CLOUDERA
Educational Services

Spark Application Performance Tuning with CDP
Train the Trainer

October 2020, 26
François Reynald

Agenda

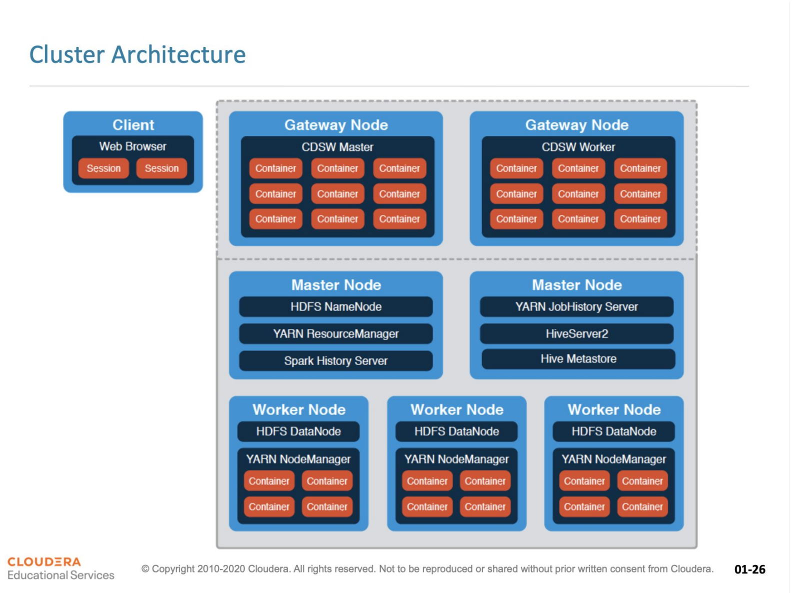
- Why this new version?
- New environment
- The notebooks
- New content
  - Catalyst and Tungsten Overview
  - WXM Introduction
  - What’s New in Spark 3.0?
- Troubleshooting
- What’s next?
- Questions

CLOUDERA
Educational Services

Why this new version?

Cluster Architecture

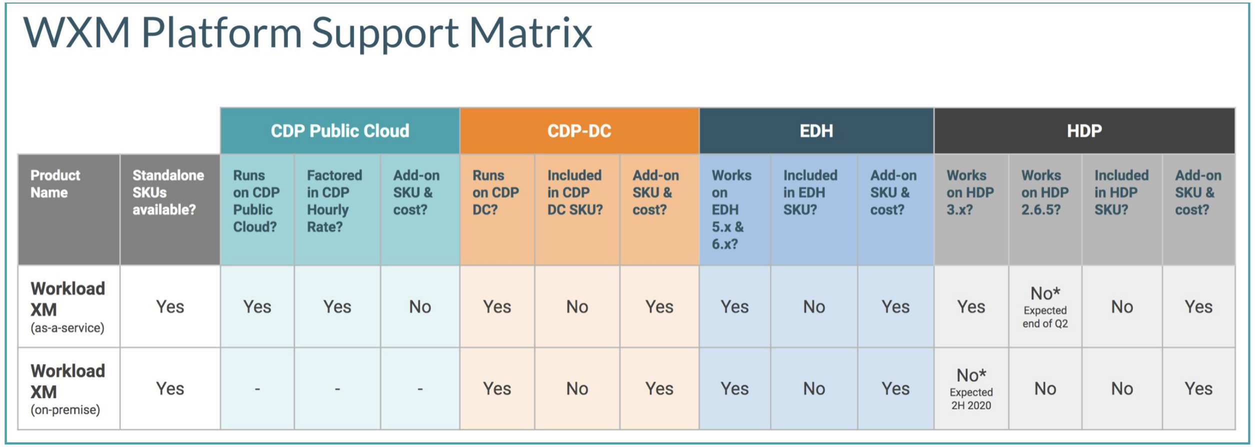
These clusters are expensive and their manual provisioning does not currently scale well across regions



