Spark documentation

<https://spark.apache.org/docs/latest/api/python/>

PySpark is a Spark library written in Python to run Python applications using Apache Spark capabilities, using PySpark we can run applications parallelly on the distributed cluster (multiple nodes).

In other words, PySpark is a Python API for Apache Spark. Apache Spark is an analytical processing engine for large scale powerful distributed data processing and machine learning applications.

In real-time, PySpark has used a lot in the machine learning & Data scientists community; thanks to vast python machine learning libraries.

**Advantages of PySpark**

* PySpark is a general-purpose, in-memory, distributed processing engine that allows you to process data efficiently in a distributed fashion.
* Applications running on PySpark are 100x faster than traditional systems.
* You will get great benefits using PySpark for data ingestion pipelines.
* Using PySpark we can process data from Hadoop HDFS, AWS S3, and many file systems.
* PySpark also is used to process real-time data using Streaming and Kafka.
* Using PySpark streaming you can also stream files from the file system and also stream from the socket.
* PySpark natively has machine learning and graph libraries.

## PySpark Modules & Packages

* PySpark RDD ([pyspark.RDD](https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.RDD.html))
* PySpark DataFrame and SQL ([pyspark.sql](https://spark.apache.org/docs/3.1.1/api/python/reference/api/pyspark.sql.DataFrame.html))
* PySpark Streaming ([pyspark.streaming](https://spark.apache.org/docs/2.2.0/api/python/pyspark.streaming.html))
* PySpark MLib ([pyspark.ml](https://spark.apache.org/docs/latest/api/python/pyspark.ml.html), [pyspark.mllib](https://spark.apache.org/docs/2.0.0/api/python/pyspark.mllib.html))
* PySpark GraphFrames ([GraphFrames](https://graphframes.github.io/graphframes/docs/_site/index.html))

We will be starting with spark , we will try to integrate python with spark

We are going to use pyspark for doing some data analysis , we will be working on some machine learning models in pyspark

Hadoop , in to work on aws (amazon web service)

We will be creating accounts in aws (free tier accounts), you should have credit or debit card

Work on servers, these ec2 instances (elastic cloud compute ), there we will be launching our linux machines , one virtual machine linux, we will be installing java , because Hadoop requires java ,install Hadoop , setting up the configuration , then we will be discussing small examples using map reduce concepts in Hadoop

In the background we wiil be using small examples of programs written n java and then we are going to apply them on data

Word count, mean of words present in a a text , calculate the pi value examples

Databricks, platform where you can directly launch your spark +python kernel and you can start working using your pyspark concepts

Just create one free account, we can start working (no need of having any credit card and debit card )

<https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/>

1)yum search jdk

2)sudo yum install java-1.8.0-openjdk -y

3)which java

4)java -version

5)pwd(present working directory)

6)clear (clear the screen)

7)ssh-keygen

8)ll -a(lookup in the directory)

9)cd .ssh

10)ll(lookup in ssh directory)

11)cat id\_rsa.pub >> authorized\_keys

12)cat authorized\_keys

13)ssh localhost

14)yes

15) exit

16)cd ~ (home directory)

17)ll (list directory)

18) [**https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz**](https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz)

19)curl -O <https://dlcdn.apache.org/hadoop/common/hadoop-3.3.6/hadoop-3.3.6.tar.gz>

20) ll (list directory)

21) tar -xzf hadoop-3.3.4.tar.gz

22) ll (list directory)

23)ll hadoop-3.3.4

24)mv hadoop-3.3.4/ hadoop

25)ll

Install Hadoop

26)which java

27)ll /usr/bin/java

28)ll /etc/alternatives/java

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.342.b07-1.amzn2.0.1.x86\_64/jre

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.402.b06-1.amzn2.0.1.x86\_64/jre

/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.402.b06-1.amzn2.0.1.x86\_64/jre

cd hadoop

29)vim etc/hadoop/hadoop-env.sh

30)JAVA\_HOME change

Install HDFS

31)vim etc/hadoop/core-site.xml

32)vim etc/hadoop/hdfs-site.xml

33)./bin/hdfs namenode -format

34)./sbin/start-dfs.sh

35)./bin/hdfs dfs -ls /

36)./bin/hdfs dfs -mkdir -p /user/cloud\_user

37)./bin/hdfs dfs -ls /

38) ./bin/hdfs dfs -ls /user

39)./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar

40) ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar wordcount /user/cloud\_user/NOTICE.txt notice\_wordcount

41)./bin/hdfs dfs -cat /user/ec2-user/notice\_wordcount/\*

42) ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar pi 10 100

43) ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar wordmean /user/cloud\_user/NOTICE.txt notice\_wordmean

44) ./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.4.jar wordmedian /user/cloud\_user/NOTICE.txt notice\_wordmedian

45) ./bin/hdfs dfs -put ../hadoop/NOTICE.txt /user/cloud\_user/NOTICE.txt

./bin/hadoop jar share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar wordstandarddeviation /user/cloud\_user/NOTICE.txt notice\_wordstandarddeviation

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

R Nageswar Rao

Machine learning using python

Core python programming

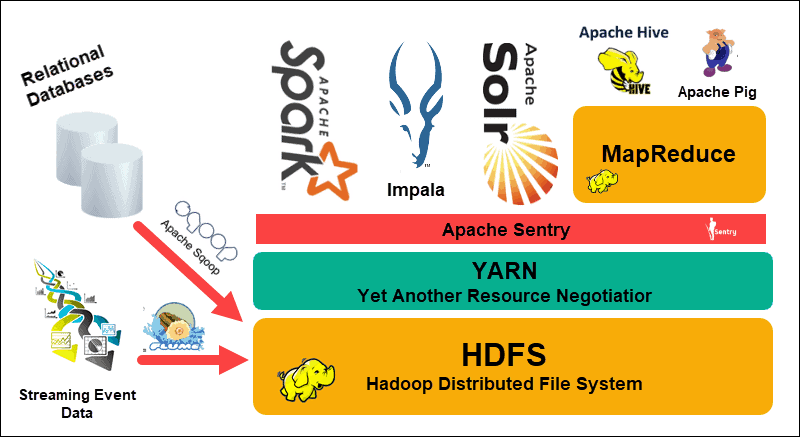
Meenu kohli

Python interview questions

Data structures and algorithm thinking with python – Narasimha karumanchi

Hhh

Ttt



Htt

Hth

Tth

Tht

Hht

thh