CLOUDERA
Educational Services

Preparing with Cloudera Data Engineering

Instructor Guide

DENG-254 220331

Agenda

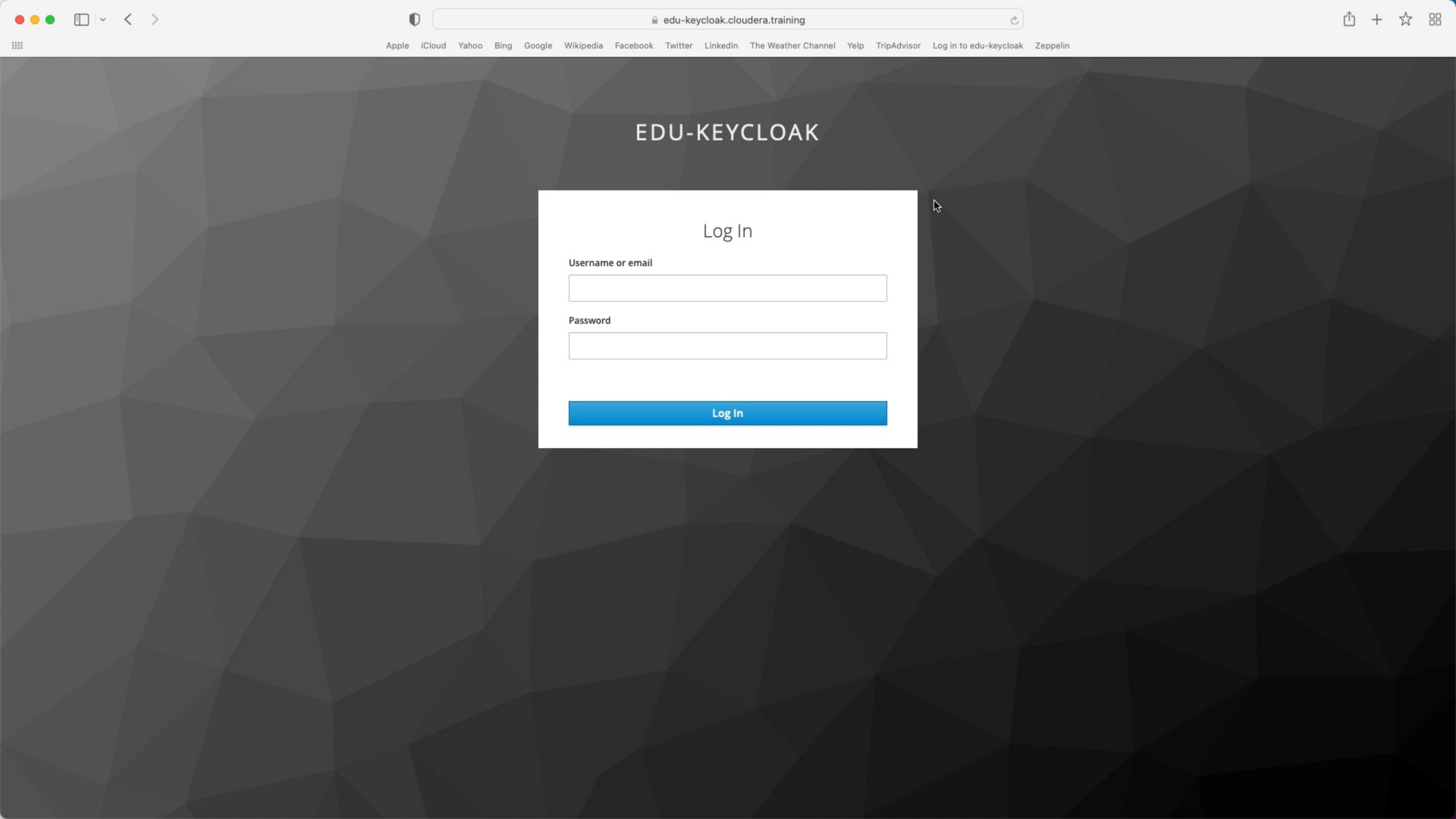
- Why this new version?
- The environment
- The notebooks
- The content
- Timings
- Troubleshooting
- What’s next?
- Questions

CLOUDERA
Educational Services

Why this new version?

Too Much RDDs

- The content is outdated, a lot of stuff predates 2016 (Spark 1.5). This translates to a lot of RDD based content which nobody in their right minds uses nowadays. (RDD appears in 160 slides vs 71 in the new version)
- The labs rely on stark interpreters which again nobody uses professionally.
- It does not foster HDFS best practices: HDFS permissions are disabled (I guess for convenience).
- The Common Patterns in Spark Data Processing is a joke made of the description of the PageRank and kMeans algorithms that exist out of the box in recent versions and an awkward introduction to Machine Learning
- No content on Hive, although Hive and Spark have a long history of being closely integrated



EDU-KEYCLOAK

Log In

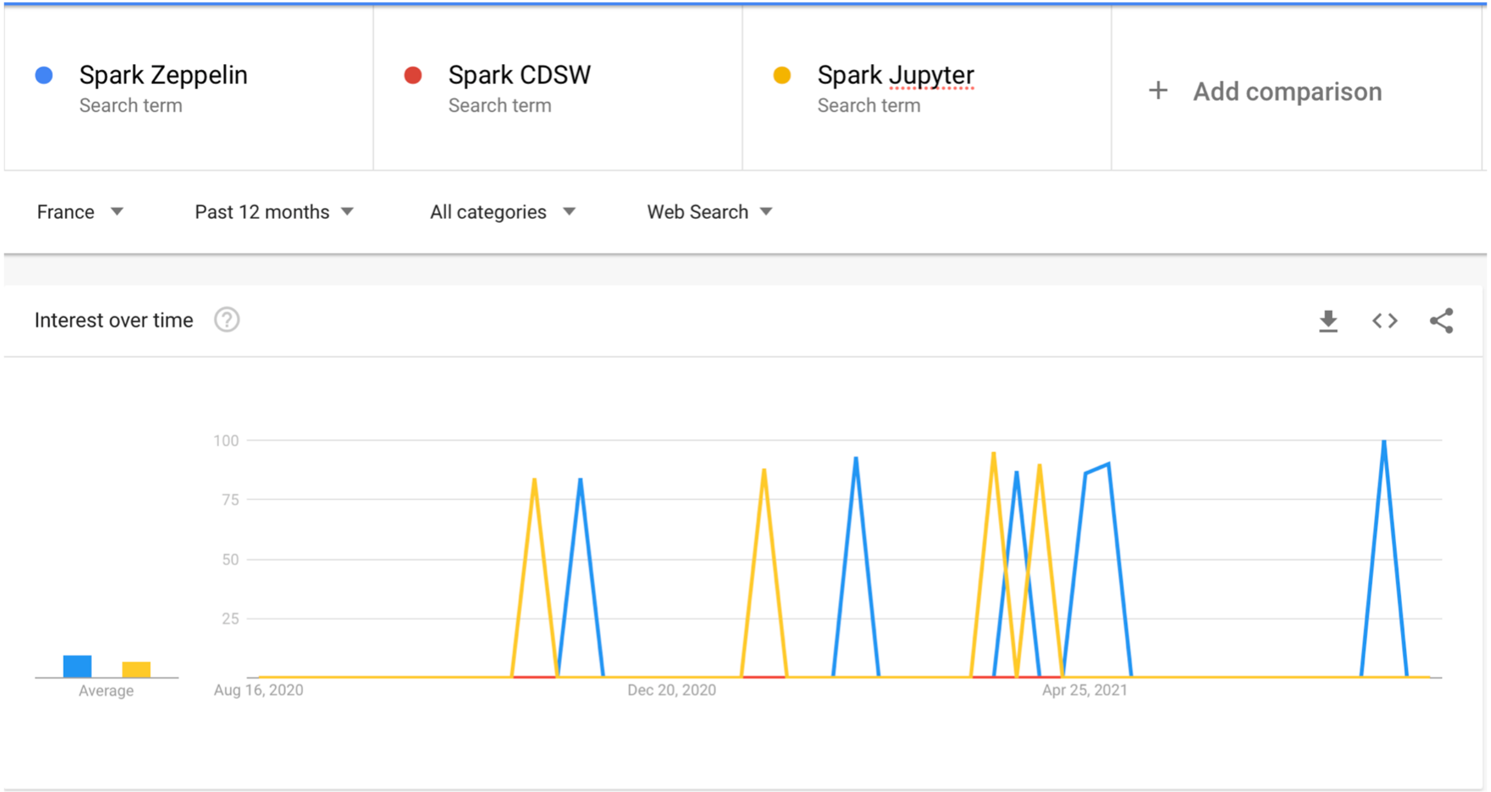
No Commonality with Spark Performance

- Spark performance issues are much more likely to occur in Data Engineering than in Data Science stages
- This course should use the same environment so that both contents can be fungible
- Clients will likely request mixed custom courses that will leverage both courses

Spark 3.0

- Spark 3.0 contains major performance related improvements that promise to perform up to 17* faster
- If we do not jump on this train fast our content will be quickly outdated

