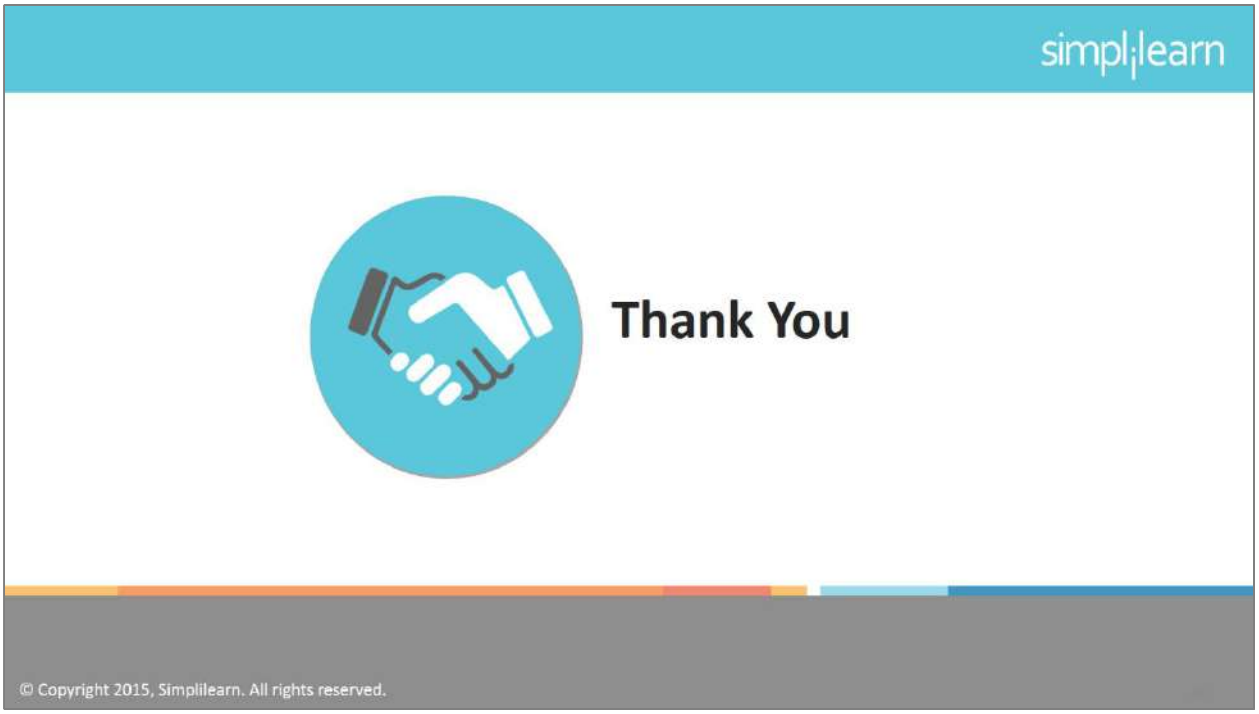
1. Check the dashboard for all the tools installed.
2. Check the services and their status.
3. Check the host machines installed.
4. Check alerts in case of Server malfunction.
5. Check resource usage like memory, disk, and network.

 45



Summary

- Let us summarize the topics covered in this lesson:
- Hadoop can be optimized based on infrastructure and available resources.
- Hadoop is an open-source application, and the support provided for complicated optimization is less.
- Optimization is performed through .xml files.
- Logs are the best medium through which an administrator can understand a problem and troubleshoot it accordingly.
- Hadoop relies on the Kerberos-based security mechanism.



Conclusion

With this we conclude the last lesson of Big Data and Hadoop Developer course.

Thank you and happy learning!

Usage of Trademarks

1. VMware reserves all rights to its trademarks, service marks, and logos, which together with the trademarks, service marks, and logos of subsidiaries of VMware will be collectively referred to as "VMware Marks." VMware does not object to the use of the VMware Marks (other than logos) to refer specifically to the products of VMware. However, a purely referential use or nominative fair use of the VMware Marks must be in plain text. Qualified legal counsel should be consulted in order to distinguish a referential use from an infringing use.
2. The Apache Software Foundation has stated that only software officially released by the Apache Hadoop Project can be called Apache Hadoop or Distributions of Apache Hadoop.[89] The naming of products and derivative works from other vendors and the term "compatible" are somewhat controversial within the Hadoop developer community.[90]
3. Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation.
4. Apache Hive, Hive, Apache, the Apache feather logo, and the Apache Hive project logo are trademarks of The Apache Software Foundation.
5. Apache Hadoop, Hadoop, HDFS, Hbase, and the HBase project logo are trademarks of the Apache Software Foundation.

Usage of Trademarks (contd.)

6. Apache Hadoop, Hadoop, HDFS, Avro, Cassandra, Chukwa, HBase, Hive, Mahout, Pig, and Zookeeper are trademarks of the Apache Software Foundation.
7. Apache ZooKeeper, ZooKeeper, Apache, the Apache feather logo, and the Apache ZooKeeper project logo are trademarks of The Apache Software Foundation.
8. Cloudera, the Cloudera logo, Cloudera Impala, Impala, and any other product or service names or slogans contained in the Services are trademarks of Cloudera, except as otherwise noted, and may not be copied, imitated, or used, in whole or in part, without the prior written permission of Cloudera. You may not use any metatags or any other "hidden text" utilizing "Cloudera" or any other name, trademark , product, or service mark of Cloudera without our prior written permission. In addition, the look and feel of the Services, including all page headers, custom graphics, button icons and scripts, are the service mark, trademark, and/or trade dress of Cloudera and may not be copied, imitated or used, in whole or in part, without our prior written permission. All other trademarks, registered trademarks, product names, and company names or logos mentioned in the Services, including but not limited to Hadoop and the Hadoop elephant logo, are the property of their respective owners and may not be copied, imitated, or used, in whole or in part, without their prior written permission. Reference to any products, services, processes, or other information, by trade name, trademark, manufacturer, supplier, or otherwise does not constitute or imply endorsement, sponsorship, or recommendation by Cloudera.