No Commonality with DevSH

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

No WXM



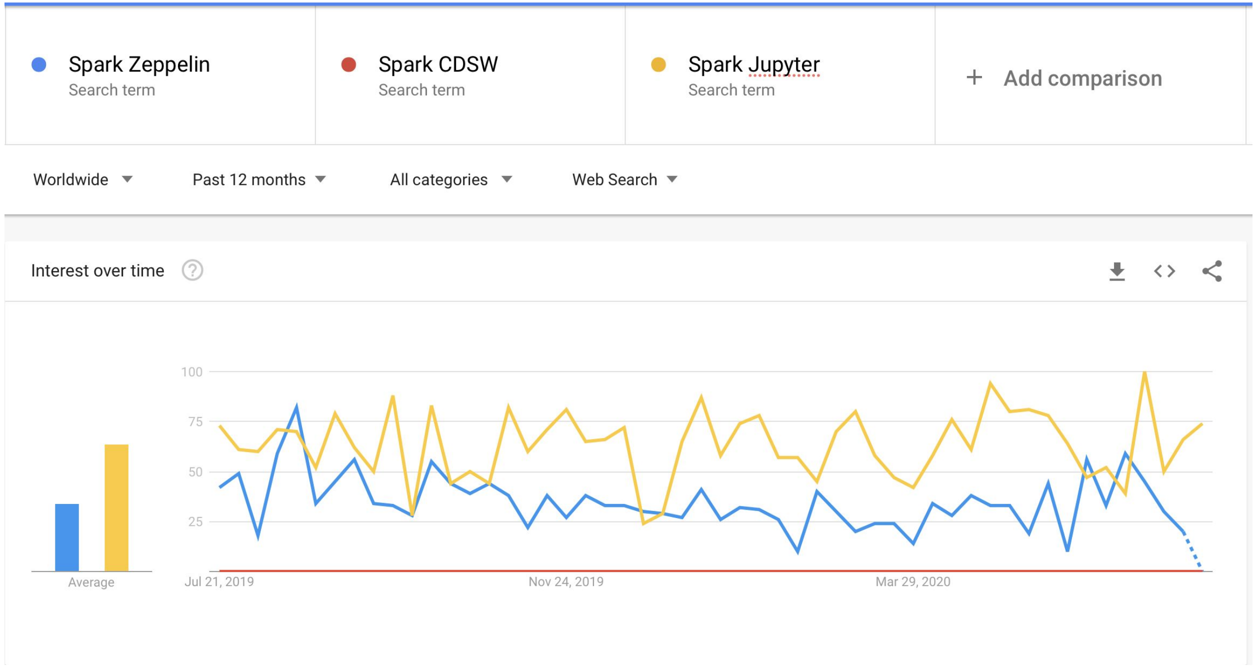
WXM Platform Support Matrix

		CDP Public Cloud			CDP-DC			EDH			HDP			
Product Name	**Standalone SKUs available?**	**Runs on CDP Public Cloud?**	**Factored in CDP Hourly Rate?**	**Add-on SKU & cost?**	**Runs on CDP DC?**	**Included in CDP DC SKU?**	**Add-on SKU & cost?**	**Works on EDH 5.x & 6.x?**	**Included in EDH SKU?**	**Add-on SKU & cost?**	**Works on HDP 3.x?**	**Works on HDP 2.6.5?**	**Included in HDP SKU?**	**Add-on SKU & cost?**
Workload XM (as-a-service)	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes	No* Expected end of Q2	No	Yes
Workload XM (on-premise)	Yes	-	-	-	Yes	No	Yes	Yes	No	Yes	No* Expected 2H 2020	No	No	Yes

No Spark 3.0

- Spark 3.0 is only available in CDP and I sincerely doubt it will be made available for CDH 5.15
- Spark 3.0 contains major performance related improvements that promise to perform 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- If we do not jump on this train fast our content will be quickly outdated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

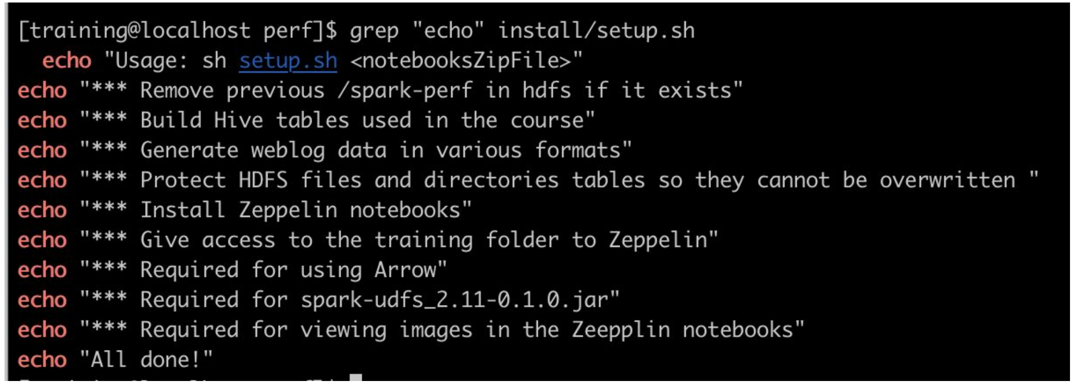


CLOUDERA
Educational Services

New environment

The Environment

- Started from the latest DevSH template
- Created a /home/training/training-materials/perf home folder
- Updated to CDP PVC BE 7.1.3,
- installed Spark 3.0.1,
- Boosted the drive space to 150 GB to improve stability
- 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

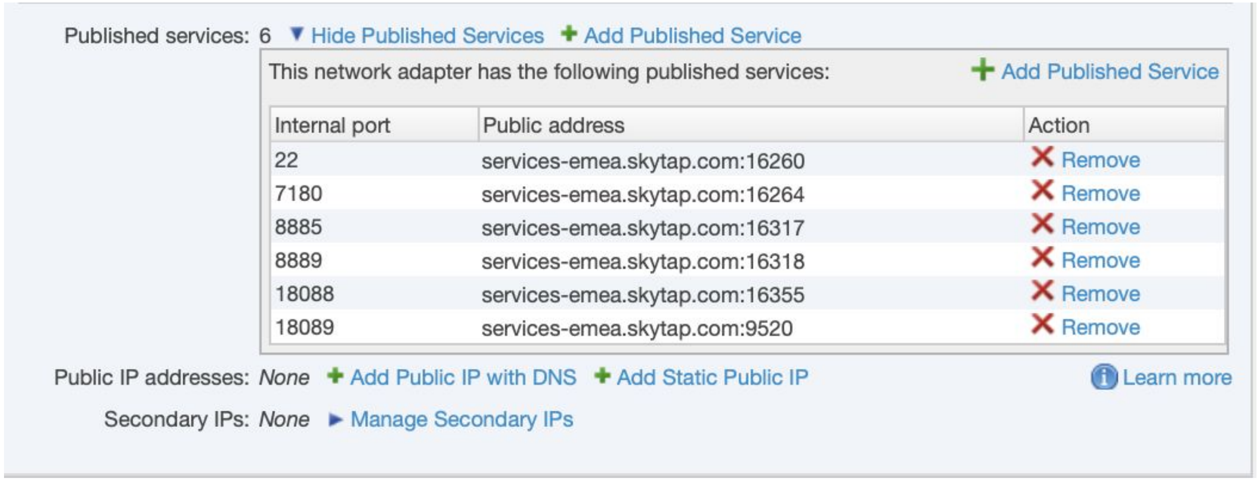


```
[training@localhost perf]$ grep "echo" install/setup.sh
  echo "Usage: sh setup.sh <notebooksZipFile>"
echo "*** Remove previous /spark-perf in hdfs if it exists"
echo "*** Build Hive tables used in the course"
echo "*** Generate weblog data in various formats"
echo "*** Protect HDFS files and directories tables so they cannot be overwritten "
echo "*** Install Zeppelin notebooks"
echo "*** Give access to the training folder to Zeppelin"
echo "*** Required for using Arrow"
echo "*** Required for spark-udfs_2.11-0.1.0.jar"
echo "*** Required for viewing images in the Zeepplin notebooks"
echo "All done!"
```

