
# Big Data Hadoop and Spark Developer



simplilearn


# Apache Spark - Next Generation Big Data Framework

# Learning Objectives

By the end of this lesson, you will be able to:

- Know the history of Spark
- Understand the advantages of Spark
- Interpret the companies implementing Spark with use cases
- Understand Spark and its core components
- Learn Spark's architecture
- Use Spark cluster in real world - Development, QA, and Production

DATA AND
ARTIFICIAL INTELLIGENCE

# History of Spark

# History of Spark

Started at UC Berkeley AMPLab by Matei Zaharia

**2009**

amplab UC BERKELEY

**2010**

BSD

Open sourced under a BSD license

Given to the Apache Software Foundation and the license was changed to Apache 2.0

**2013**

The Apache Software Foundation

**2014**

databricks

Became an Apache top-level project. Used by Databricks to set a world record in large-scale sorting

Exists as a next generation real-time and batch processing framework

**Present**

# Batch vs. Real-Time Processing

## Batch Processing

- A large group of data or transactions is processed in a single run.
- Jobs are run without any manual intervention.
- The entire data is pre-selected and fed using command-line parameters and scripts.
- It is used to execute multiple operations, handle heavy data load, reporting, and offline data workflow.

**Example**:
Regular reports that require decision-making

## Real-Time Processing

- Data processing takes place upon data entry or command receipt instantaneously.
- It must execute real-time within stringent constraints.

**Example**: Fraud detection

# Limitations of MapReduce in Hadoop

**Unsuitable for real-time processing**

Being batch oriented, it takes minutes to execute jobs depending on the amount of data and number of nodes in the cluster.

**Unsuitable for trivial operations**

For operations like Filter and Joins, you might need to rewrite the jobs, which becomes complex because of the key-value pattern.

**Unsuitable for large data on network**

Since it works on the data locality principle, it cannot process a lot of data that requires shuffling over the network.

# Limitations of MapReduce in Hadoop

**Unsuitable with OLTP**

OLTP requires a large number of short transactions, as it works on the batch-oriented framework.

**Unsuitable for processing graphs**

The Apache Graph library processes graphs, that adds additional complexity on top of MapReduce.

**Unsuitable for iterative execution**

Being a stateless execution, MapReduce doesn't fit in use cases like k-means that need iterative execution.

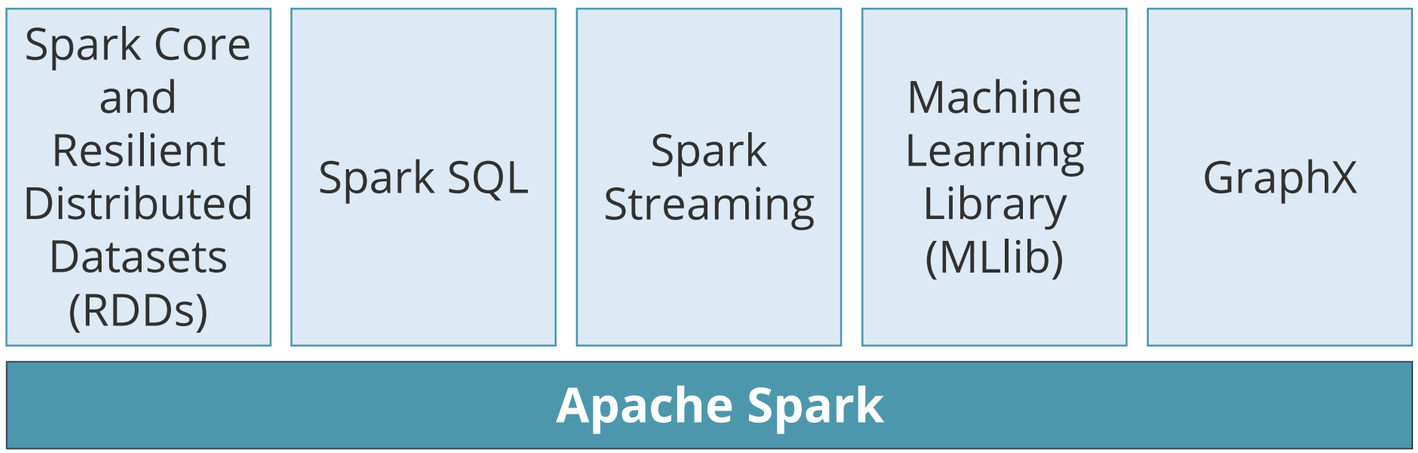
# Introduction to Apache Spark

Is suitable for real-time processing, trivial operations, and processing larger data on a network