Simplilearn offers flexible modes of learning to suit the learning needs of working professionals like you.

Self-Learning

Instructor-led Training

Classroom | Online

Courses we offer:

- PMP® Training
- PRINCE2® Training
- PMI-ACP® Training
- CSM Training
- CBAP® Training
- Agile and Scrum Training
- RMP® Training
- Big Data & Hadoop Developer Training
- Cloud Computing Training
- ITIL® Training
- TOGAF Training
- Six Sigma Training
- CFA Training
- Financial Modeling Training
- FRM Training
- CEH Training
- CISSP® Training
- CCNA® Training
- SAP Training
- Microsoft Training
- Salesforce Training

+200 more courses

www.simplilearn.com

Simplilearn is a leading provider of a suite of professional certification courses that address unique learning needs of working professionals

ACCREDITATIONS

Global Registered Education Provider (REP) for PMI, USA. Our REP ID is 3147	Accredited Training Organization (ATO) by APMG International, UK for PRINCE2® Foundation and Practitioner	Accredited Training Organization (ATO) by APMG International, UK for ITIL® 2011 Foundation and ITIL® 2011Intermediate	Listed on CFA Institute Prep Providers for CFA Level I. Follows CFA Institute Prep Provider Guidelines	Listed on GARP Exam Preparation Providers for FRM Part 1	Accredited Training Provider (ATP) and Authorized Examination Center (AEC) by ISTQB
IASSC Accredited Examination Center (AEC) by PEOPLECERT	Accredited Training Provider (ATP) and Accredited Examination Center (AEC) by EXIN	Accredited Training Provider (ATP) by ASTQB	Accredited Training Organization (ATO) by TUV-SUD for ITIL® 2011 Foundation and ITIL® 2011 Intermediate	Simplilearn is a Registered Education Provider (REP) of Scrum Alliance, Inc. Scrum Alliance is the governing body for CSM and CSPO certifications	Simplilearn is an Authorized Training Partner (ATP) and Accredited Training Center (ATC) for EC-Council

Disclaimer

"PMI®", "PMBOK®", "PMP®" and "PMI-ACP®" are registered marks of the Project Management Institute, Inc.

The Swirl logoTM is a trade mark of AXELOS Limited.
ITIL® is a registered trade mark of AXELOS Limited.
PRINCE2® is a Registered Trade Mark of AXELOS Limited.
MSP® is a Registered Trade Mark of AXELOS Limited

"CSM", "CST" are Registered Trade Marks of The Scrum Alliance, USA.

COBIT® is a trademark of ISACA® registered in the United States and other countries.
CISA® is a Registered Trade Mark of the Information Systems Audit and Control Association (ISACA) and the IT Governance Institute
CISSP is a registered mark of the The International Information Systems Security Certification Consortium ((ISC)2)

FRM®, GARPTM and Global Association of Risk ProfessionalsTM, are trademarks owned by the Global Association of Risk Professionals, Inc
Global Association of Risk Professionals, Inc. (GARPTM) does not endorse, promote, review or warrant the accuracy of the products or services offered by Simplilearn for FRM® related information, nor does it endorse any pass rates claimed by the provider. Further, GARP is not responsible for any fees or costs paid by the user to
Simplilearn nor is GARP responsible for any fees or costs of any person or entity providing any services to Simplilearn Study Program.

CFA Institute does not endorse, promote, or warrant the accuracy or quality of the products or services offered by Simplilearn. CFA Institute, CFA®, and Chartered Financial Analyst® are trademarks owned by CFA Institute.

SAP® is the trademark or registered trademark of SAP AG. Simplilearn is not affiliated with or endorsed by SAP AG. All training material is proprietary contents of Simplilearn and its Partners. SAP Access is provided by Simplilearn in partnership with Leading C, a licensed SAP provider in accordance with SAP EULA dated March 2004.


200,000+ PROFESSIONALS TRAINED
150+ COUNTRIES
200+ COURSES
ACCREDITED BY 20+ GLOBAL BODIES
PMP® Training
PROJECT MANAGEMENT
PRINCE2® Certification
CSM Training
AGILE & SCRUM CERTIFICATION
(PMI-ACP)® Training
ITIL® Intermediate Certification
IT SERVICE MANAGEMENT
ITIL® Expert Training
COMPTIA Cloud+ Training
BIG DATA & CLOUD COMPUTING
Big Data & Cloud Computing Training
CISSP® Training
IT SECURITY MANAGEMENT
COBIT® Training
Six Sigma Certification
QUALITY MANAGEMENT
Lean Management Training
Financial Modeling with MS Excel Training
FINANCE MANAGEMENT
CFA Leve1 Training
CCNA® Training
CCNP® Training
CISCO CERTIFICATION
Microsoft® Excel Training
MICROSOFT® CERTIFICATION
Microsoft® Project Training
Software Testing Training
SOFTWARE TESTING
CTFL® Training
SAP CERTIFICATION
SAP Technical Modules
SAP Functional Modules
support@simplilearn.com
Live Chat 24/7 on simplilearn.com