CLOUDERA Educational Services

Published Services

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



Published services: 6 ▼ Hide Published Services ✚ Add Published Service

This network adapter has the following published services: ✚ Add Published Service

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* ✚ Add Public IP with DNS ✚ Add Static Public IP ⓘ Learn more

Secondary IPs: *None* ▶ Manage Secondary IPs

/home/training/training_materials/perf

- data
  - The datasets used in the legacy labs
- images
  - The images that are displayed in the notebooks
- install
  - The main installation script setup.sh which calls utility scripts that are in the scripts subfolder
- jars
  - Jar file required for the UDFs
- notebooks
  - Zip file of the current version of notebooks
- tpcds-kit
  - Home folder of the TPCDS kit used to generate TPCDS data

The entire perf folder is zipped into a single master file: `zip -r perf-20201002.zip perf/*`

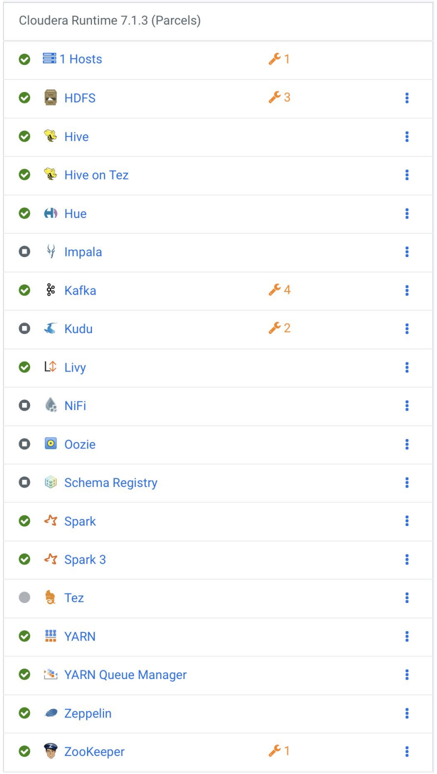
/home/training/training_materials/devsh

- notebooks
  - Added the DevSH notebooks in a notebooks folder

Available Services

Only differences are

- Spark 3 and
- Yarn Queue Manager
  - I had to install it to update 7.1.3
  - We are not using it



Cloudera Runtime 7.1.3 (Parcels)

✓	1 Hosts	1
✓	HDFS	3
✓	Hive	
✓	Hive on Tez	
✓	Hue	
○	Impala	
✓	Kafka	4
○	Kudu	2
✓	Livy	
○	NiFi	
○	Oozie	
○	Schema Registry	
✓	Spark	
✓	Spark 3	
●	Tez	
✓	YARN	
✓	YARN Queue Manager	
✓	Zeppelin	
✓	ZooKeeper	1

CLOUDERA Educational Services

Resource Management

- You cannot run two Spark sessions in parallel so you need to stop the Zeppelin service to start a pyspark3 shell
- As a good practice, stop the services you do not need

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
  - Lab *
  - Result
  - Solution
  - Tear down
  - Footer

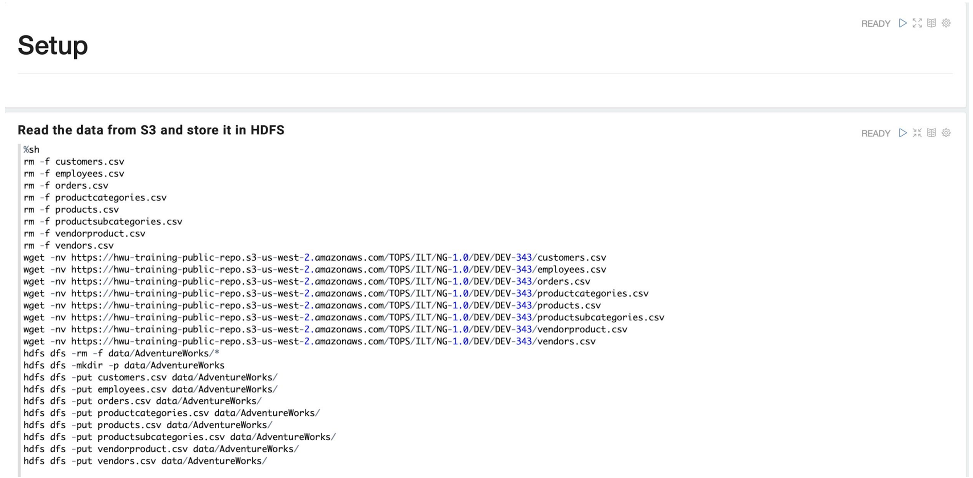
* This part is optional for Demo only notebooks