Is an open source cluster computing framework

Provides up to 100 times faster performance for a few applications with in-memory primitives, compared to the two-stage disk-based MapReduce paradigm of Hadoop

Is suitable for machine learning algorithms, as it allows programs to load and query data repeatedly



DATA AND
ARTIFICIAL INTELLIGENCE

# Components of Spark

# Components of a Spark Project

The components of a Spark project are explained below:

**Spark Core and RDDs**

As the foundation, it provides basic I/O, distributed task dispatching, and scheduling. RDDs can be created by applying coarse-grained transformations or referencing external datasets.

**Spark SQL**

As a component lying on the top of Spark Core, it introduces SchemaRDD, which can be manipulated. It supports SQL with ODBC/JDBC server and command-line interfaces.

# Components of a Spark Project

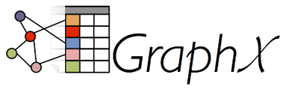
The components of a Spark project are explained below:


**Spark Streaming**

It leverages the fast scheduling capability of Spark Core, ingests data in small batches, and performs RDD transformations on them.


**MLlib**

As a distributed machine learning framework on top of Spark, it is nine times faster than the Hadoop disk-based version of Apache Mahout.




**GraphX**

Being a distributed graph processing framework on top of Spark, it gives an API and provides an optimized runtime for the Pregel abstraction.

# Application of In-Memory Processing

In column-centric databases, informations that are similar, can be stored together. The working of in-memory processing can be explained as below:

The entire information is loaded into memory, eliminating the need for indexes, aggregates, optimized databases, star schemas, and cubes.

Compression algorithms are used by most of the in-memory tools, thereby reducing the in-memory size.

Querying the data loaded into the memory is different from caching.

With in-memory tools, the analysis of data can be flexible in size and can be accessed within seconds by concurrent users with an excellent analytics potential.

It is possible to access visually rich dashboards and existing data sources.

# Language Flexibility in Spark

Spark is popular for its performance benefits over MapReduce.
Another important benefit is language flexibility, as explained below:

**Support for various development languages**

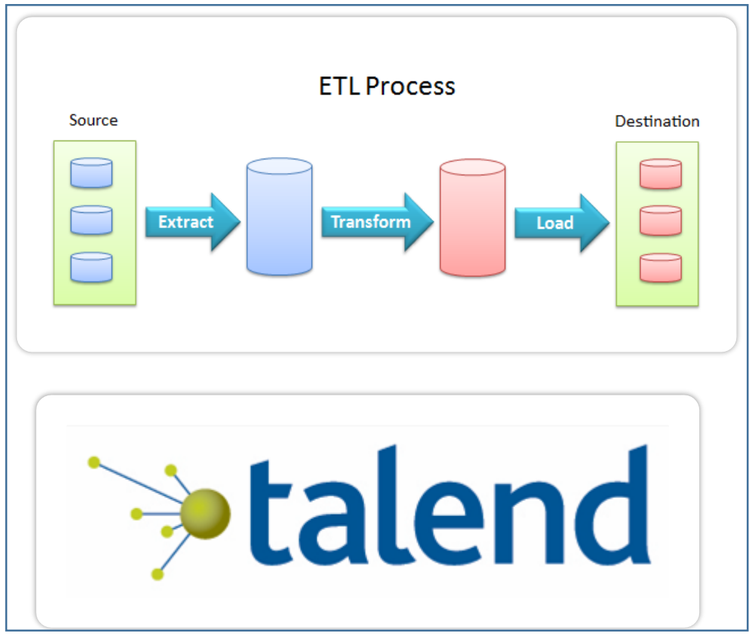
Spark supports popular development languages like Java, Scala, and Python and will support R.