Zeppelin is available in CDP and is part of the Spark ecosystem



● Spark Zeppelin
Search term

● Spark CDSW
Search term

● Spark Jupyter
Search term

+ Add comparison

France ▾ Past 12 months ▾ All categories ▾ Web Search ▾

Interest over time

CLOUDERA
Educational Services

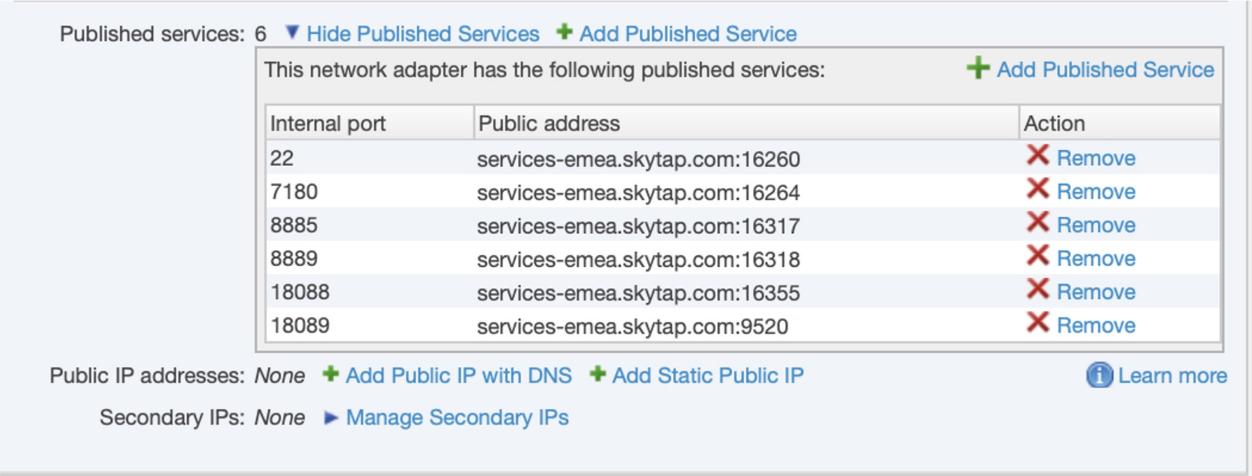
The environment

The Environment

- Started from the latest Spark Application Performance Tuning template
- Created a /home/training/training-materials/dev home folder
- Updated to CDP PVC BE 7.1.7,
- installed Spark 3.1,
- Installed and built the data in HDFS
- Created the Hive tables
- Installed the required jar files
- Installed Arrow
- Installed jq
- Uploaded the notebooks in Zeppelin
- Installed the images for the notebooks

Published Services

- ssh
- Cloudera Manager
- Zeppelin
- Hue
- Spark 2 History Server
- Spark 3 History Server



Published services: 6

This network adapter has the following published services:

Internal port	Public address	Action
22	services-emea.skytap.com:16260	✕ Remove
7180	services-emea.skytap.com:16264	✕ Remove
8885	services-emea.skytap.com:16317	✕ Remove
8889	services-emea.skytap.com:16318	✕ Remove
18088	services-emea.skytap.com:16355	✕ Remove
18089	services-emea.skytap.com:9520	✕ Remove

Public IP addresses: *None*

Secondary IPs: *None*

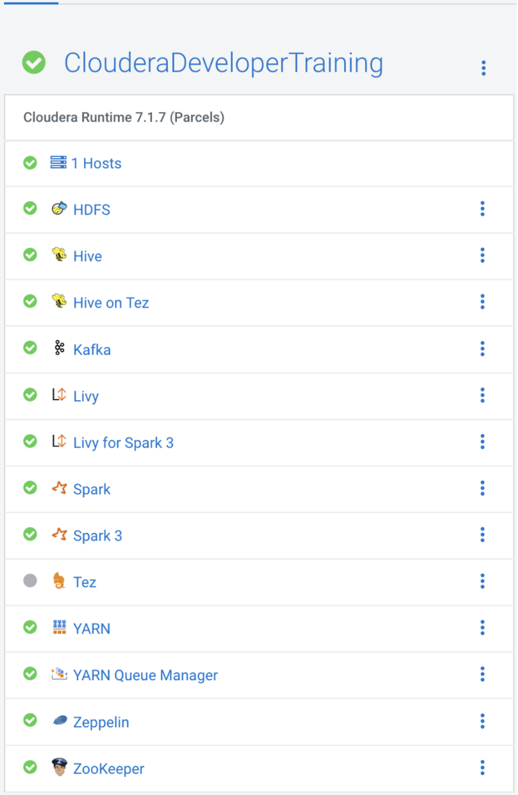
/home/training/training_materials/dev

- data
  - The datasets used in the legacy labs
- exercises
  - Spark-application
  - yarn
- scripts
  - Utility scripts
  - Don’t trust the setup.sh script to build a new environment, it hasn’t been tested. Start one from Skytap instead.
- notebooks
  - Zip file of the current version of notebooks

The entire dev folder is zipped into a single master file: `zip -r dev-20210816.zip dev/*`

Available Services

- Removed unnecessary services to save on resources
  - Impala
  - Hue
  - Oozie
  - NiFi
  - NiFi Registry
- 0 warning starting point
- HDFS permissions are enabled



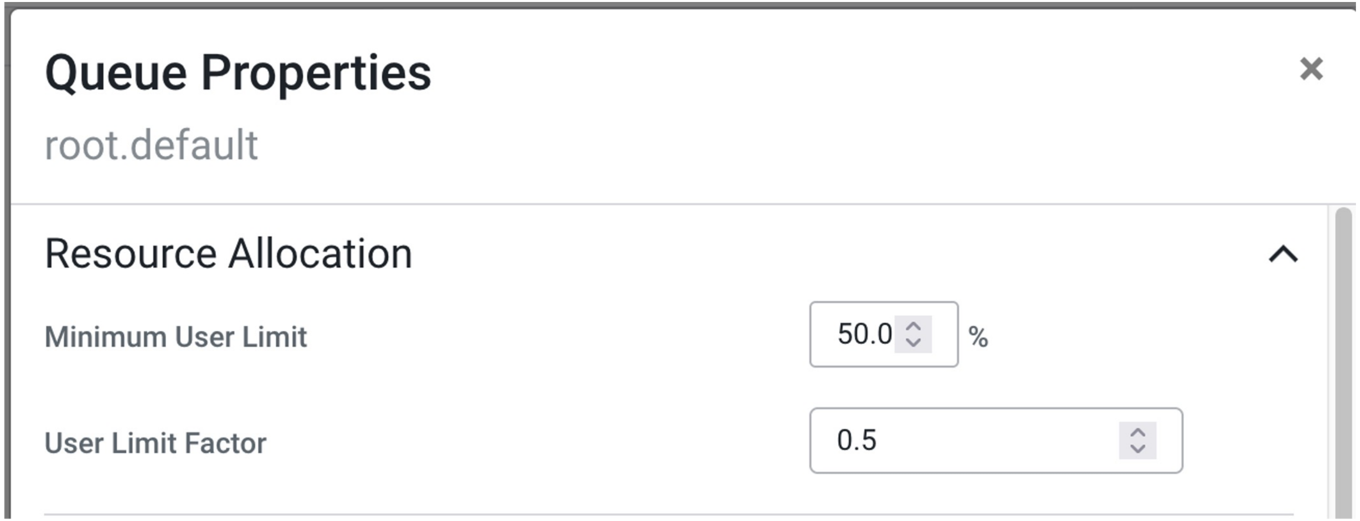
ClouderaDeveloperTraining

Cloudera Runtime 7.1.7 (Parcels)

- ✓ 1 Hosts
- ✓ HDFS
- ✓ Hive
- ✓ Hive on Tez
- ✓ Kafka
- ✓ Livy
- ✓ Livy for Spark 3
- ✓ Spark
- ✓ Spark 3
- ● Tez
- ✓ YARN
- ✓ YARN Queue Manager
- ✓ Zeppelin
- ✓ ZooKeeper

Resource Management

- You should be able to run two applications simultaneously in the default queue
- As a good practice, stop the services you do not need



Queue Properties

root.default

Resource Allocation

Minimum User Limit 50.0 %

User Limit Factor 0.5

CLOUDERA
Educational Services

The notebooks

Notebooks Design Principles

- A single artefact without internal duplication
  - to simplify maintenance
- That contains the solution, instructions yet that allows the student to type his code
  - for effective learning
- That runs without errors
  - to enable easy regression and performance testing

Notebooks Structure

- All the Zeppelin notebooks share the same structure
  - About
  - Setup
  - Demo or Lesson
  - Lab *
  - Result
  - Solution
  - Tear down
  - Footer