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

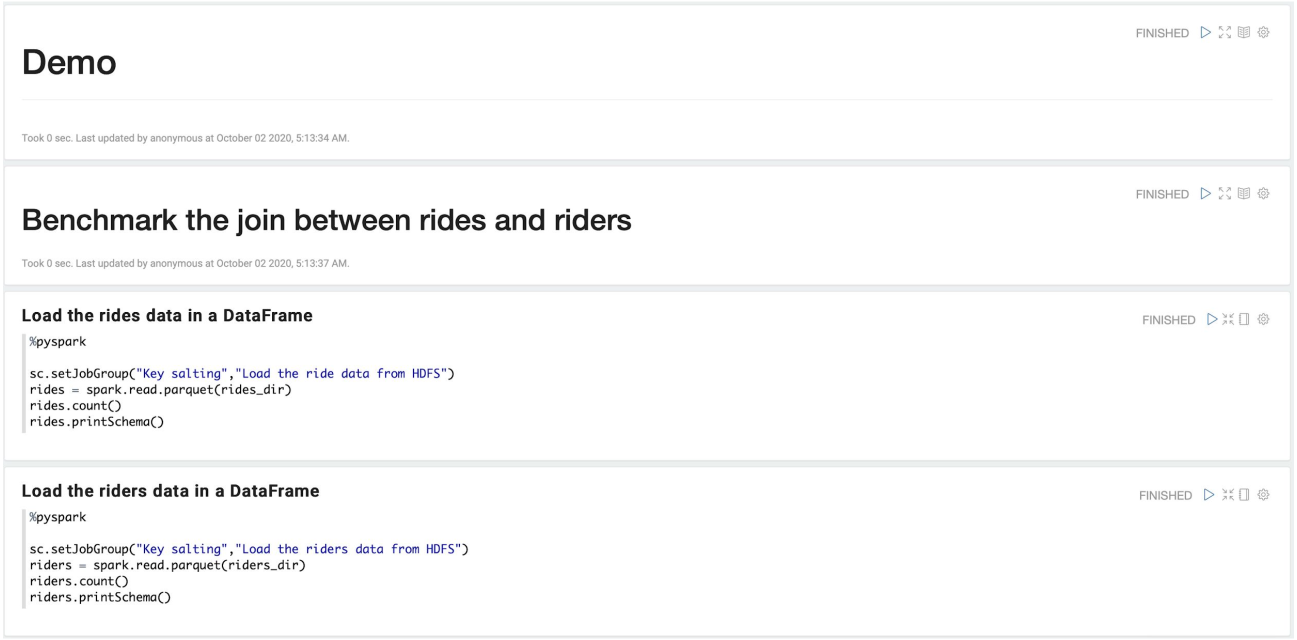
Read the data from S3 and store it in HDFS

```
%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/
```

CLOUDERA
Educational Services

Demo

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



Demo

FINISHED

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

Benchmark the join between rides and riders

FINISHED

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

Load the rides data in a DataFrame

FINISHED

```
%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

FINISHED

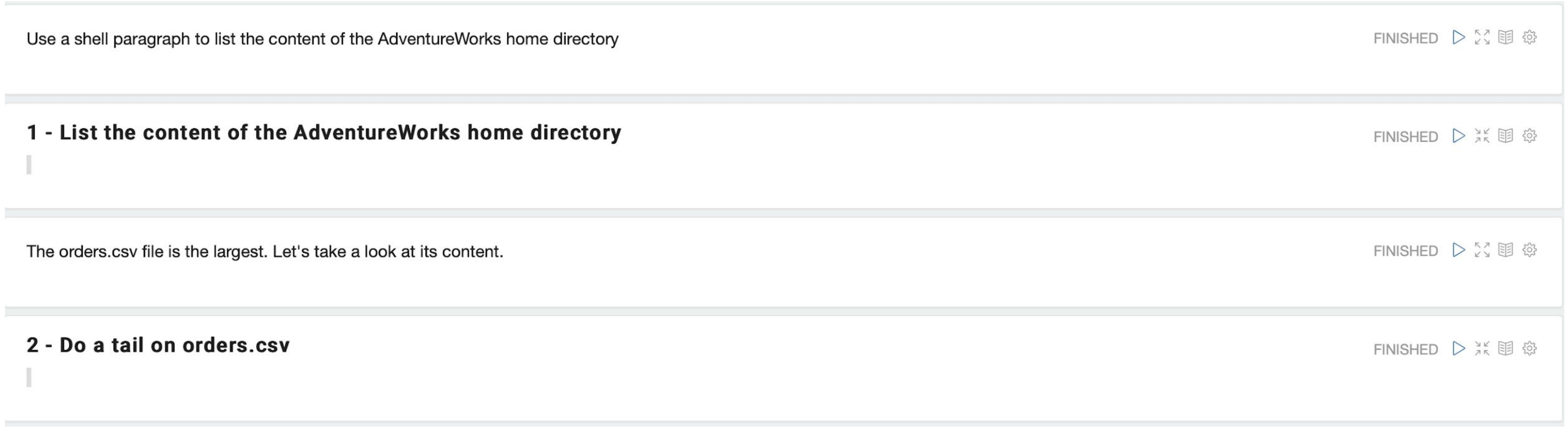
```
%pyspark

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

CLOUDERA
Educational Services

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

FINISHED

1 - List the content of the AdventureWorks home directory

FINISHED

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

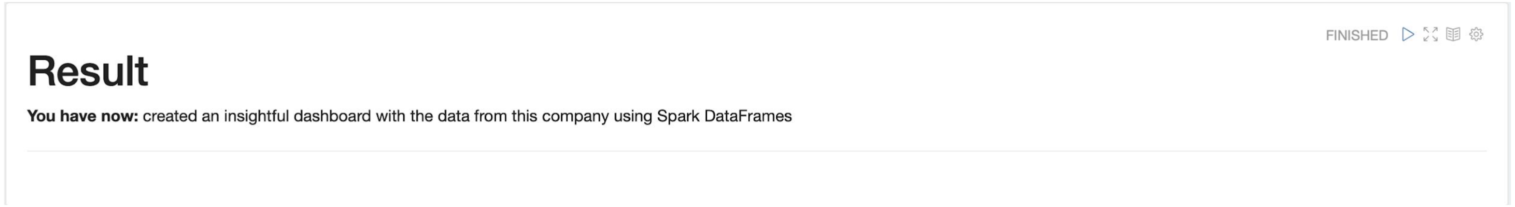
FINISHED

2 - Do a tail on orders.csv

FINISHED

Result

- This section summarizes what the student has just achieved.