**Capability to define functions in-line**

With the temporary exception of Java, a common element in these languages is that, they provide methods to express operations using lambda functions and closures.

# Hadoop Ecosystem vs. Spark








**Hadoop Ecosystem**

# Hadoop Ecosystem vs. Spark

You can perform every type of data processing using Spark that you execute in Hadoop. They are:

**Batch Processing:** Spark batch can be used over Hadoop MapReduce.

**Structured Data Analysis:** Spark SQL can be used with SQL.

**Machine Learning Analysis:** MLlib can be used for clustering, recommendations, and classification.

**Interactive SQL Analysis:** Spark SQL can be used over Stringer, Tez, or Impala.

**Real-time Streaming Data Analysis:** Spark streaming can be used over specialized library like Storm.

DATA AND
ARTIFICIAL INTELLIGENCE

# Advantages of Spark

# Advantages of Spark

The different advantages of Spark are:

**Speed:** Extends the MapReduce model to support computations like stream processing and interactive queries

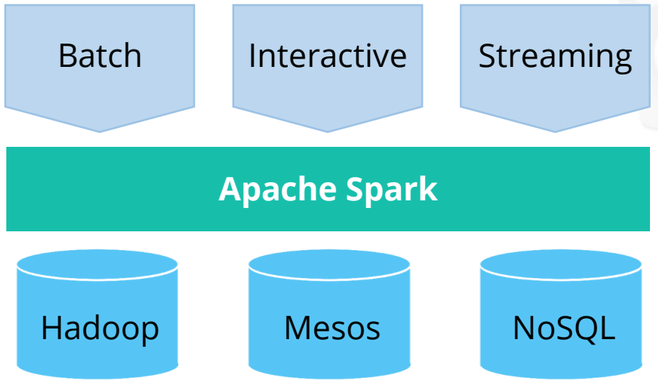
**Combination:** Covers various workloads that require different distributed systems, which makes it easy to combine different processing types and allows easy tools management

**Hadoop Support:** Allows creation of distributed datasets from any file stored in the Hadoop Distributed File System (HDFS) or any other supported storage systems

?

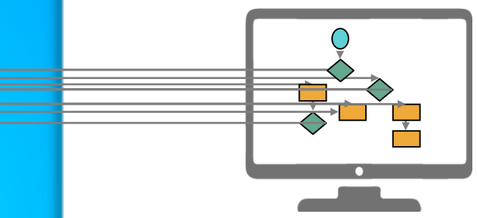
**Why does unification matter?**

- Developers need to learn only one platform
- Users can take their apps everywhere



# Advantages of Spark

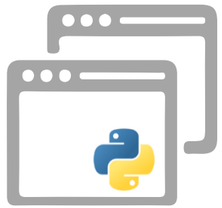
Some more advantages of Spark are:



Contains various closely integrated components for distributing, scheduling, and monitoring applications with many computational tasks


Empowers various higher level components specialized for different workloads like machine learning or SQL

Integrates and easily combines different processing models; for example, ability to write an application using machine learning to categorize data in real time as it is ingested from sources of streaming



Allows to access the same data through the Python shell for ad hoc analysis and in standalone batch applications