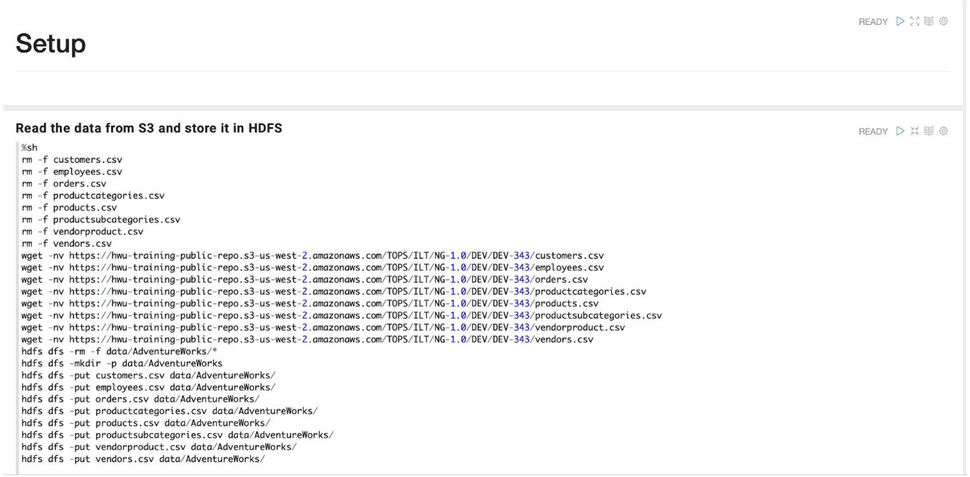
* This part is optional for Demo only notebooks

About

- High level information that mainly helps decide whether this is the notebook you are looking for or not
  - Objective: <Short description of the notebook>
  - Files locations:
  - Successful outcome:
  - Before you begin: <Dependencies>
  - Related lessons:
  - Copyright
- Not always rigorously filled to be honest

Setup

- This is the section in which all the preparation required for the lab should be carried out.
- It can contain code to retrieve the data required for the lab as well as catch up code to make the notebook independent from previous labs.
- In order for the notebook to always run without errors, special care should be taken when creating directories or files such as deleting them before recreating them.



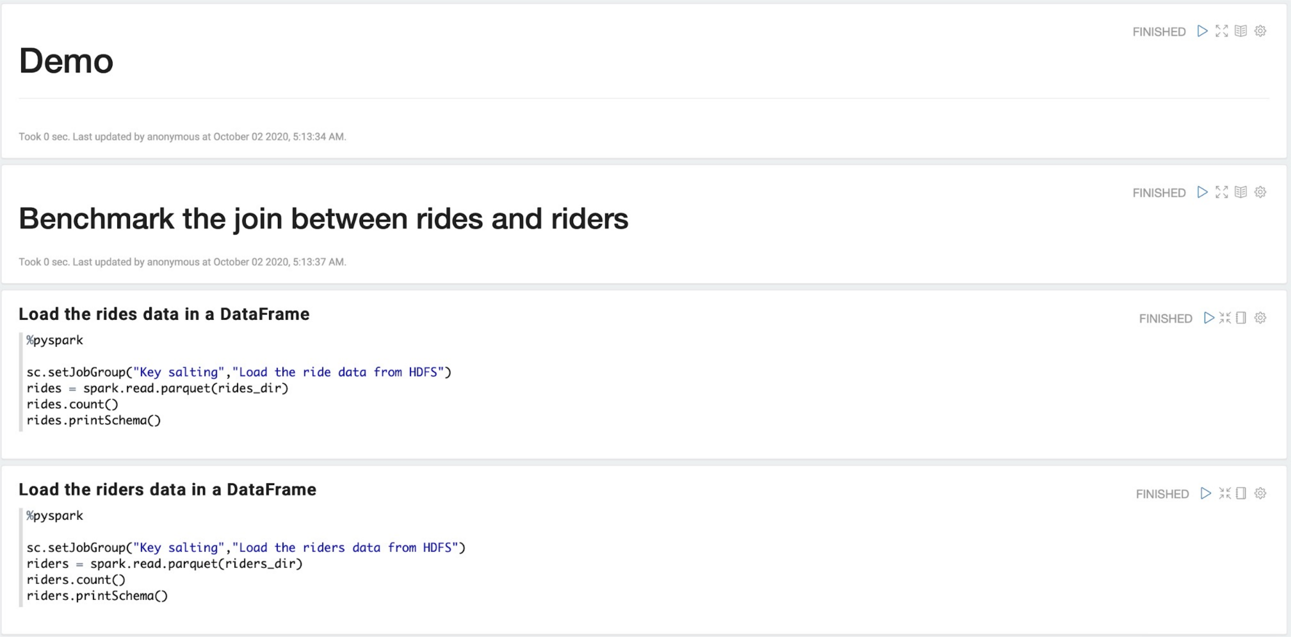
Setup

Read the data from S3 and store it in HDFS

```
%sh
rm -f customers.csv
rm -f employees.csv
rm -f orders.csv
rm -f productcategories.csv
rm -f products.csv
rm -f productsubcategories.csv
rm -f vendorproduct.csv
rm -f vendors.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/customers.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/employees.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/orders.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/productcategories.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/products.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/productsubcategories.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/vendorproduct.csv
wget -nv https://hwu-training-public-repo.s3-us-west-2.amazonaws.com/TOPS/ILT/NG-1.0/DEV/DEV-343/vendors.csv
hdfs dfs -rm -f data/AdventureWorks/*
hdfs dfs -mkdir -p data/AdventureWorks
hdfs dfs -put customers.csv data/AdventureWorks/
hdfs dfs -put employees.csv data/AdventureWorks/
hdfs dfs -put orders.csv data/AdventureWorks/
hdfs dfs -put productcategories.csv data/AdventureWorks/
hdfs dfs -put products.csv data/AdventureWorks/
hdfs dfs -put productsubcategories.csv data/AdventureWorks/
hdfs dfs -put vendorproduct.csv data/AdventureWorks/
hdfs dfs -put vendors.csv data/AdventureWorks/
```

Demo or Lesson

- This is the section where the instructor walks the students through each paragraph to illustrate the topic of the notebook.



Demo

Took 0 sec. Last updated by anonymous at October 02 2020, 5:13:34 AM.

Benchmark the join between rides and riders

Took 0 sec. Last updated by anonymous at October 02 2020, 5:13:37 AM.

Load the rides data in a DataFrame

```
%pyspark

sc.setJobGroup("Key salting","Load the ride data from HDFS")
rides = spark.read.parquet(rides_dir)
rides.count()
rides.printSchema()
```

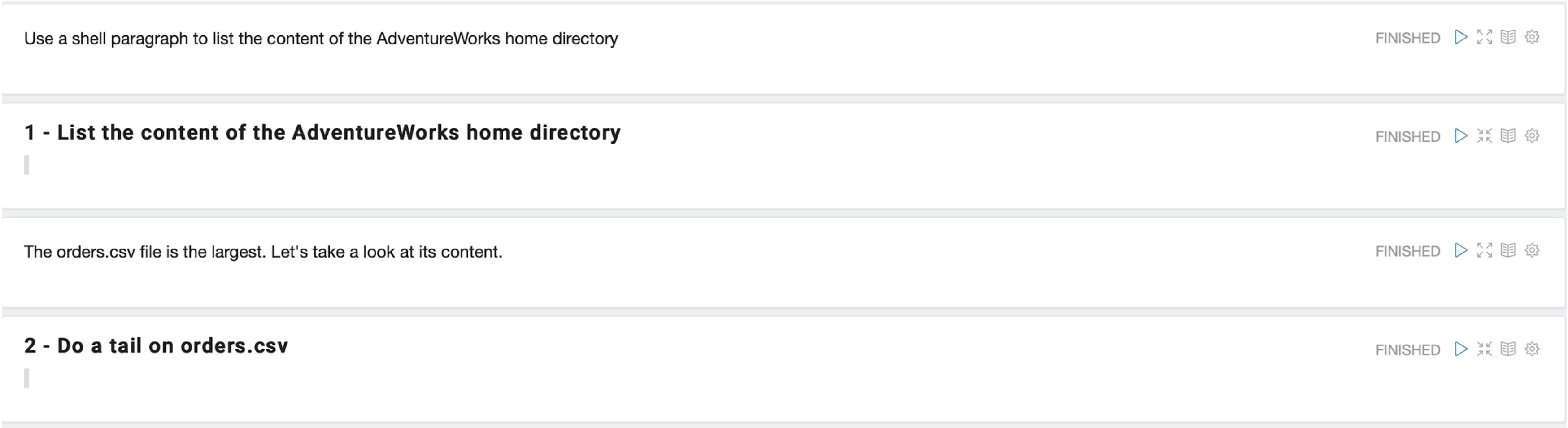
Load the riders data in a DataFrame

```
%pyspark

sc.setJobGroup("Key salting","Load the riders data from HDFS")
riders = spark.read.parquet(riders_dir)
riders.count()
riders.printSchema()
```

Lab

- This is the section where the student will try to perform the lab steps.
  It should contain the lab instructions in markdown paragraphs interspersed with empty code paragraphs with numbered titles



Use a shell paragraph to list the content of the AdventureWorks home directory

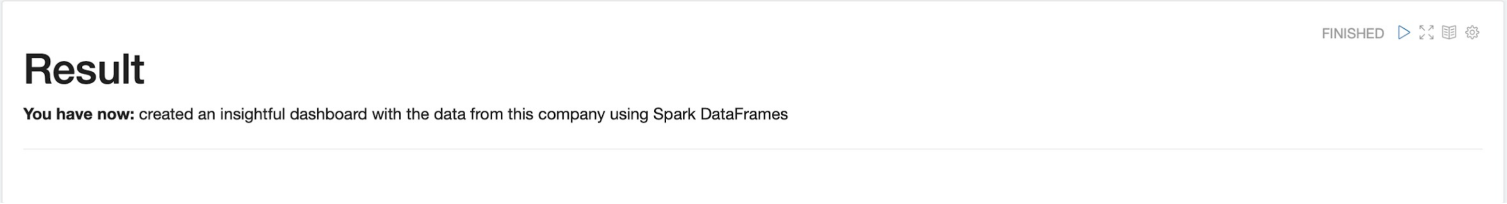
1 - List the content of the AdventureWorks home directory

The orders.csv file is the largest. Let's take a look at its content.