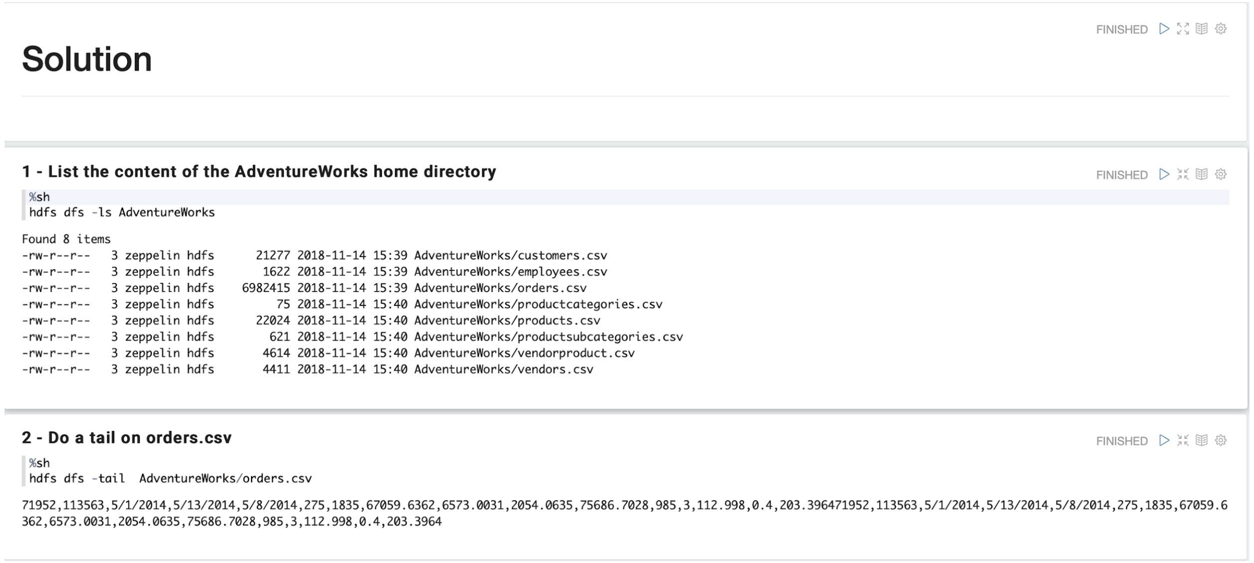

FINISHED

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

FINISHED

1 - List the content of the AdventureWorks home directory

FINISHED

```
%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

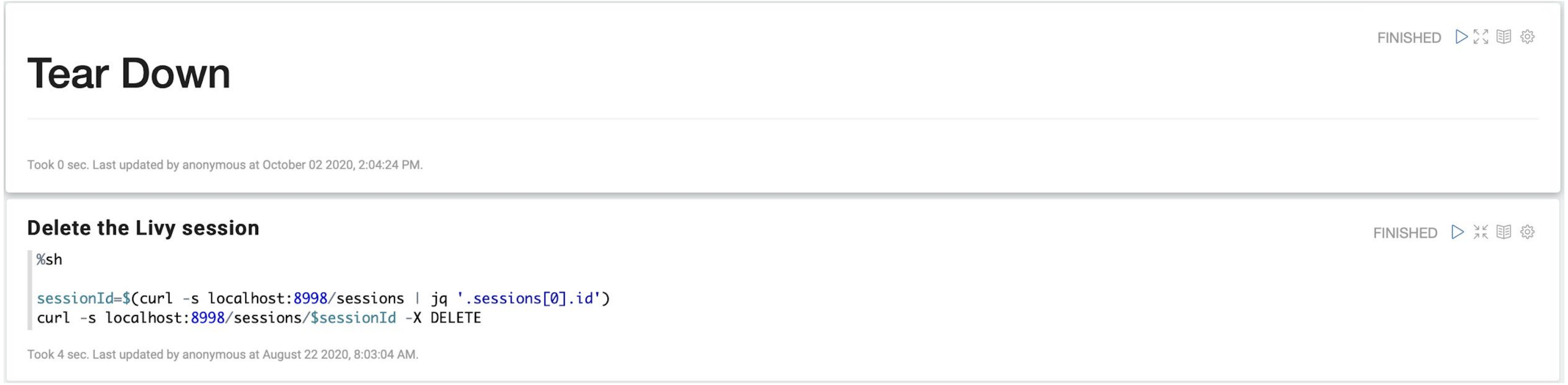
FINISHED

