Apache Spark

Spark is

1)open source

2)parallel

3)Scalable

4)Supports In-memory computation

4)Lazy evaluation

5)Fault Tolerant

6)Immutability

7)Partitioning

8)Persistence

9)Coarse Grained Operations

Spark is used for data analytics much like map-reduce but better than that

Unlike map-reduce which shuffles files on disk spark works in memory it’s fast as well

Features:

1)High-Speed

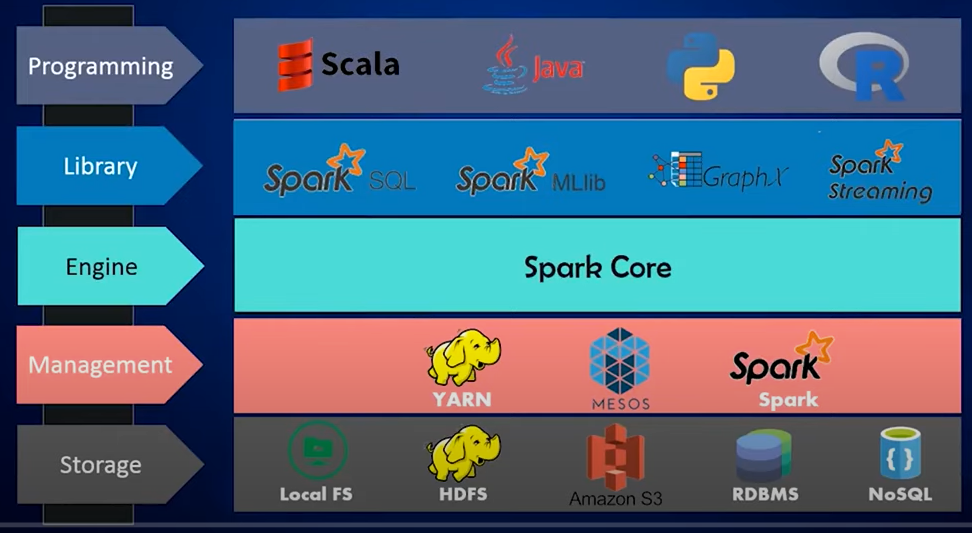
2)Powerful caching

3)Deployment is easy

4)Real-time computation

5)Supports multiple languages

Spark Eco-System:



RDD:

To reduce number of operation performed through hdfs can be achieved through in-memory data computation that’s where RDD’s come into picture RDD’s are fault tolerant fundamental data structure of spark which can handle both schema and schema less structures

Data read into spark🡪Same Data read into RDD🡪 Transformation of data🡪 New RDD is created depending on need🡪The data obtained is stored to persistent storage

Objects in RDD are immutable

RDD has logical partitions and the data is present in multiple partitions

so the multiple data copies are present and when one data is lost in some partition other partition data will help