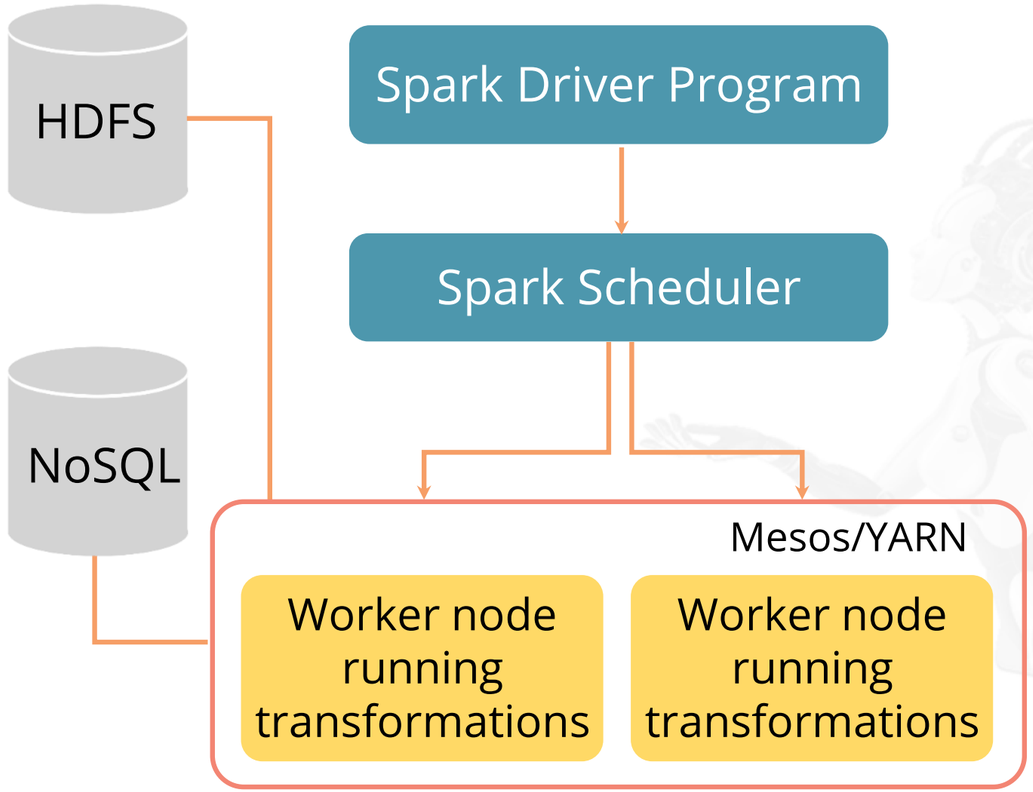
DATA AND
ARTIFICIAL INTELLIGENCE

# Spark Architecture

# Spark Architecture

The components of the Spark execution architecture are explained below:

- **Spark-submit script:** Used to launch applications on a cluster; can use all cluster managers through a uniform interface
- **Spark applications:** Run as independent sets of processes on a cluster and are coordinated by the SparkContext object in the driver program
- **Cluster managers:** Supported cluster managers are Standalone, Apache Mesos, and Hadoop YARN
- **Spark's EC2 launch scripts:** Make launching a standalone cluster easy on Amazon EC2



