Table of Contents

REQUIREMENTS

1. STACK FUNCTIONALITIES

The demo must have the following Stack components

1.1. Apache Spark with Scala

The Apache Spark engine must be installed, configurable and fully operational.

1.1.1. Code examples for read operations

The tool must contain a simple example of reading db data

1.1.1.1. Reading from files

The demo must have a working example of reading data form files on POSIX file system

1.1.1.2. Reading of RDBMS into DataFrames

The demo must have a working example of reading RDMS data into dataframes objects

1.1.2. Code examples for data tranformation operations

1.1.2.1. Filter dataframes objects by attribute name and value

Filter dataframes objects by attribute name and value

1.1.3. Code examples for write operations operations

1.1.3.1. Saving DataFrame objects data files

Saving DataFrame objects data files

1.1.3.2. Saving DataFrame objects data files on POSIX file system

The demo must have example for saving DataFrame objects data files on POSIX file system

1.1.3.3. Saving DataFrame objects data files on HDFS

The demo must have example for saving DataFrame objects data files on HDFS

1.2. Scala

The Scala run-time must be installed and fully operational.
The main logic of the demo app must be implemented with the Scala language.

1.3. Python

The demo must have a small bunch of Python code.

1.3.1. Hadoop

The demo must have installed Hadoop instances

2. RELIABILITY AND STABILITY

2.1. Logging

The console app should demonstrate proper logging to STDOUT and file

3. PERFORMANCE

4. SCALABILITY

4.1. Scalabiity for deployment

The app should demonstrate scalability to deployment.

4.2. Feature scalabiity

The app should be written so that new features and code examples could be added easily and efficiently.