RDD can be created by

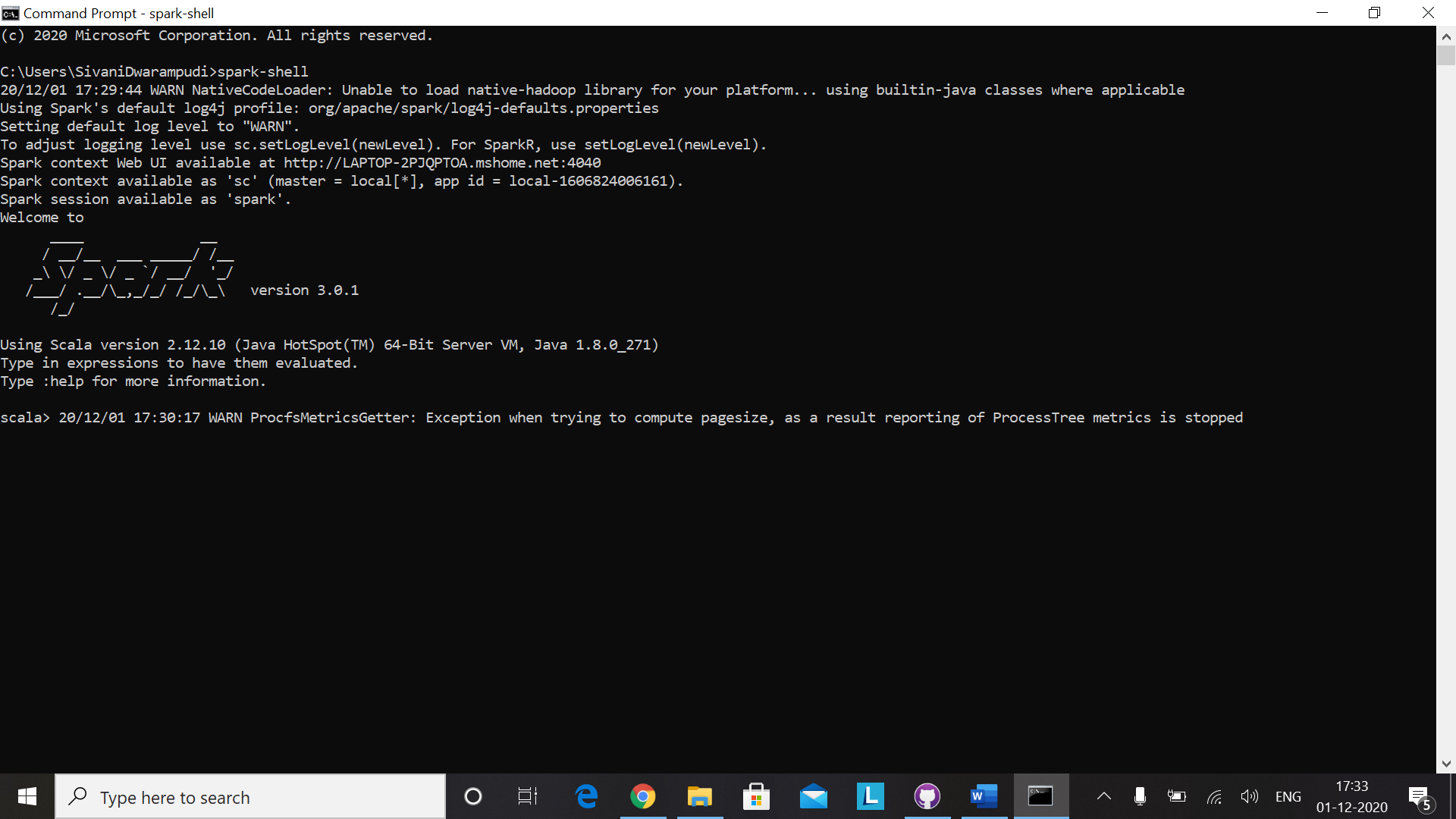
1)parallelized collections

2)From existing RDD’s

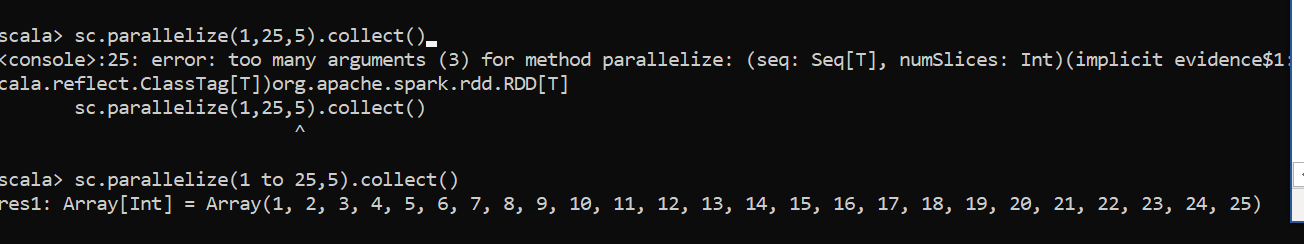
3)External Data

Steps for creating RDD using Parallelized collections:

Start Spark-Shell:

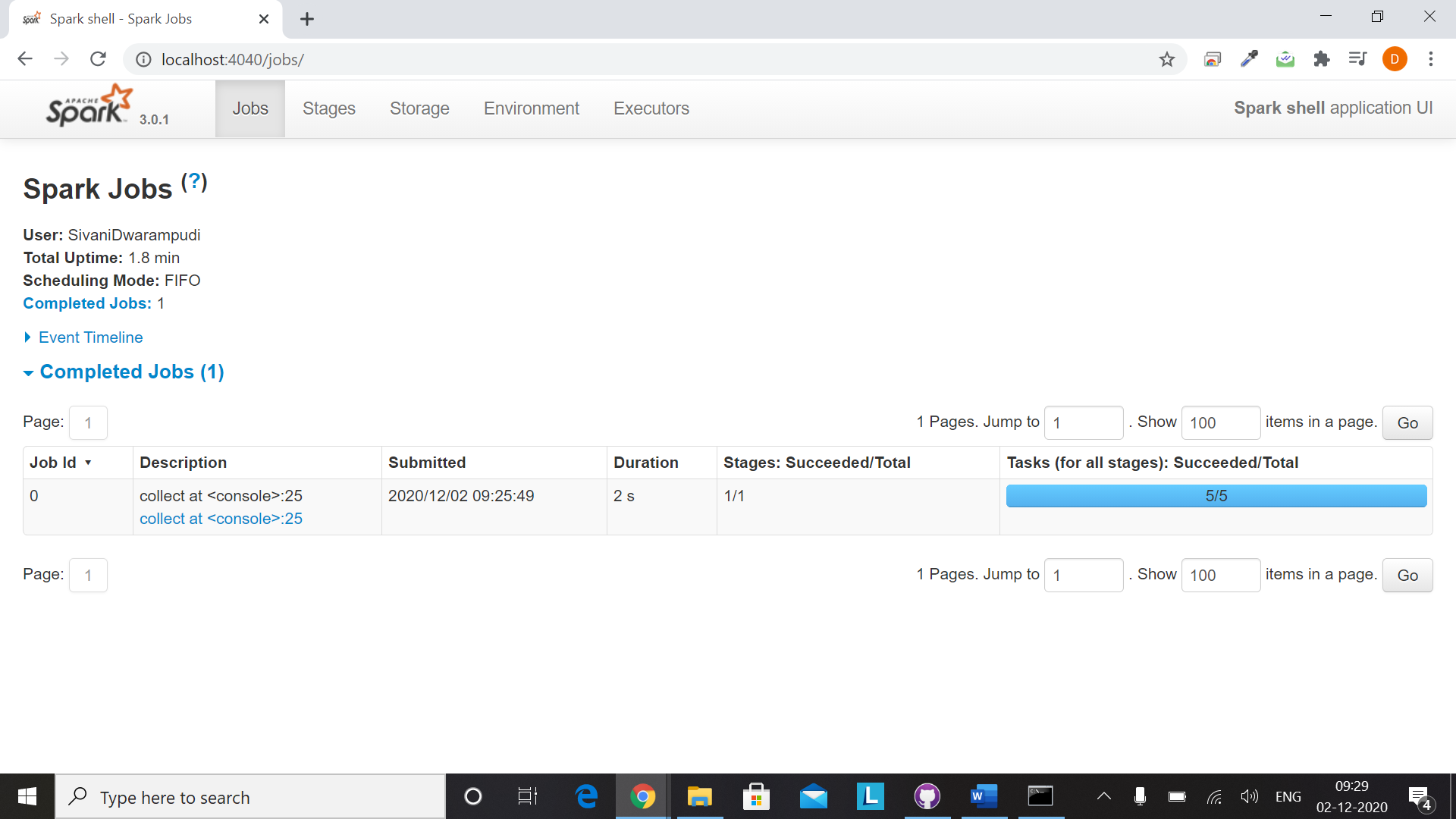


Create RDD:

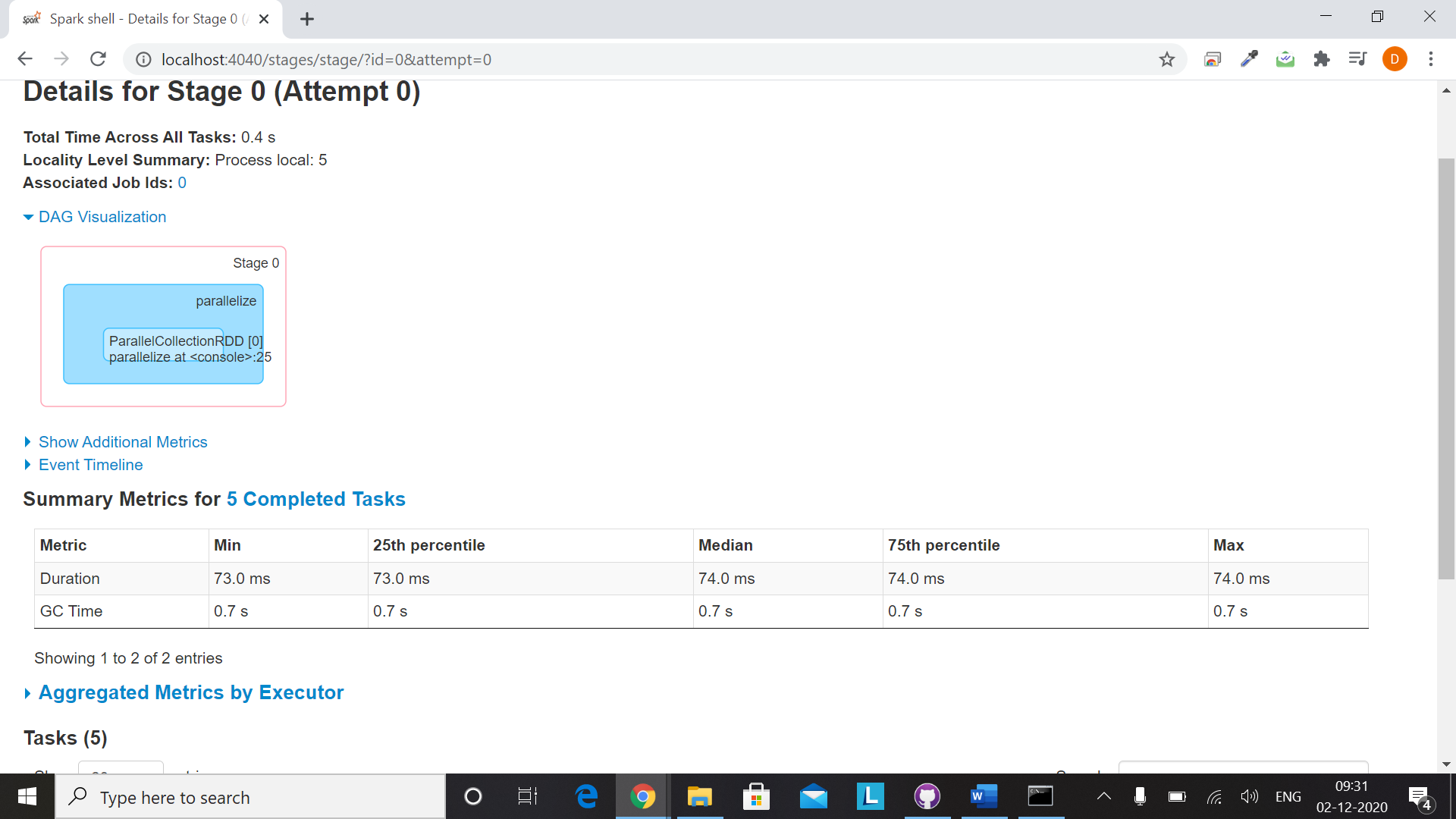


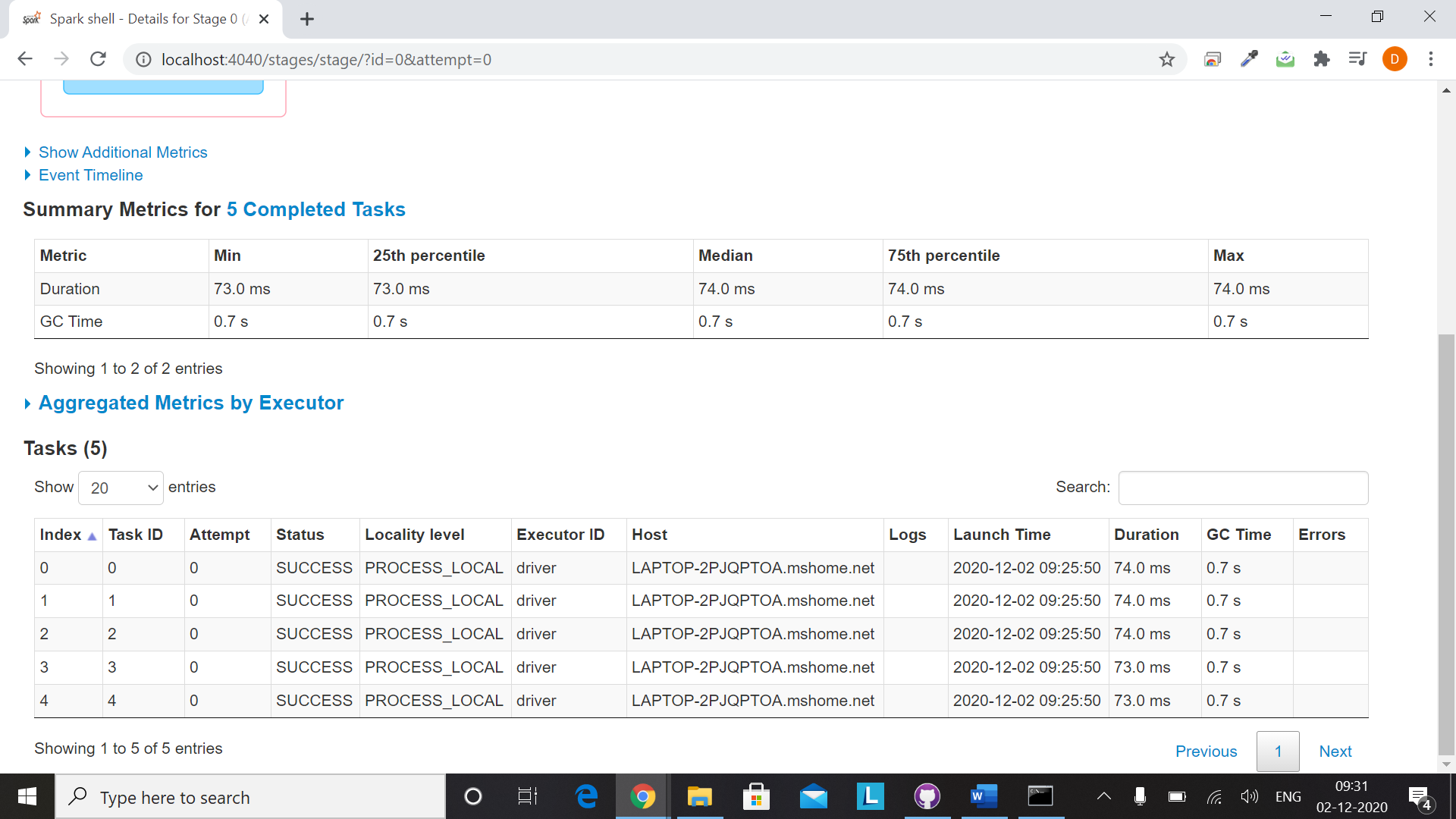
To view the content on web:

Localhost:4040 is the address to be typed on the web



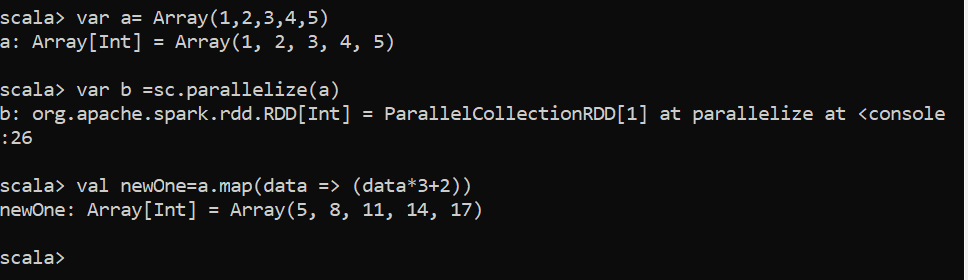
To get more information such as DAG click on the RDD that you’ve created