2 - Do a tail on orders.csv

Result

- This section summarizes what the student has just achieved.

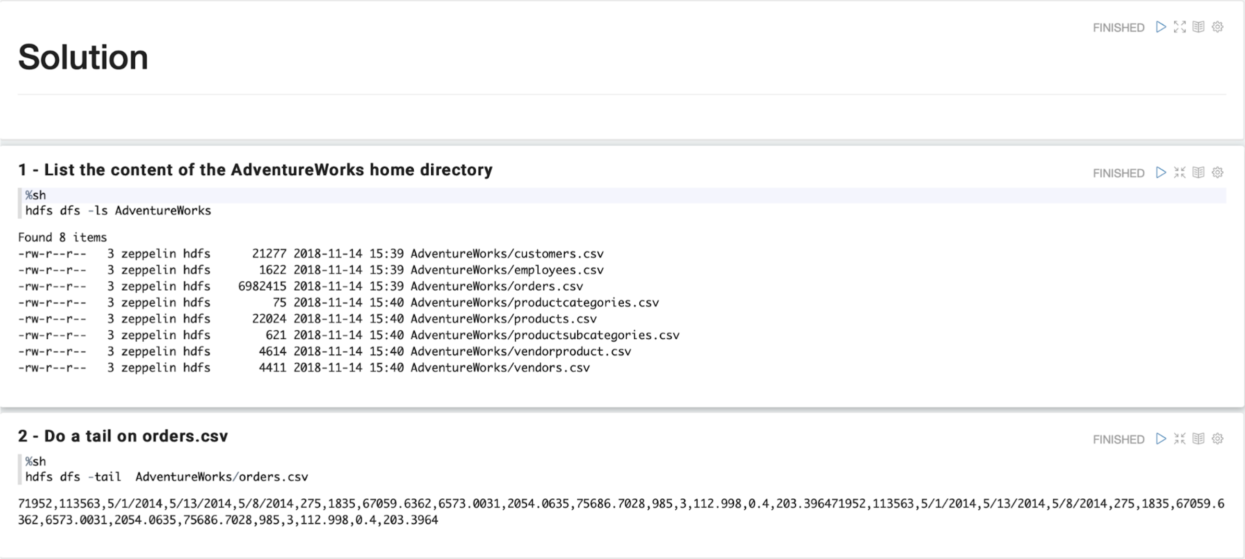


Result

You have now: created an insightful dashboard with the data from this company using Spark DataFrames

Solution

- This is the section where the student can look up solutions to the lab steps using the matching numbered titles. It contains only code paragraphs.



Solution

1 - List the content of the AdventureWorks home directory

```
%sh
hdfs dfs -ls AdventureWorks
```

```
Found 8 items
-rw-r--r--   3 zeppelin hdfs      21277 2018-11-14 15:39 AdventureWorks/customers.csv
-rw-r--r--   3 zeppelin hdfs       1622 2018-11-14 15:39 AdventureWorks/employees.csv
-rw-r--r--   3 zeppelin hdfs    6982415 2018-11-14 15:39 AdventureWorks/orders.csv
-rw-r--r--   3 zeppelin hdfs         75 2018-11-14 15:40 AdventureWorks/productcategories.csv
-rw-r--r--   3 zeppelin hdfs      22024 2018-11-14 15:40 AdventureWorks/products.csv
-rw-r--r--   3 zeppelin hdfs        621 2018-11-14 15:40 AdventureWorks/productsubcategories.csv
-rw-r--r--   3 zeppelin hdfs       4614 2018-11-14 15:40 AdventureWorks/vendorproduct.csv
-rw-r--r--   3 zeppelin hdfs       4411 2018-11-14 15:40 AdventureWorks/vendors.csv
```

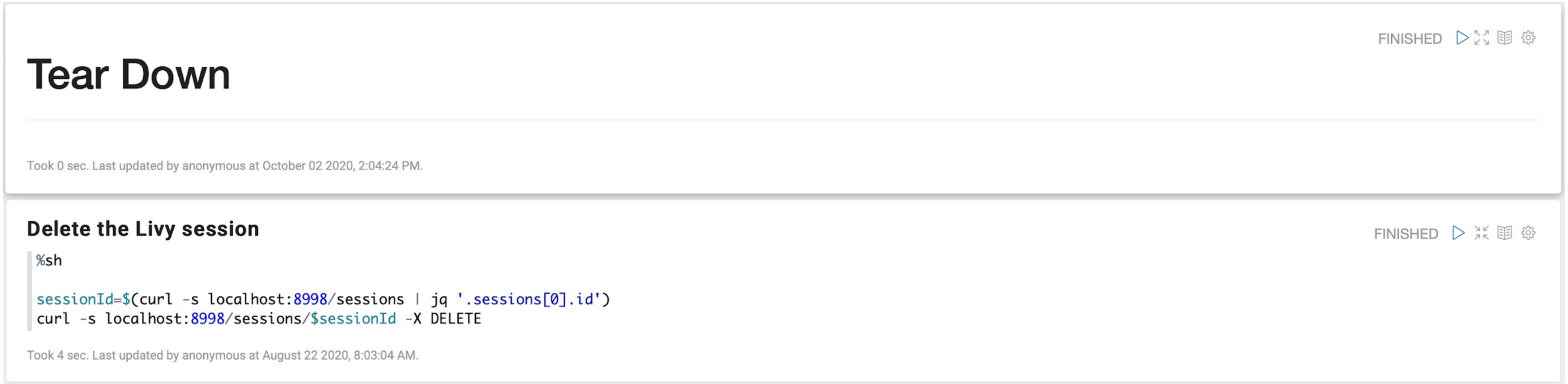
2 - Do a tail on orders.csv

```
%sh
hdfs dfs -tail  AdventureWorks/orders.csv
```

```
71952,113563,5/1/2014,5/13/2014,5/8/2014,275,1835,67059.6362,6573.0031,2054.0635,75686.7028,985,3,112.998,0.4,203.396471952,113563,5/1/2014,5/13/2014,5/8/2014,275,1835,67059.6
362,6573.0031,2054.0635,75686.7028,985,3,112.998,0.4,203.3964
```

Tear Down

- This section is specific to this new cluster that uses Livy as a broker between Zeppelin and Spark. It contains a single paragraph that deletes the Livy session that the notebook created at the beginning. This ensures that each notebook starts with a fresh Livy session thus avoiding accumulation phenomenons that eventually lead to random failures.
- The script relies on the jq framework that is installed on the cluster.



Tear Down

Took 0 sec. Last updated by anonymous at October 02 2020, 2:04:24 PM.

Delete the Livy session

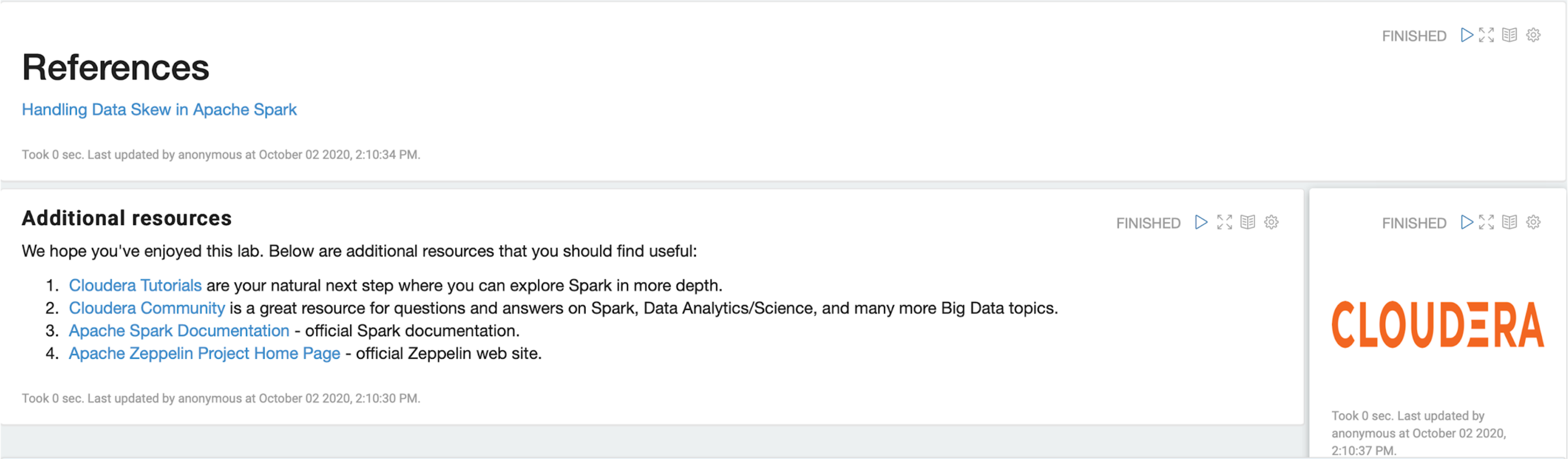
```
%sh

sessionId=$(curl -s localhost:8998/sessions | jq '.sessions[0].id')
curl -s localhost:8998/sessions/$sessionId -X DELETE
```

Took 4 sec. Last updated by anonymous at August 22 2020, 8:03:04 AM.