```
%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

FINISHED

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

Delete the Livy session

FINISHED

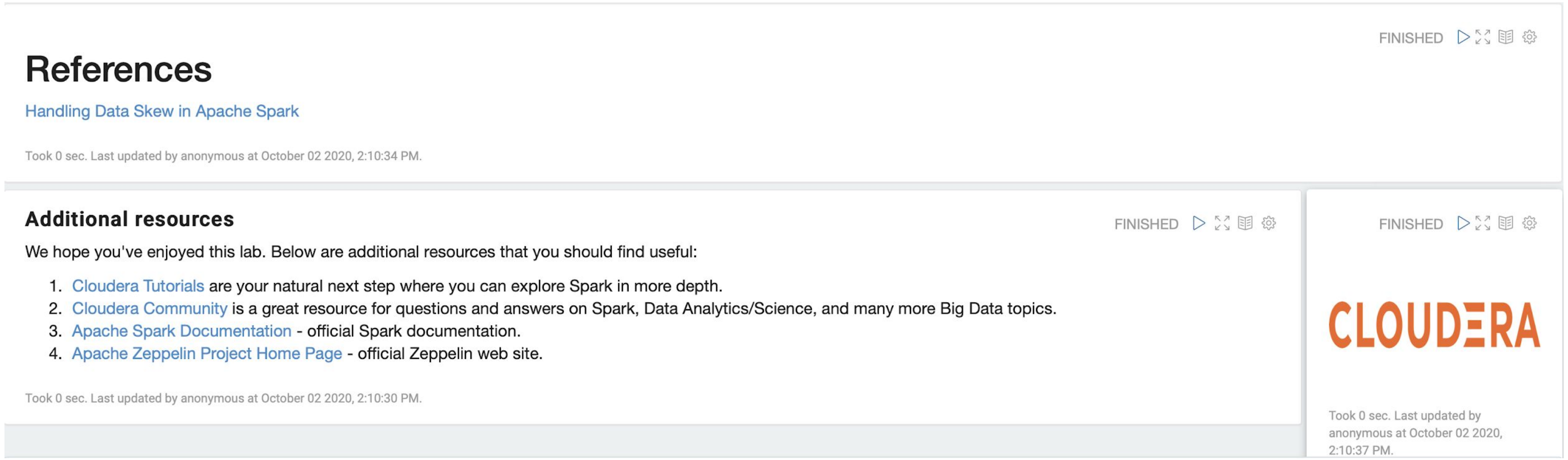
```
%sh

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 DevSH notebooks

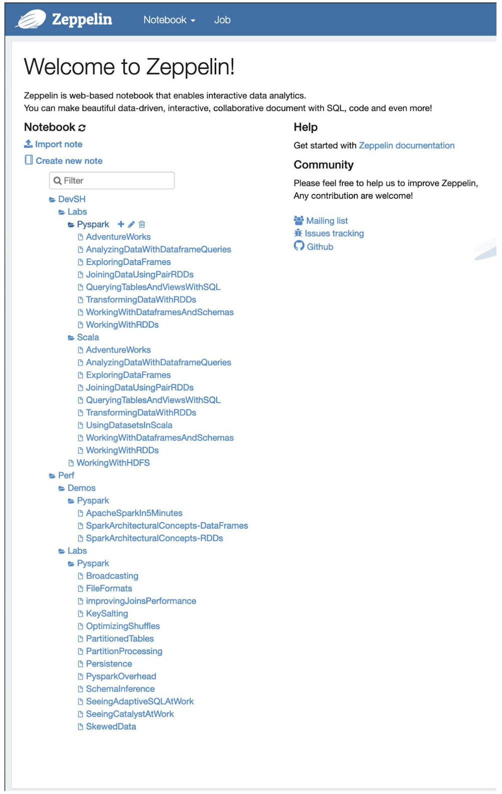
If you want to experiment with the DevSH notebooks, you can install them using this script:

```
cd /home/training/training_materials/perf/install/scripts
sh uploadAllNotebooks.sh localhost.localdomain:8885 'userName=admin&password=admin'
/home/training/training_materials/devsh/notebooks/DevSH-notebooks-20200918.zip
```

CLOUDERA
Educational Services

Notebooks Folders Structure

- DevSH
  - Labs
    - Pyspark
    - Scala
  - Demos
    - Pyspark
    - Scala
- Perf
  - Labs
    - Pyspark
  - Demos
    - Pyspark



Welcome to Zeppelin!

Zeppelin is web-based notebook that enables interactive data analytics.
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

Notebook

Import note
Create new note
Filter

- DevSH
  - Labs
    - Pyspark
      - AdventureWorks
      - AnalyzingDataWithDataframeQueries
      - ExploringDataFrames
      - JoiningDataUsingPairRDDs
      - QueryingTablesAndViewsWithSQL
      - TransformingDataWithRDDs
      - WorkingWithDataframesAndSchemas
      - WorkingWithRDDs
    - Scala
      - AdventureWorks
      - AnalyzingDataWithDataframeQueries
      - ExploringDataFrames
      - JoiningDataUsingPairRDDs
      - QueryingTablesAndViewsWithSQL
      - TransformingDataWithRDDs
      - UsingDatasetsInScala
      - WorkingWithDataframesAndSchemas
      - WorkingWithRDDs
    - WorkingWithHDFS
- Perf
  - Demos
    - Pyspark
      - ApacheSparkIn5Minutes
      - SparkArchitecturalConcepts-DataFrames
      - SparkArchitecturalConcepts-RDDs
  - Labs
    - Pyspark
      - Broadcasting
      - FileFormats
      - improvingJoinsPerformance
      - KeySalting
      - OptimizingShuffles
      - PartitionedTables
      - PartitionProcessing
      - Persistence
      - PysparkOverhead
      - SchemaInference
      - SeeingAdaptiveSQLAtWork
      - SeeingCatalystAtWork
      - SkewedData

Help

Get started with Zeppelin documentation

Community

Please feel free to help us to improve Zeppelin,
Any contribution are welcome!

Mailing list
Issues tracking
Github

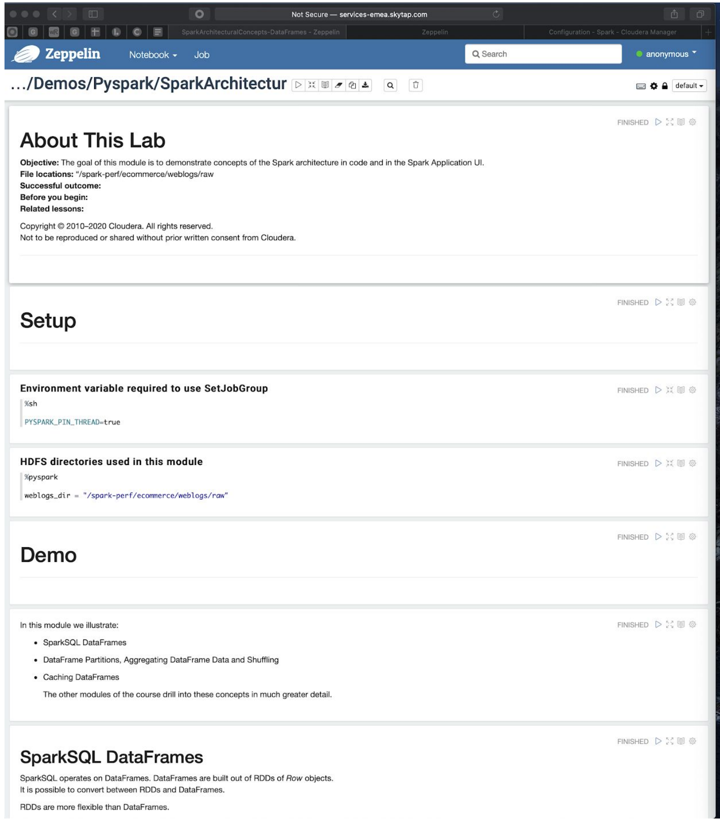
Notebooks List

- Most of the notebooks are Zeppelin transpositions of their CDSW counterparts:
  - /Demos/Pyspark/ApacheSparkArchitecturalConcepts-RDDs
  - /Demos/Pyspark/ApacheSparkArchitecturalConcepts-DataFrames
  - /Labs/Pyspark/FileFormats
  - /Labs/Pyspark/SchemaInference
  - /Labs/Pyspark/SkewedData
  - /Labs/Pyspark/OptimizingShuffles
  - /Labs/Pyspark/PartitionedTables
  - /Labs/Pyspark/ImprovingJoinsPerformance
  - /Labs/Pyspark/PysparkOverhead
  - /Labs/Pyspark/Persistence
  - /Labs/Pyspark/PartitionProcessing
  - /Labs/Pyspark/Broadcasting
- Four are new:
  - /Demos/Pyspark/ApacheSparkIn5Minutes
  - /Labs/Pyspark/SeeingCatalystAtWork
  - /Labs/Pyspark/KeySalting
  - /Labs/Pyspark/SeeingAdaptiveSQLAtWork

Main changes brought by the migration to Zeppelin

- External visualization packages such as Seaborn => replaced by Zeppelin internal visualizations
- Pandas dataframe in the FileFormats notebook to store results => replaced by a Hive table
- %time no longer available => replaced by the footer of each Zeppelin paragraph

Exhaustive use of setJobGroup


.../Demos/Pyspark/SparkArchitectur

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.

Setup

Environment variable required to use SetJobGroup