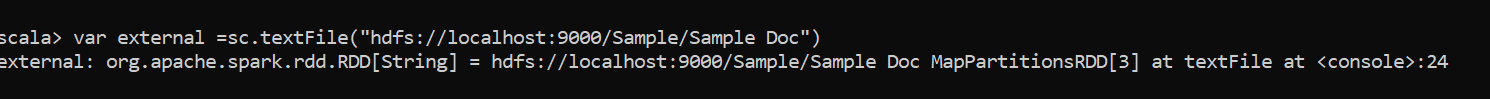


The status SUCCESS in the above screenshot mentions that all 5 tasks are executed successfully

RDD from existing RDD:



RDD from external file sources:

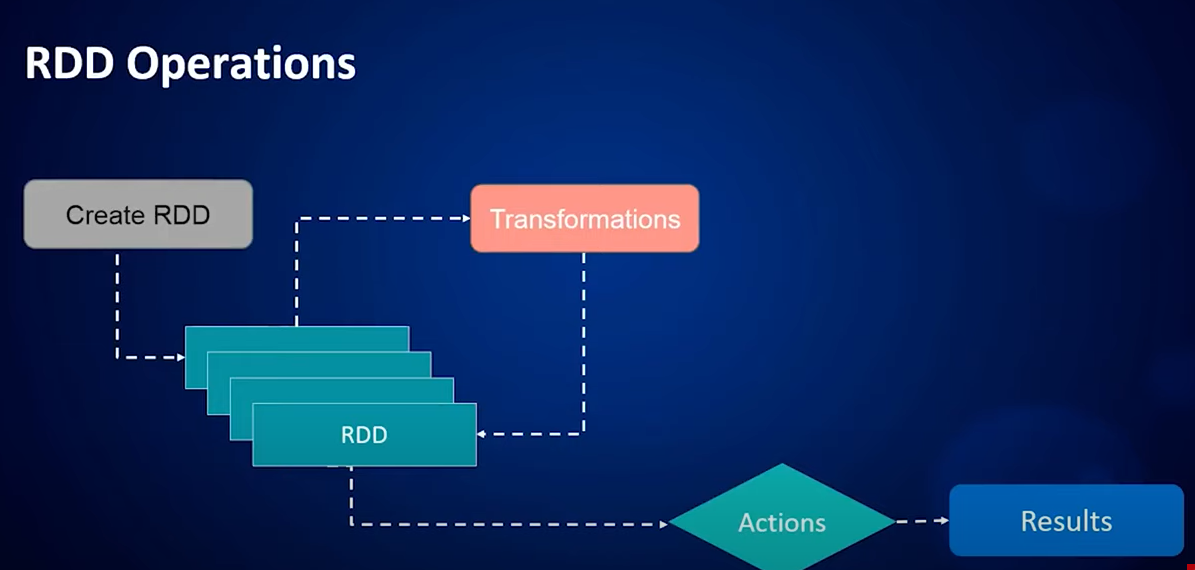


RDD operations:

1)Transformations

2)Actions

As RDD are immutable when we perform transformation that will be executed on old one and returns new one



Spark Architecture:

Spark Consists of RDD and DAG



Spark architecture primarily consists of spark context which is the entry point for any spark application just like how we establish data base connection before writing code in database to perform any operation we need to have spark context such that all functionalities related to our spark code

Spark User code🡪DAG🡪physical execution plan🡪Task🡪sent to cluster

“Install Hadoop before”

Creation of spark application using scala:

Task is to count the number of word in file which is in hdfs directory

==================================================================================