Footer

- This footer of the Solution section provides links to additional resources as well as the Cloudera Educational Services home page.



References

Handling Data Skew in Apache Spark

Took 0 sec. Last updated by anonymous at October 02 2020, 2:10:34 PM.

Additional resources

We hope you've enjoyed this lab. Below are additional resources that you should find useful:

1. Cloudera Tutorials are your natural next step where you can explore Spark in more depth.
2. Cloudera Community is a great resource for questions and answers on Spark, Data Analytics/Science, and many more Big Data topics.
3. Apache Spark Documentation - official Spark documentation.
4. Apache Zeppelin Project Home Page - official Zeppelin web site.

Took 0 sec. Last updated by anonymous at October 02 2020, 2:10:30 PM.

Took 0 sec. Last updated by anonymous at October 02 2020, 2:10:37 PM.

Adding the dev notebooks

To upload the notebooks in Zeppelin, the students need to run the command that is in the README file of their ~/training_materials/dev directory.

```
cd /home/training/training_materials/dev
sh scripts/uploadAllNotebooks.sh notebooks/dev-notebooks-20210830.zip
```

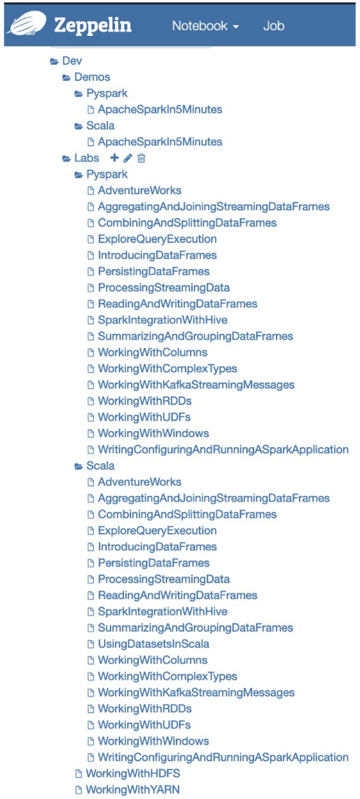
Adding the perf notebooks

If you want to experiment with the Perf notebooks, you can install them using this script:

```
cd /home/training/training_materials/perf/install/scripts
sh uploadAllNotebooks.sh localhost.localdomain:8885 'userName=admin&password=admin'
/home/training/training_materials/perf/notebooks/Perf-notebooks20210810.zip
```

Notebooks Folders Structure

- Dev
  - Labs
    - Pyspark
    - Scala
  - Demos
    - Pyspark
    - Scala
- Perf
  - Labs
    - Pyspark
  - Demos
    - Pyspark



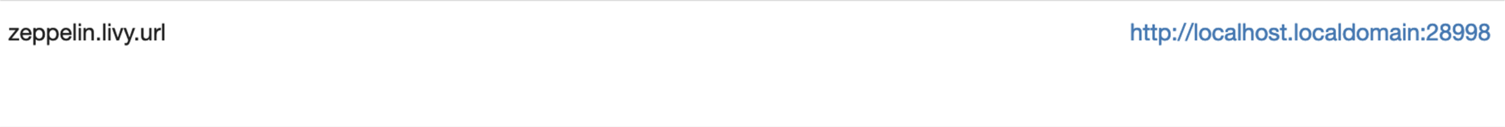
- Dev
  - Demos
    - Pyspark
      - ApacheSparkIn5Minutes
    - Scala
      - ApacheSparkIn5Minutes
  - Labs
    - Pyspark
      - AdventureWorks
      - AggregatingAndJoiningStreamingDataFrames
      - CombiningAndSplittingDataFrames
      - ExploreQueryExecution
      - IntroducingDataFrames
      - PersistingDataFrames
      - ProcessingStreamingData
      - ReadingAndWritingDataFrames
      - SparkIntegrationWithHive
      - SummarizingAndGroupingDataFrames
      - WorkingWithColumns
      - WorkingWithComplexTypes
      - WorkingWithKafkaStreamingMessages
      - WorkingWithRDDs
      - WorkingWithUDFs
      - WorkingWithWindows
      - WritingConfiguringAndRunningASparkApplication
    - Scala
      - AdventureWorks
      - AggregatingAndJoiningStreamingDataFrames
      - CombiningAndSplittingDataFrames
      - ExploreQueryExecution
      - IntroducingDataFrames
      - PersistingDataFrames
      - ProcessingStreamingData
      - ReadingAndWritingDataFrames
      - SparkIntegrationWithHive
      - SummarizingAndGroupingDataFrames
      - UsingDatasetsInScala
      - WorkingWithColumns
      - WorkingWithComplexTypes
      - WorkingWithKafkaStreamingMessages
      - WorkingWithRDDs
      - WorkingWithUDFs
      - WorkingWithWindows
      - WritingConfiguringAndRunningASparkApplication
    - WorkingWithHDFS
    - WorkingWithYARN

Notebooks

- There is no Exercise Manual, all the exercises are in notebooks
- Sometimes notebooks will contain only instructions and solutions in markdown cells
  - Spark Streaming
  - Submit a Spark Application
- Most of the times, the notebooks will run the code required to perform the tasks described in the instructions
- The notebooks come from several sources
  - Legacy DevSh
  - Data Science
  - HWX Spark course
  - Some I made specifically for this class
- **The notebooks coming from the Data Science class on day #2 include a Lesson part that instructors need to present**

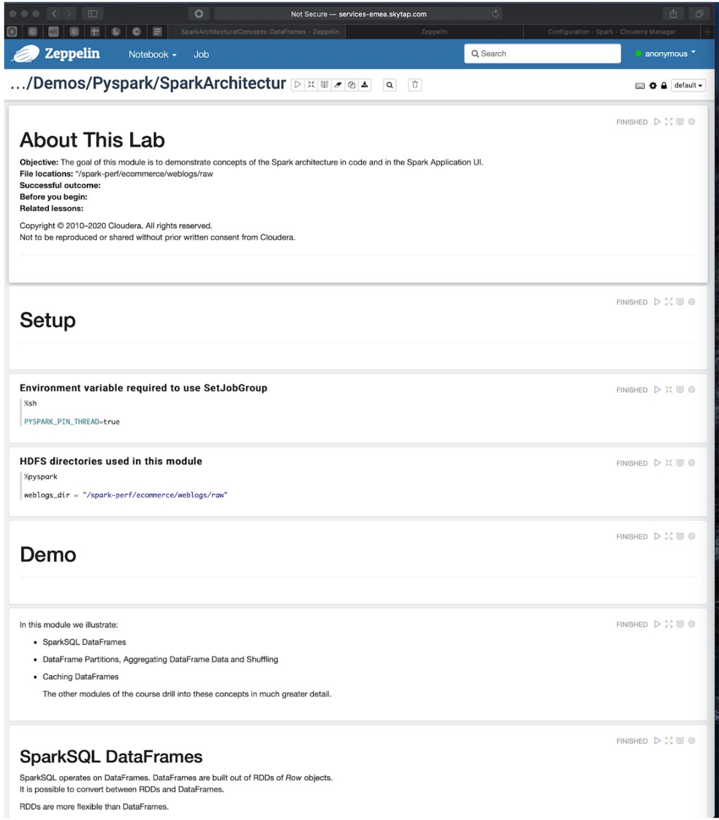
Working with Spark 3

- Spark3 has its own Application History Server listening on port 18089
- Zeppelin uses Livy for Spark3 to communicate with Spark 3
- Livy for Spark3 uses port 28998
- When Zeppelin restarts it defaults to the Livy for Spark2 port 8998
- Always check in the interpreter page that the zeppelin.livy.url is set to 28998



zeppelin.livy.url	http://localhost.localdomain:28998

Exhaustive use of setJobGroup



Not Secure — services-emea.skytap.com
SparkArchitecturalConcepts-DataFrames - Zeppelin
Zeppelin
Configuration - Spark - Cloudera Manager
Zeppelin
Notebook
Job
Search
anonymous
default

.../Demos/Pyspark/SparkArchitectur

FINISHED

About This Lab

Objective: The goal of this module is to demonstrate concepts of the Spark architecture in code and in the Spark Application UI.
File locations: "/spark-perf/ecommerce/weblogs/raw
Successful outcome:
Before you begin:
Related lessons:

Copyright © 2010–2020 Cloudera. All rights reserved.
Not to be reproduced or shared without prior written consent from Cloudera.