```
%sh

PYSPARK_PIN_THREAD=true
```

HDFS directories used in this module

```
%pyspark

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

Demo

In this module we illustrate:

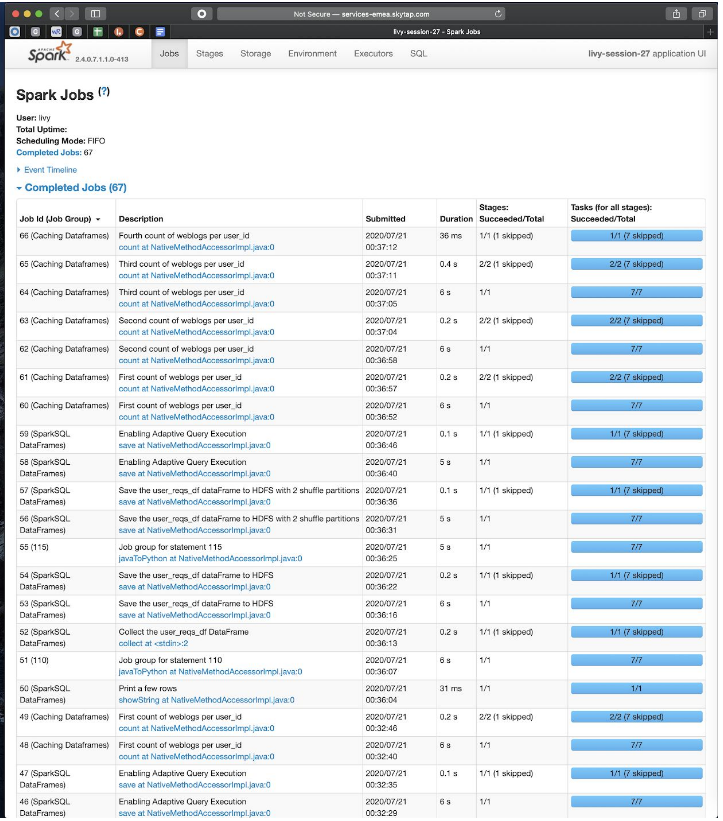
- SparkSQL DataFrames
- DataFrame Partitions, Aggregating DataFrame Data and Shuffling
- Caching DataFrames

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

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.


Spark 2.4.0.7.1.1.0-413 — 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

CLOUDERA
Educational Services

The Paragraphs

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

CLOUDERA
Educational Services

New content

Course components

Component	Version Control	Version
Students Presentation	Google Docs internal version control	SparkApplicationPerformanceTuningWithCDP-20201026.pdf
16 Zeppelin notebooks	Zip file name includes date	notebooks20201002.zip
8 minutes demo video of WXM for Spark	Non applicable	WXM Deep Dive - Mar 20 - Spark.mp4
Skytap template	Template name includes date	TM713 - Spark Performance - 20201002 - CDP_7.1.3 - Spark 3.0.1 - 150GiB
Course outline	Google Docs internal version control	perf-20201026-course-outline.pdf
TTT presentation	File name includes date	SparkApplicationPerformanceTuningWithCDP-20201026TTT.pdf

Agenda

Day 1	Day 2	Day 3	Optional (*)
Introduction	Dealing with Skewed Data	Pyspark Overhead and UDFs	Partition Processing
Spark Architecture	**Catalyst and Tungsten Overview**	Caching Data for Reuse	Broadcasting
Data Sources and Formats	Mitigating Spark Shuffles	**WXM Introduction**	Scheduling
Inferring Schemas	Partitioned and Bucketed Tables	**What's New in Spark 3.0?**	
	Improving Joins Performance		

(*) if time allows

Catalyst and Tungsten Overview

- Content taken from legacy Hortonworks slides
- Added a short notebook to illustrate what Catalyst does

WXM Introduction

- Sliced and diced several WXM presentations to create this content
- Demoing WXM requires a WXM cluster and relevant historical data
  - This proves to be very demanding requirements
  - So I just edited an excellent demo from Raman to just take the 8 minutes about Spark
  - It works great with students.

