PySpark

Contents

[PySpark 3](#_Toc89161631)

[Spark 3](#_Toc89161632)

[PySpark and AWS 4](#_Toc89161633)

[*PySpark AWS Architecture* 4](#_Toc89161634)

[Amazon EMR 4](#_Toc89161635)

[Apache Hadoop 5](#_Toc89161636)

[Reference 6](#_Toc89161637)

# PySpark

|  |  |
| --- | --- |
| * PySpark has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for Spark. * In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programming language. * PySpark is a great language for performing exploratory data analysis at scale, building machine learning pipelines, and creating ETLs for a data platform. | * PySpark is an interface for Apache Spark in Python. It not only allows you to write Spark applications using Python APIs, but also provides the PySpark shell for interactively analyzing your data in a distributed environment. * PySpark is a Python-based API for utilizing the Spark framework in combination with Python. As is frequently said, Spark is a Big Data computational engine, whereas Python is a programming language. |
| **PySpark vs Panda**   * In very simple words Pandas run operations on a single machine whereas PySpark runs on multiple machines. If you are working on a Machine Learning application where you are dealing with larger datasets, PySpark is a best fit which could processes operations many times(100x) faster than Pandas. * Because of parallel execution on all the cores, PySpark is faster than Pandas in the test, even when PySpark didn't cache data into memory before running queries. | **Following are the steps to build a Machine Learning program with PySpark:**  Step 1) Basic operation with PySpark.  Step 2) Data preprocessing.  Step 3) Build a data processing pipeline.  Step 4) Build the classifier: logistic.  Step 5) Train and evaluate the model.  Step 6) Tune the hyperparameter.  Note: [As of v2.2](https://issues.apache.org/jira/browse/SPARK-18267?jql=project%20%3D%20SPARK%20AND%20component%20%3D%20PySpark%20AND%20text%20~%20%22pip%22), executing pip install pyspark will install Spark. |

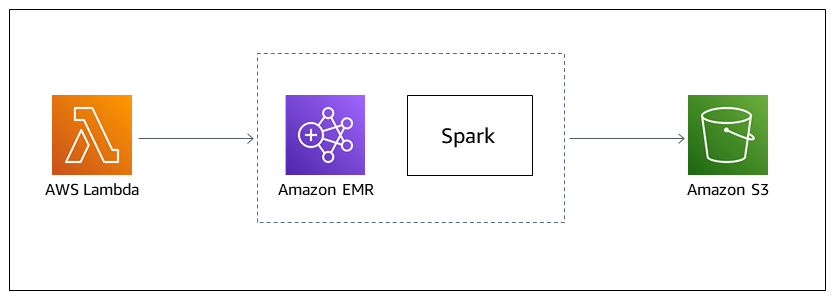
# Spark

|  |  |
| --- | --- |
| * SPARK is a formally defined computer programming language based on the Ada programming language, intended for the development of high integrity software used in systems where predictable and highly reliable operation is essential. * Apache Spark is an open-source unified analytics engine for large-scale data processing. * Spark provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. * Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, | PySpark is a great language for data scientists to learn because it enables scalable analysis and ML pipelines. ... Check the references which help to learn PySpark easier and faster. Feel free to access/use the code that I have written in the article by using below colab notebook and GitHub  <https://towardsdatascience.com/beginners-guide-to-pyspark-bbe3b553b79f> |
| * Spark relies on the fault tolerant HDFS (Hadoop Distributed File System) for large volumes of data. |  |

# PySpark and AWS

|  |  |
| --- | --- |
| To run PySpark, you use EMR. To launch EMR, you can use various options including the AWS console, awscli, or a Lambda function. You don't have to use Lambda, but you could if it makes sense e.g. the EMR cluster launch is triggered by data arriving in an S3 bucket. | **AWS Target technology stack:**   * Amazon EMR * AWS Lambda * Amazon S3 * Apache Spark |
| **To run PySpark, you use EMR:**  To launch EMR, you can use various options including the AWS console, awscli, or a Lambda function. You don't have to use Lambda, but you could if it makes sense e.g. the EMR cluster launch is triggered by data arriving in an S3 bucket.Jun 3, 2020  [Launch a Spark job in a transient EMR cluster using a Lambda function](https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/launch-a-spark-job-in-a-transient-emr-cluster-using-a-lambda-function.html) This link contains the project plan to build the stack/architecture |  |

## PySpark AWS Architecture



# Amazon EMR - Elastic MapReduce

|  |  |
| --- | --- |
| * Amazon EMR (previously called Amazon Elastic MapReduce) is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. * Using these frameworks and related open-source projects, you can process data for analytics purposes and business intelligence workloads. * Amazon EMR also lets you transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) | * [Amazon EMR](https://aws.amazon.com/elasticmapreduce/) – This service page provides Amazon EMR highlights, product details, and pricing information. * [Tutorial: Getting started with Amazon EMR](https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html) – This tutorial gets you started using Amazon EMR quickly. |
| * Amazon EMR is based on Apache Hadoop, a Java-based programming framework that supports the processing of large data sets in a distributed computing environment. |  |

# Apache Hadoop

|  |  |
| --- | --- |
| * The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. * It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. * Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. |  |
|  |  |

# Reference

* [Launch a Spark job in a transient EMR cluster using a Lambda function](https://docs.aws.amazon.com/prescriptive-guidance/latest/patterns/launch-a-spark-job-in-a-transient-emr-cluster-using-a-lambda-function.html) This link contains the project plan to build the stack/architecture