FINISHED

Setup

FINISHED

Environment variable required to use SetJobGroup

```
%sh

PYSPARK_PIN_THREAD=true
```

FINISHED

HDFS directories used in this module

```
%pyspark

weblogs_dir = "/spark-perf/ecommerce/weblogs/raw"
```

FINISHED

Demo

FINISHED

In this module we illustrate:

- SparkSQL DataFrames
- DataFrame Partitions, Aggregating DataFrame Data and Shuffling
- Caching DataFrames

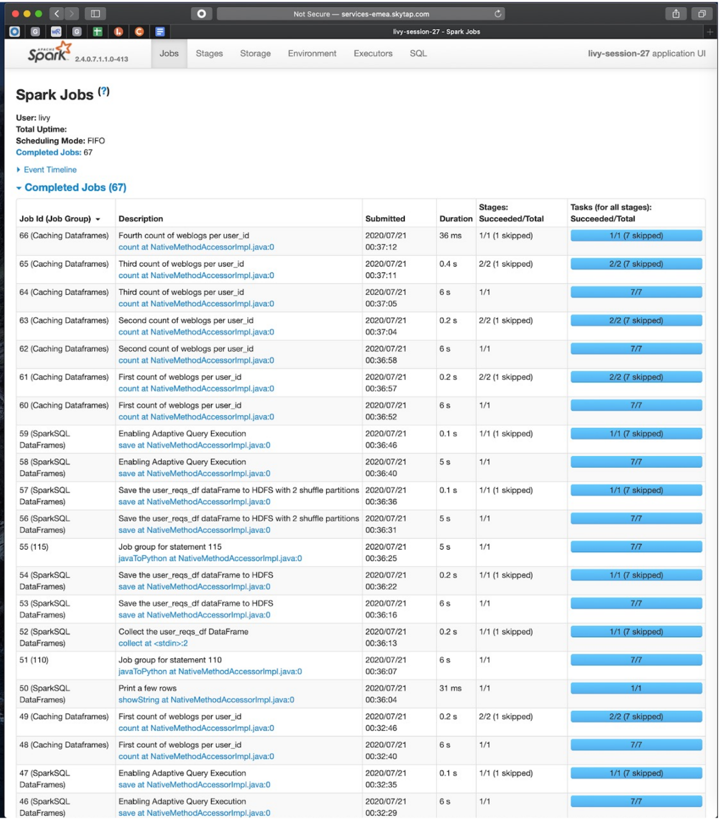
The other modules of the course drill into these concepts in much greater detail.

FINISHED

SparkSQL DataFrames

SparkSQL operates on DataFrames. DataFrames are built out of RDDs of *Row* objects.
It is possible to convert between RDDs and DataFrames.

RDDs are more flexible than DataFrames.



Not Secure — services-emea.skytap.com
livy-session-27 - Spark Jobs
Spark 2.4.0.7.1.1.0-413
Jobs
Stages
Storage
Environment
Executors
SQL
livy-session-27 application UI

Spark Jobs (?)

User: livy
Total Uptime:
Scheduling Mode: FIFO
Completed Jobs: 67

▸ Event Timeline

▾ **Completed Jobs (67)**

Job Id (Job Group) ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
66 (Caching Dataframes)	Fourth count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:37:12	36 ms	1/1 (1 skipped)	1/1 (7 skipped)
65 (Caching Dataframes)	Third count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:37:11	0.4 s	2/2 (1 skipped)	2/2 (7 skipped)
64 (Caching Dataframes)	Third count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:37:05	6 s	1/1	7/7
63 (Caching Dataframes)	Second count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:37:04	0.2 s	2/2 (1 skipped)	2/2 (7 skipped)
62 (Caching Dataframes)	Second count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:58	6 s	1/1	7/7
61 (Caching Dataframes)	First count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:57	0.2 s	2/2 (1 skipped)	2/2 (7 skipped)
60 (Caching Dataframes)	First count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:52	6 s	1/1	7/7
59 (SparkSQL DataFrames)	Enabling Adaptive Query Execution save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:46	0.1 s	1/1 (1 skipped)	1/1 (7 skipped)
58 (SparkSQL DataFrames)	Enabling Adaptive Query Execution save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:40	5 s	1/1	7/7
57 (SparkSQL DataFrames)	Save the user_reqs_df dataFrame to HDFS with 2 shuffle partitions save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:36	0.1 s	1/1 (1 skipped)	1/1 (7 skipped)
56 (SparkSQL DataFrames)	Save the user_reqs_df dataFrame to HDFS with 2 shuffle partitions save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:31	5 s	1/1	7/7
55 (115)	Job group for statement 115 javaToPython at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:25	5 s	1/1	7/7
54 (SparkSQL DataFrames)	Save the user_reqs_df dataFrame to HDFS save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:22	0.2 s	1/1 (1 skipped)	1/1 (7 skipped)
53 (SparkSQL DataFrames)	Save the user_reqs_df dataFrame to HDFS save at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:16	6 s	1/1	7/7
52 (SparkSQL DataFrames)	Collect the user_reqs_df DataFrame collect at <stdin>:2	2020/07/21 00:36:13	0.2 s	1/1 (1 skipped)	1/1 (7 skipped)
51 (110)	Job group for statement 110 javaToPython at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:07	6 s	1/1	7/7
50 (SparkSQL DataFrames)	Print a few rows showString at NativeMethodAccessorImpl.java:0	2020/07/21 00:36:04	31 ms	1/1	1/1
49 (Caching Dataframes)	First count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:32:46	0.2 s	2/2 (1 skipped)	2/2 (7 skipped)
48 (Caching Dataframes)	First count of weblogs per user_id count at NativeMethodAccessorImpl.java:0	2020/07/21 00:32:40	6 s	1/1	7/7
47 (SparkSQL DataFrames)	Enabling Adaptive Query Execution save at NativeMethodAccessorImpl.java:0	2020/07/21 00:32:35	0.1 s	1/1 (1 skipped)	1/1 (7 skipped)
46 (SparkSQL DataFrames)	Enabling Adaptive Query Execution save at NativeMethodAccessorImpl.java:0	2020/07/21 00:32:29	6 s	1/1	7/7

The Paragraphs

- There are no empty paragraphs in the notebooks
- Just paragraphs waiting to be executed

CLOUDERA
Educational Services

The content

Course components

Component	Version Control	Version
Students Presentation	Git	https://github.com/HortonworksUniversity/contentFactory
Zeppelin notebooks	Git	https://github.com/HortonworksUniversity/contentFactory
Skytap template	Template name includes date	
Instructor Guide	File name includes date	
Data Sheet	File name includes date	

Agenda

Day 1	Day 2	Day 3	Day 4
Class Introduction	Spark DataFrames	Integration with Hive	Introduction to Streaming DataFrames
Zeppelin Introduction	Reading DataFrames	Visualization with Zeppelin	Kafka Introduction
HDFS Introduction	Working with Columns	Distributed Processing	Integration with Kafka
YARN Introduction	Transforming DataFrames	Distributed Persistence	Aggregating and Joining Streaming DataFrames
Distributed Processing History	Working with UDFs	Building Spark Applications	(*) Appendix: Scala Datasets
Spark RDDs	Working with Windows		

(*) if time allows Comes from the HWX Spark class Comes
from the Data Science class

Design Principles

- #Death2RDDs
  - Only one chapter and one notebook on RDDs
  - RDD appears in 71 slides vs 160 in the previous version
- #Death2Slides
  - 17 Pyspark notebooks
  - 19 Scala notebooks
  - 2 Shell notebooks
- #Death2Monoliths
  - The content is made of 59 components listed in a manifest file
  - Those components are managed in a github repository:

    https://github.com/HortonworksUniversity/contentFactory
- #VivaKnowledgeChecks
  - I adapted some of the Knowledge Checks we had at HWX and created one

The Manifest (1 / 4)

```
Lesson|DEV/Spark/ClouderaDeveloperTrainingClassIntroduction
Lesson|DEV/Spark/ZeppelinIntroduction
Demo/Notebook|Dev_Demos_Pyspark_ApacheSparkIn5Minutes.json
Demo/Notebook|Dev_Demos_Scala_ApacheSparkIn5Minutes.json
Lesson|DEV/Spark/HDFSIntroduction
Lab/Notebook|Dev_Labs_WorkingWithHDFS.json
Lesson|DEV/Spark/YARNIntroduction
Lab/Notebook|Dev_Labs_WorkingWithYARN.json
Lesson|DEV/Spark/DistributedProcessingHistory
Lesson|DEV/Spark/WorkingWithRDDs
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithRDDs.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithRDDs.json
Lesson|DEV/Spark/WorkingWithDataFrames
Lab/Notebook|Dev_Labs_Pyspark_IntroducingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_IntroducingDataFrames.json
Lab/Notebook|Dev_Labs_Pyspark_ReadingAndWritingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_ReadingAndWritingDataFrames.json
```