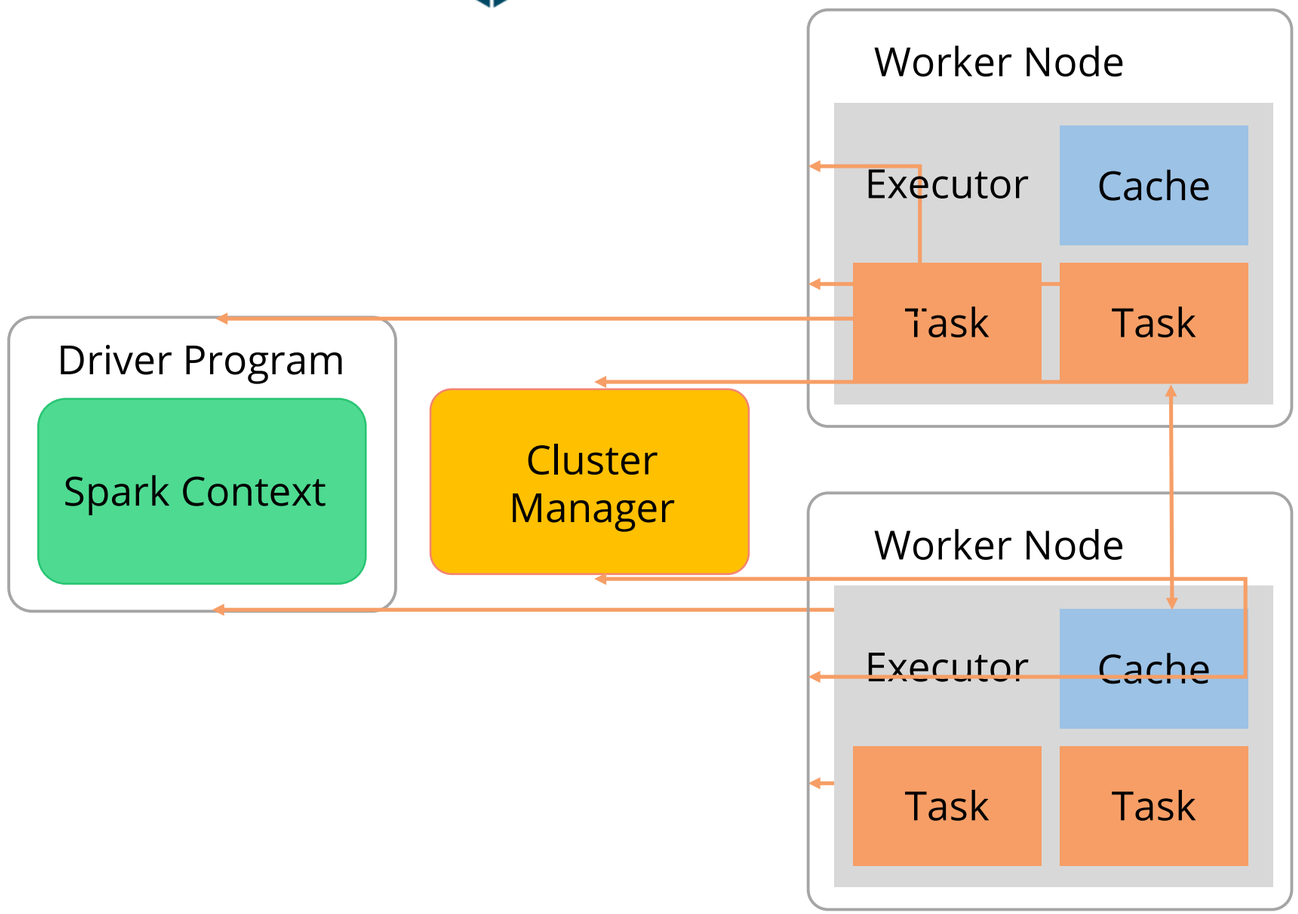
# Spark Execution Architecture

The components of the Spark execution architecture are explained below:

Spark
Standalone Cluster

Apache MESOS™

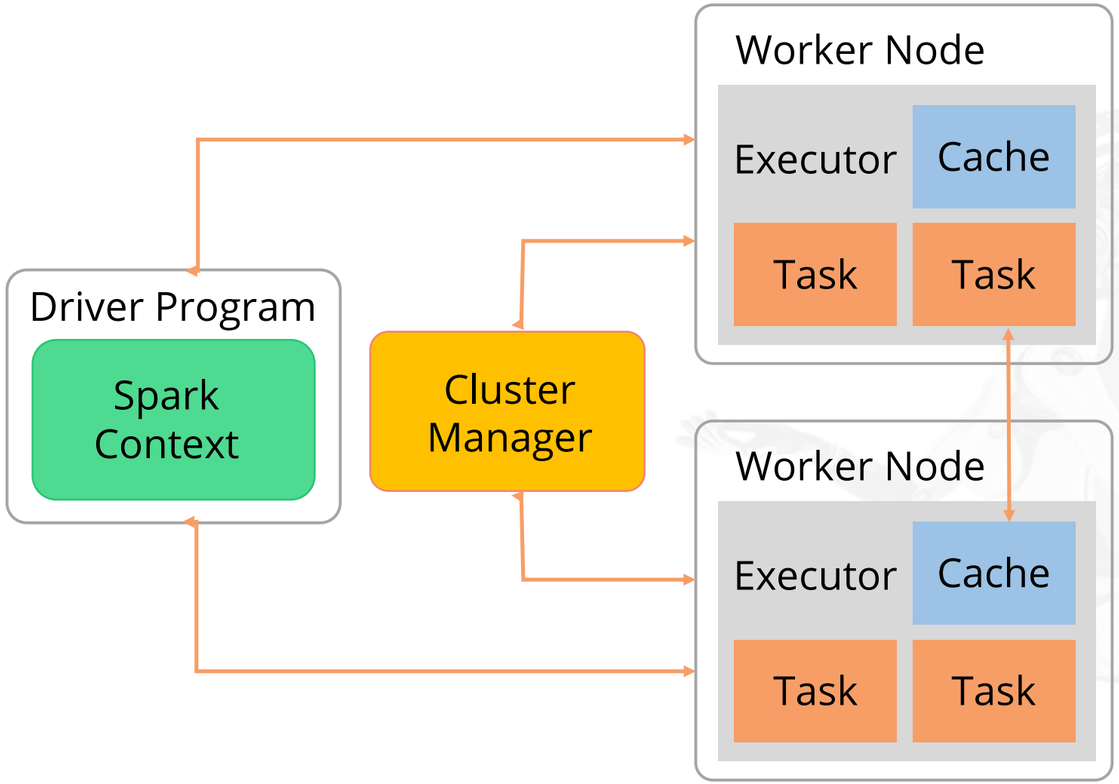
hadoop YARN



# Spark Execution: Automatic Parallelization

Spark Execution is explained as follows:

- To make the sequence of MapReduce jobs parallel in case of a complex pipeline, a scheduler tool like Apache Oozie is generally required.
- The series of individual tasks is expressed as a single program flow, which allows to parallelize the flow of operators automatically without any intervention.
- This allows certain optimizations to the engine.



DATA AND
ARTIFICIAL INTELLIGENCE

# Spark Cluster in Real World

# Running Spark in Different Modes

The different deployment modes of Spark are explained below:


## Spark as Standalone

Can be launched manually by using launch scripts, or starting a master and workers; used for development and testing


## Spark on Mesos

Has advantages like scalable partitioning among different Spark instances and dynamic partitioning between Spark and other frameworks


## Spark on YARN

Has all parallel processing and benefits of the Hadoop cluster


## Spark on EC2

Has key-value pair benefits of Amazon

# Spark Shell

The Spark Shell provides interactive data exploration (REPL).

## $Spark-shell

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.3.0
      /_/

Using Scala version 2.10.4 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)
```

## $pySpark

```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.3.0
      /_/

Using Python version 2.7.8 (default, Aug 27 2015 05:23:36)
SparkContext available as sc, HiveContext available as sqlCtx.

In [1]:
```

# SparkContext

- It is the main entry point of Spark API. Every Spark application requires a SparkContext.
- Spark Shell provides a preconfigured SparkContext called sc.

Scala

$Spark-shell

```
Spark context available as sc.
16/08/10 04:42:03 INFO repl.SparkILoop: Created sql context (with Hive support).
.
SQL context available as sqlContext.

scala>
```

python

$pySpark

```
Using Python version 2.7.8 (default, Aug 27 2015 05:23:36)
SparkContext available as sc, HiveContext available as sqlCtx.