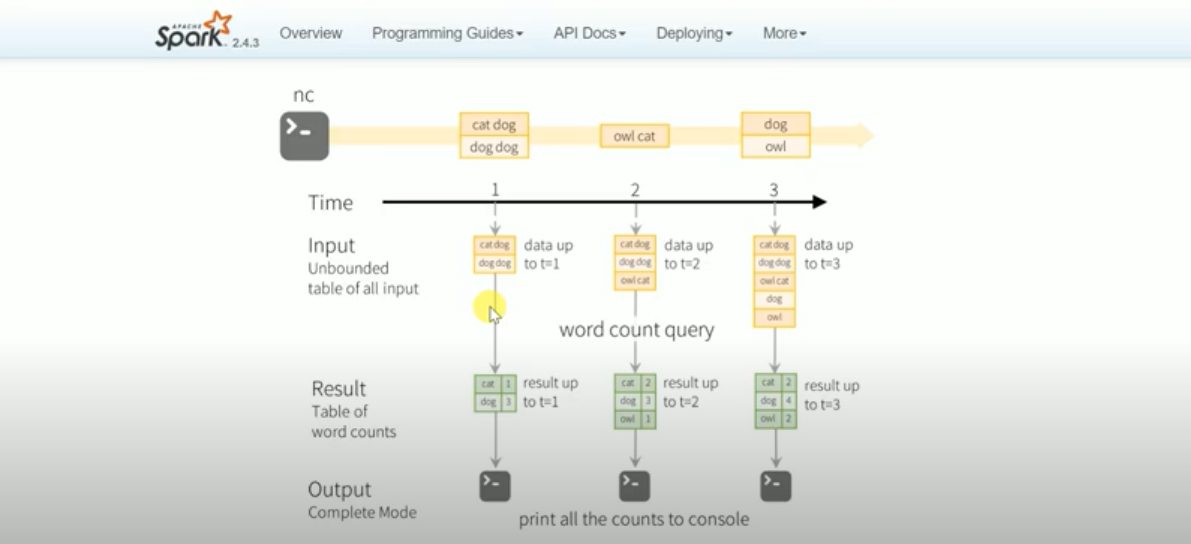
Kafka and Spark

Batch processing: Consumes data from different sources over a period data time then calculation followed by report generation

Real time Processing: **Real**-**time data processing** is the execution of **data** in a short **time** period, providing near-instantaneous output. The **processing** is done as the **data** is inputted, so it needs a continuous stream of input **data** in order to provide a continuous output.

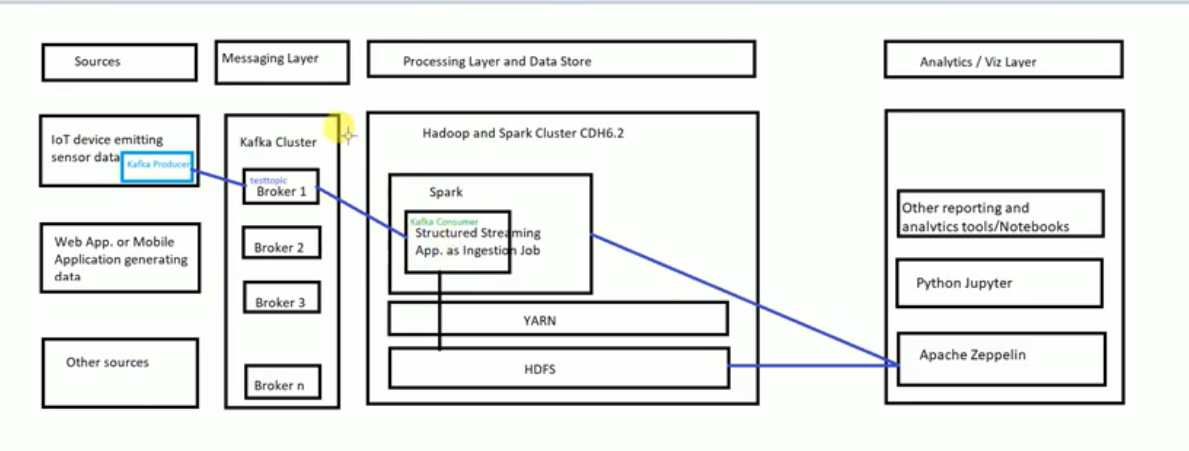
Component named structured streaming in spark supports this rea time processing

In structured streaming we data from sockets or kafka etc.