The Manifest (2 / 4)

```
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithColumns.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithColumns.json
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithComplexTypes.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithComplexTypes.json
Lab/Notebook|Dev_Labs_Pyspark_CombiningAndSplittingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_CombiningAndSplittingDataFrames.json
Lab/Notebook|Dev_Labs_Pyspark_SummarizingAndGroupingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_SummarizingAndGroupingDataFrames.json
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithUDFs.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithUDFs.json
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithWindows.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithWindows.json
Lesson|DEV/Spark/ApacheHiveIntroduction
Lesson|DEV/Spark/HiveSparkIntegration
Lab/Notebook|Dev_Labs_Pyspark_SparkIntegrationWithHive.json
Lab/Notebook|Dev_Labs_Scala_SparkIntegrationWithHive.json
Lesson|DEV/Spark/DataVisualizationWithZeppelin
```

The Manifest (3 / 4)

```
Lab/Notebook|Dev_Labs_Pyspark_AdventureWorks.json
Lab/Notebook|Dev_Labs_Scala_AdventureWorks.json
Lesson|DEV/Spark/DistributedProcessingChallenges
Lesson|DEV/Spark/SparkDistributedProcessing
Lab/Notebook|Dev_Labs_Pyspark_ExploreQueryExecution.json
Lab/Notebook|Dev_Labs_Scala_ExploreQueryExecution.json
Lesson|DEV/Spark/SparkDistributedPersistence
Lab/Notebook|Dev_Labs_Pyspark_PersistingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_PersistingDataFrames.json
Lesson|DEV/Spark/WritingConfiguringAndRunningSparkApplications
Lab/Notebook|Dev_Labs_Pyspark_WritingConfiguringAndRunningASparkApplication.json
Lab/Notebook|Dev_Labs_Scala_WritingConfiguringAndRunningASparkApplication.json
Lesson|DEV/Spark/IntroductionToStructuredStreaming
Lab/Notebook|Dev_Labs_Pyspark_ProcessingStreamingData.json
Lab/Notebook|Dev_Labs_Scala_ProcessingStreamingData.json
Lesson|DEV/Spark/MessageProcessingWithApacheKafka
Lesson|DEV/Spark/StructuredStreamingWithApacheKafka
```

The Manifest (4 / 4)

```
Lab/Notebook|Dev_Labs_Pyspark_WorkingWithKafkaStreamingMessages.json
Lab/Notebook|Dev_Labs_Scala_WorkingWithKafkaStreamingMessages.json
Lesson|DEV/Spark/AggregatingAndJoiningStreamingDataFrames
Lab/Notebook|Dev_Labs_Pyspark_AggregatingAndJoiningStreamingDataFrames.json
Lab/Notebook|Dev_Labs_Scala_AggregatingAndJoiningStreamingDataFrames.json
Lesson|DEV/Spark/ClouderaDeveloperTrainingClassConclusion
Lesson|DEV/Spark/WorkingWithDatasetsInScala
Lab/Notebook|Dev_Labs_Scala_UsingDatasetsInScala.json
```

The orange notebooks include a lesson section that should be presented by instructors.

Optional content

- Working with Datasets in Scala
  - Scala specific

About the Knowledge Checks

- Use them to engage with students and quickly check their understanding
- But don't get sidetracked into protracted conversations
- You don't have to ask all of them, use the ones
  - That you like or
  - That will validate an important point
- Some questions will be about topics that were not introduced in the slides, you can
  - Choose to skip them or
  - Use them to introduce the corresponding topic

CLOUDERA
Educational Services

Walkthrough of the ‘New’ Content

- Zeppelin Introduction
- Distributed Processing History
- Integration with Hive
- Distributed Processing Challenges
- Visualization with Zeppelin

CLOUDERA
Educational Services

Timings

Day #1

#	Item	Day	Duration
1	Lesson|DEV/Spark/ClouderaDeveloperTrainingClassIntroduction	1	45
2	Lesson|DEV/Spark/ZeppelinIntroduction	1	20
3	Demo/Notebook|Dev_Demos_Pyspark_ApacheSparkIn5Minutes.json	1	20
5	Lesson|DEV/Spark/HDFSIntroduction	1	45
6	Lab/Notebook|Dev_Labs_WorkingWithHDFS.json	1	30
7	Lesson|DEV/Spark/YARNIntroduction	1	45
8	Lab/Notebook|Dev_Labs_WorkingWithYARN.json	1	30
9	Lesson|DEV/Spark/DistributedProcessingHistory	1	30
10	Lesson|DEV/Spark/WorkingWithRDDs	1	45
11	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithRDDs.json	1	40

Day #2

#	Item	Day	Duration
13	Lesson|DEV/Spark/WorkingWithDataFrames	2	45
14	Lab/Notebook|Dev_Labs_Pyspark_IntroducingDataFrames.json	2	50
16	Lab/Notebook|Dev_Labs_Pyspark_ReadingAndWritingDataFrames.json	2	40
18	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithColumns.json	2	50
20	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithComplexTypes.json	2	30
22	Lab/Notebook|Dev_Labs_Pyspark_CombiningAndSplittingDataFrames.json	2	40
24	Lab/Notebook|Dev_Labs_Pyspark_SummarizingAndGroupingDataFrames.json	2	40
26	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithUDFs.json	2	30
28	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithWindows.json	2	30

Day #3

#	Item	Day	Duration
30	Lesson|DEV/Spark/ApacheHiveIntroduction	3	20
31	Lesson|DEV/Spark/HiveSparkIntegration	3	20
32	Lab/Notebook|Dev_Labs_Pyspark_SparkIntegrationWithHive.json	3	40
34	Lesson|DEV/Spark/DataVisualizationWithZeppelin	3	20
35	Lab/Notebook|Dev_Labs_Pyspark_AdventureWorks.json	3	40
37	Lesson|DEV/Spark/DistributedProcessingChallenges	3	20
38	Lesson|DEV/Spark/SparkDistributedProcessing	3	30
39	Lab/Notebook|Dev_Labs_Pyspark_ExploreQueryExecution.json	3	40
41	Lesson|DEV/Spark/SparkDistributedPersistence	3	30
42	Lab/Notebook|Dev_Labs_Pyspark_PersistingDataFrames.json	3	40
44	Lesson|DEV/Spark/WritingConfiguringAndRunningSparkApplications	3	30
45	Lab/Notebook|Dev_Labs_Pyspark_WritingConfiguringAndRunningASparkApplication.json	3	40

Day #4

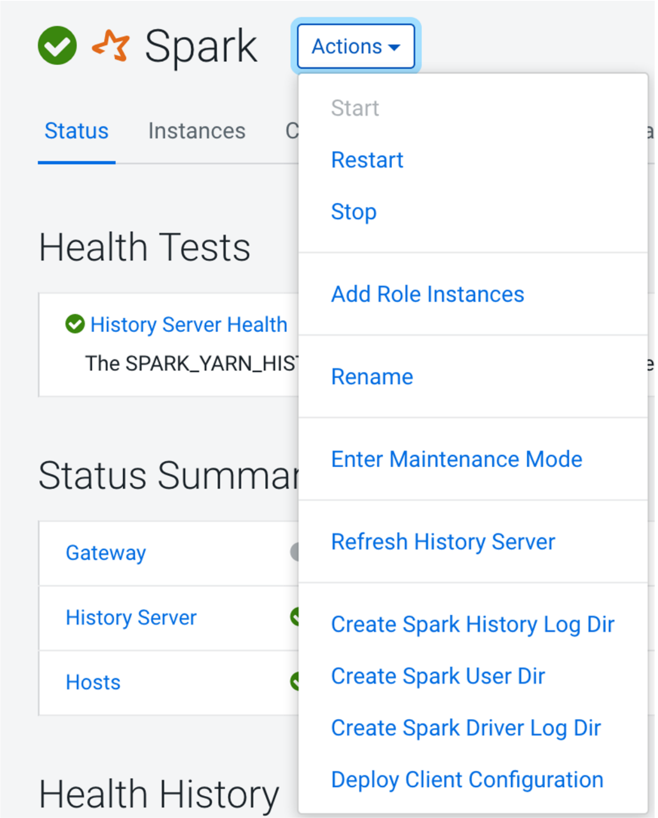
#	Item	Day	Duration
47	Lesson|DEV/Spark/IntroductionToStructuredStreaming	4	40
48	Lab/Notebook|Dev_Labs_Pyspark_ProcessingStreamingData.json	4	60
50	Lesson|DEV/Spark/MessageProcessingWithApacheKafka	4	40
51	Lesson|DEV/Spark/StructuredStreamingWithApacheKafka	4	30
52	Lab/Notebook|Dev_Labs_Pyspark_WorkingWithKafkaStreamingMessages.json	4	60
54	Lesson|DEV/Spark/AggregatingAndJoiningStreamingDataFrames	4	30
55	Lab/Notebook|Dev_Labs_Pyspark_AggregatingAndJoiningStreamingDataFrames.json	4	60
57	Lesson|DEV/Spark/ClouderaDeveloperTrainingClassConclusion	4	10
58	Lesson|DEV/Spark/WorkingWithDatasetsInScala	4	30
59	Lab/Notebook|Dev_Labs_Scala_UsingDatasetsInScala.json	4	30