In [1]:
```

## Assisted Practice

# Running a Scala Program in Spark Shell

Duration: 10 mins

**Problem Statement:** In this demonstration, you will learn how to run a Scala program in Spark shell.

**Access:** Click on the **Practice Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

## Assisted Practice

# Setting Up Execution Environment in IDE

Duration: 10 mins

**Problem Statement:** In this demonstration, you will learn how to set up an execution environment in IDE.

**Access:** Click on the **Practice Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

## Assisted Practice

# Understanding Various Components of Spark Web UI

Duration: 10 mins

**Problem Statement:** In this demonstration, you will understand the various components of Spark web UI.

**Access:** Click on the **Practice Labs** tab on the left side panel of the LMS. Copy or note the username and password that is generated. Click on the **Launch Lab** button. On the page that appears, enter the username and password in the respective fields, and click **Login**.

# Key Takeaways

You are now able to:

- Know the history of Spark
- Understand the advantages of Spark
- Interpret the companies implementing Spark with use cases
- Understand Spark and its core components
- Learn Spark's architecture
- Use Spark cluster in real world - Development, QA, and Production environments

DATA AND
ARTIFICIAL INTELLIGENCE


# Knowledge Check

**Knowledge Check 1**

**Which of the following are the components of Spark project?**

a. Spark Core and RDDs

b. Spark SQL

c. Spark Streaming

d. All of the above

**Knowledge Check 1**

**Which of the following are the components of Spark project?**

a. Spark Core and RDDs

b. Spark SQL

c. Spark Streaming

d. All of the above

The correct answer is **d.**

**Spark Core and RDDs, Spark SQL, and Spark Streaming are some of the components of Spark project.**

**Knowledge Check 2**

**Spark was started in the year_____.**

a. 2009

b. 2010

c. 2013

d. 2014

**Knowledge Check 2**

**Spark was started in the year_____.**

a. 2009

b. 2010

c. 2013

d. 2014

The correct answer is **a.**

**Spark was started in the year 2009 at UC Berkeley AMPLab by Matei Zaharia.**

**Knowledge Check 3**

**Which of the following are the supported Cluster Managers?**

a. Standalone

b. Apache Mesos

c. Hadoop Yarn

d. All of the above

**Knowledge Check 3**

**Which of the following are the supported Cluster Managers?**

a. Standalone

b. Apache Mesos

c. Hadoop Yarn

d. All of the above

The correct answer is **d.**

**Standalone, Apache Mesos, and Hadoop Yarn are all supported Cluster Managers.**

# Lesson-End Project

**Problem Statement:**

You have enrolled as a trainee in one of the top training institutes which provides training on Big Data. You have learned Spark and Hadoop and where to use them.
Based on that knowledge, your task is to solve the below use cases using Hadoop or Spark.

**Use-cases:**

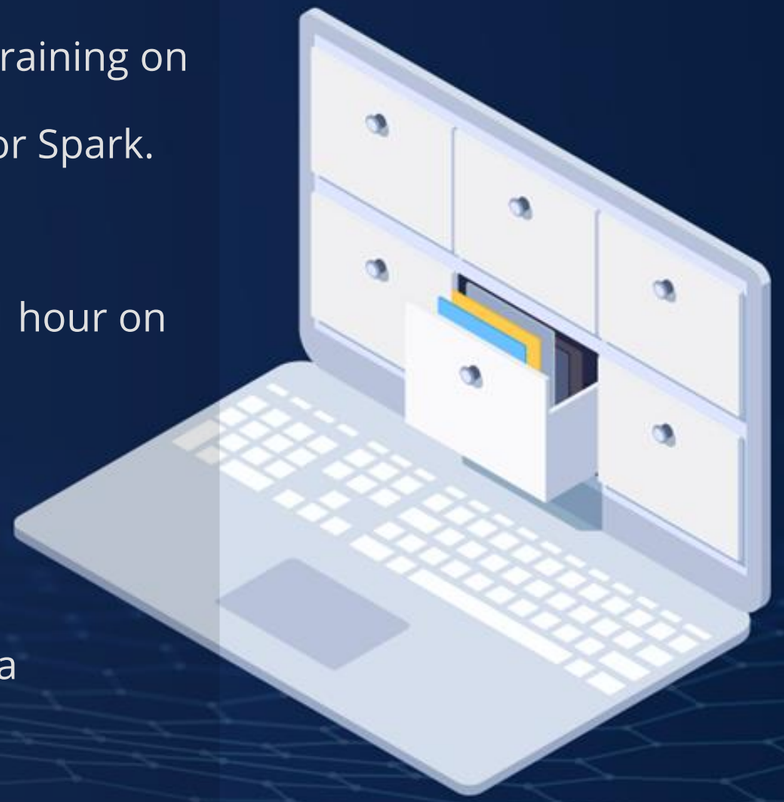
1. An E-commerce company wants to show the most trending brands in the last 1 hour on their web portal.

2. An E-commerce company wants to calculate orders for the last 5 years in the mobile category.

3. You have been given the product data of clicks and impressions. Click means when a user clicks on the product and goes to the product page.
   Impression refers to the product landing page on Amazon. You have to create a model that can predict if any product on the portal is eligible for click or not.

4. An E-commerce company wants to show the most trending products on their web portal in real-time.

5. A financial institution wants to check if the transaction is fraud or not.



DATA AND
ARTIFICIAL INTELLIGENCE

# Thank You