Resulted data from streaming data is written to sink(destination)

Three modes: complete, update, append



We gonna do:

Some data that is generated from kafka producer sent to kafka topic then ingestion is performed structured streaming job

Kafka installation in windows:

Download tar file

<https://www.apache.org/dyn/closer.cgi?path=/kafka/2.6.0/kafka_2.13-2.6.0.tgz>

Unzip using 7 Zip(preferred)

Create folder in c drive

Copy the unzip file to that

Kafka steps:

1)Start Zookeeper

bin\windows\zookeeper-server-start.sh config\zookeeper.properties

2)Start Kafka Broker

bin\windows\kafka-server-start.sh config/server.properties

3) Create topic

bin\windows\kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic test List topic bin\windows\kafka-topics.sh --list --zookeeper localhost:2181

4)Start Producer

bin\windows\kafka-console-producer.sh --broker-list localhost:9092 --topic test

a) Send message

b) Receive message

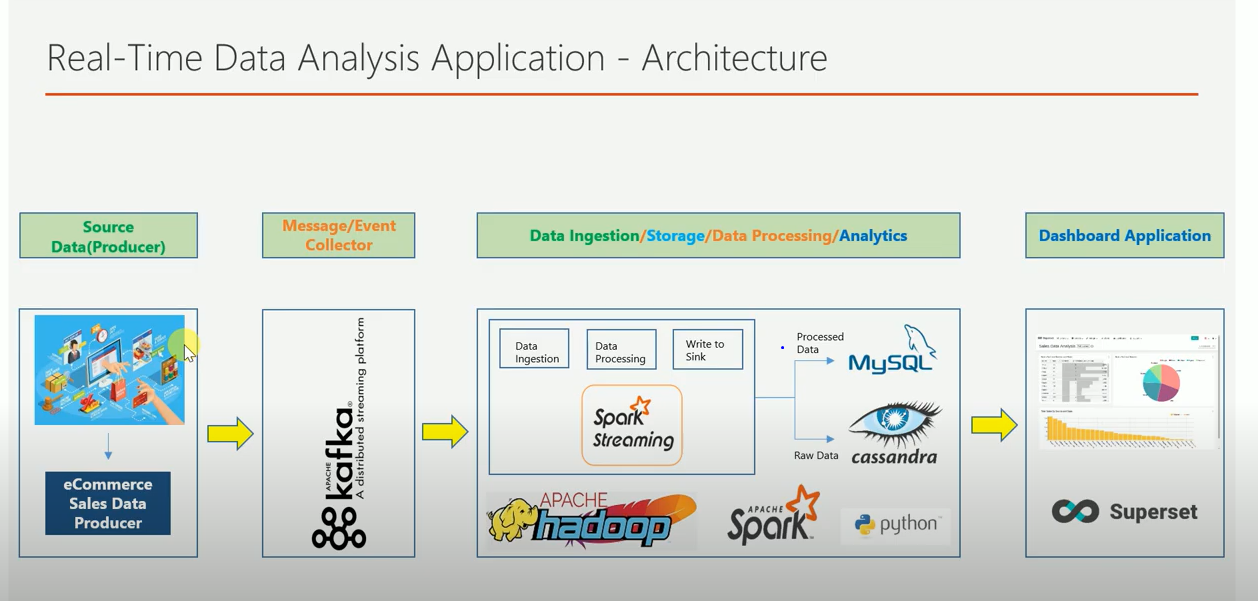
5)Start Consumer

bin\windows\kafka-console-producer.sh --broker-list localhost:9092 --topic test

The messages you sent in producer will be displayed in consumer

**Real-Time Data Analysis Application-Architecture**

**Architecture:**



How does it work?

A simple Ecommerce application which will generate series of events is the data producer

When x person purchases some product from ecommerce website then the transaction gets stored in RDBMS

Data Flow:

Ecommerce website 🡪Kafka 🡪Messaging system(Data storage, processing layer)🡪Dashboard for understanding analysis

Spark structured streaming is used in this project the major advantage of spark is we can store data in form of data frames