CLOUDERA
Educational Services

Troubleshooting

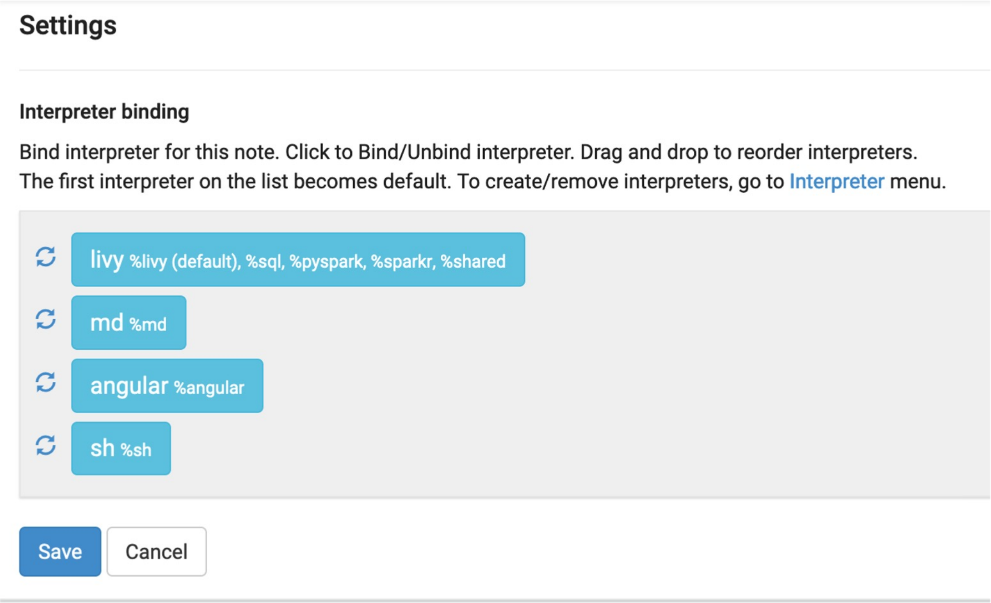
The Spark History Server becomes weak in the knees

- Just click on the 'Refresh History Server' in the Spark Actions menu:



Your Livy session is dead or smells funny

- This is unlikely to occur because I delete the Livy session at the bottom of each notebook so that you get a new one with each notebook but if it does, open the interpreter binding of the notebook and click on the blue loop icon next to the livy button.



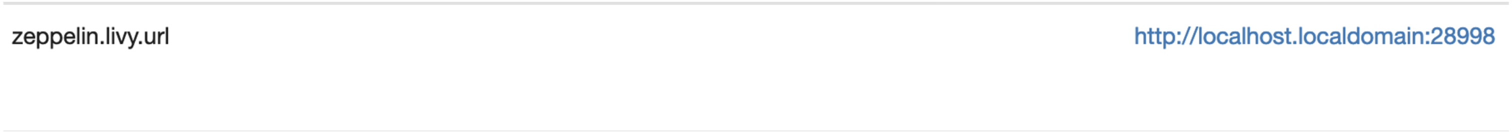
Settings

Interpreter binding

Bind interpreter for this note. Click to Bind/Unbind interpreter. Drag and drop to reorder interpreters.
The first interpreter on the list becomes default. To create/remove interpreters, go to Interpreter menu.

You get strange error messages about properties being unknown

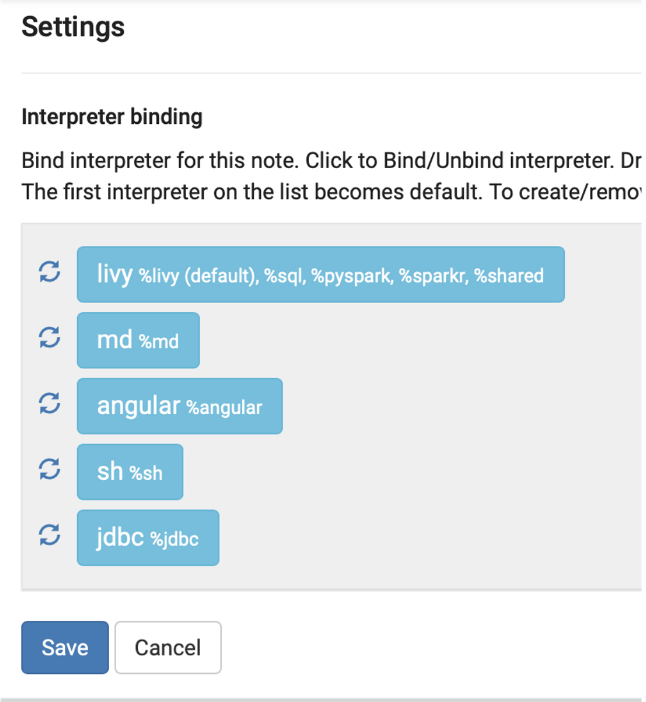
- You restarted Zeppelin
- It defaulted to the Livy for Spark2 port
- You need to change the url of zeppelin.livy.url in the livy interpreter settings back to 28998



zeppelin.livy.url http://localhost.localdomain:28998

Your first %jdbc cell reports an error

- If your first %jdbc cell reports an error just restart the jdbc interpreter
  - Open the interpreter binding panel by clicking on the small cog icon at the top right of the screen
  - Click on the looping arrows next to the jdbc blue button
  - Click Save



Settings

Interpreter binding

Bind interpreter for this note. Click to Bind/Unbind interpreter. Dr
The first interpreter on the list becomes default. To create/remo

CLOUDERA
Educational Services

Knowledge Checks Cheat sheet

HDFS - Answers

1 - Blocks, replicas/copies, reliability/robustness, data locality
2 - NameNode
3 - DataNode
4 - True. This allows the NameNode to be so highly-available
5 - False. Clients write to the first DataNode in the list created by the NameNode and the DN's then "pipeline" the data writing to the additional DN's

YARN - Answers

1 – master = ResourceManager and worker = NodeManager
2 - Memory
3 – False, they run on worker nodes and control a specific job
4 – The ApplicationMaster must decide what to do which is not always to spin up a replacement
5 – False, queues are given a certain percentage of the whole. Bonus answer is that “node labels” could be assigned to particular worker nodes which then can be configured for a given queue, but we didn’t discuss this in the preso.

Distributed Processing History - Answers

1 – Map and Reduce
2 – The number of blocks the input data is persisted to on HDFS. User can supply the number of reducers.
3 – Mappers are called with a single KVP and can return 0..m. Reducers receive a single KVP (the value is a list of values) and can also return 0..m.
4 – False, but you can have a Map-only job which would be the quickest possible job due to the lack of shuffle/sort and Reducer phases.
5 – Because they are “easier” for most developers and analysts
6 – False, Spark took many concepts from MapReduce and implemented them in a new way.
7 – False, no code change is required the GPU library mirrors the existing CPU one

Working with RDDs - Answers

1. Resilient Distributed Dataset
2. sc.parallelize() and sc.textfile()
3. False. Transformations result in new RDDs being created. In Spark, data is immutable.
4. flatmap()
5. True
6. Lazy evaluation
7. False. The intersection function performs this task. The distinct function would remove duplicates elements, so that each element is only listed once regardless of how many times it appeared in the original RDD.
8. True

Working with DataFrames - Answers

1) An API that allows to SQL to generate Spark jobs
2) Using spark.read, available formats are csv, json,parquet, orc, text
3) A DataFrame is a Dataset of type row, Datasets are strongly typed objects
4) False

Hive Introduction - Answers

1 – HiveServer2 (HS2)
2 – MR, Spark & Tez.
3 – When a managed table is dropped, all of its underlying files will be deleted.
5 – Tez, ORC, Vectorization, ACID transactions

Data Visualization with Zeppelin - Answers

1. Enable humans to make inferences and draw conclusions about large sets of data that would be impossible to make by looking at the data in tabular format.
2. Five
3. Export to JSON format, then they can import it.
4. Give them the note URL.
5. The Report view.
6. Link the paragraph
7. Dynamic forms

Spark Distributed Processing - Answers

1 - False
2 - False: an Application is a sequence of jobs
3 - Catalyst and Tungsten
4 - False, Lambdas impede Catalyst
5 - False
6 - False
7 - False, they are equivalent

CLOUDERA
Educational Services

What’s next?

For You

- Read through the presentation
  - If you have comments/questions use the Google Slides internal commenting feature
  - I am the only one to be able to edit the presentation
- Launch an instance of the Skytap template
- Run through the notebooks

Ideas for future evolution

- Consider storing the datasets in S3 for additional modularity
- Consider adding Flink modules to offer an alternative to the Spark Streaming content

CLOUDERA
Educational Services

THANK YOU