Similar content should be added to our admin and data analyst classes

What's New in Spark 3.0?

- Sliced and diced several internal presentations
- Added some content and graphics
- Created a notebook to illustrate the optimizations that AQE brings using TPCDS queries
  - I wanted to use realistic complex queries

Optional content

- Partition processing
  - RDD specific
- Broadcasting
  - RDD specific
- Scheduling
  - More of mixed admin/developer topic
  - The labs no longer made sense when each student has its own cluster
  - If you want to see resource contention, try to launch a spark-shell and a Zeppelin notebook at the same time!

CLOUDERA
Educational Services

Troubleshooting

The images do not appear in the notebooks

- That will happen is you stop Zeppelin and start it again.

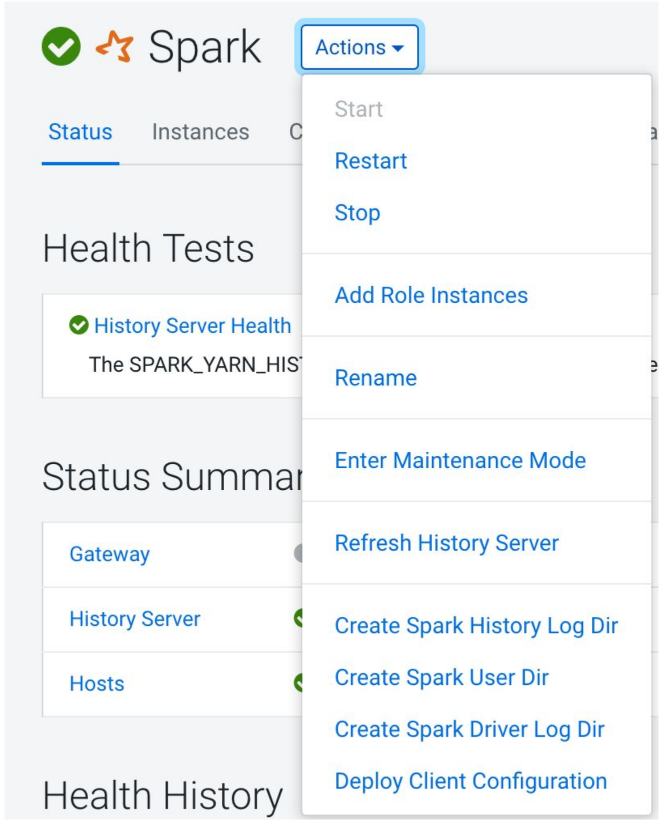
  The specific folder where the images must be located seems to be flushed when Zeppelin is stopped.

  I created a script to restore the images.

```
sh /home/training/training_materials/perf/install/scripts/copyImages.sh
```

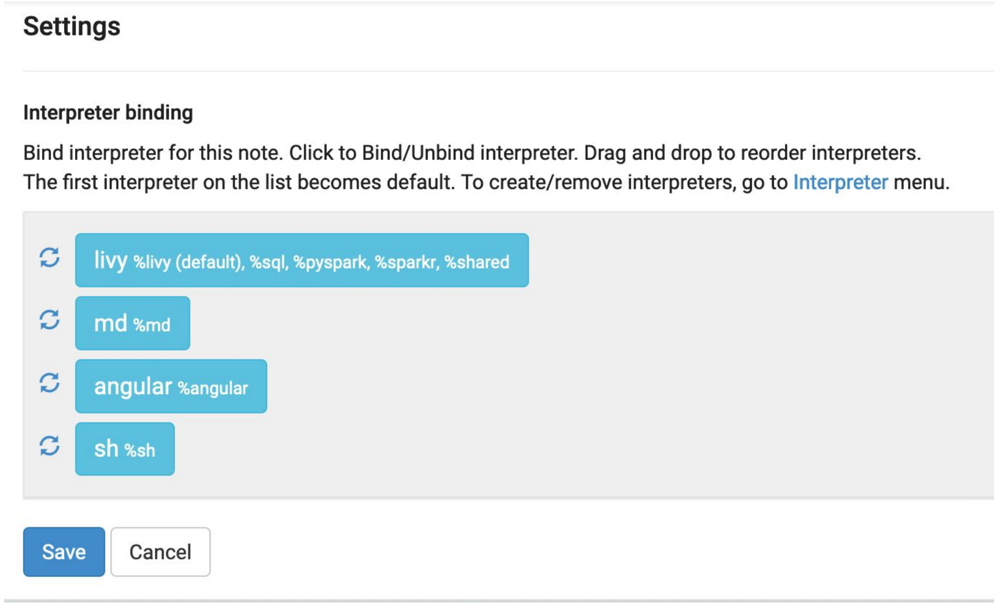
The Spark History Server becomes weak in the knees

- 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.

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
- Watch the 8 minutes demo video of WXM
- Launch an instance of the Skytap template
- Run through the notebooks

Outstanding items in my to do list

- Finding an example of skew optimization in the TPCDS queries
- Finding an example of dynamic partition pruning in the TPCDS queries

Ideas for future evolution

- Keep updating CDP
- Keep updating Spark
- When Zeppelin becomes supported for Spark3
  - test the DevSH notebooks for regression
  - test the remaining labs: Spark Streaming
- Consider storing the datasets in S3 for additional modularity
- Consider replacing the Spark Streaming content in DevSH by a Flink equivalent to create a CDP Data Engineering course

CLOUDERA
Educational Services

Questions

CLOUDERA
Educational Services

THANK YOU