Amazon Web Services (AWS) en tiempos de Big Data

Dr. Andrea Villanes

Materiales:

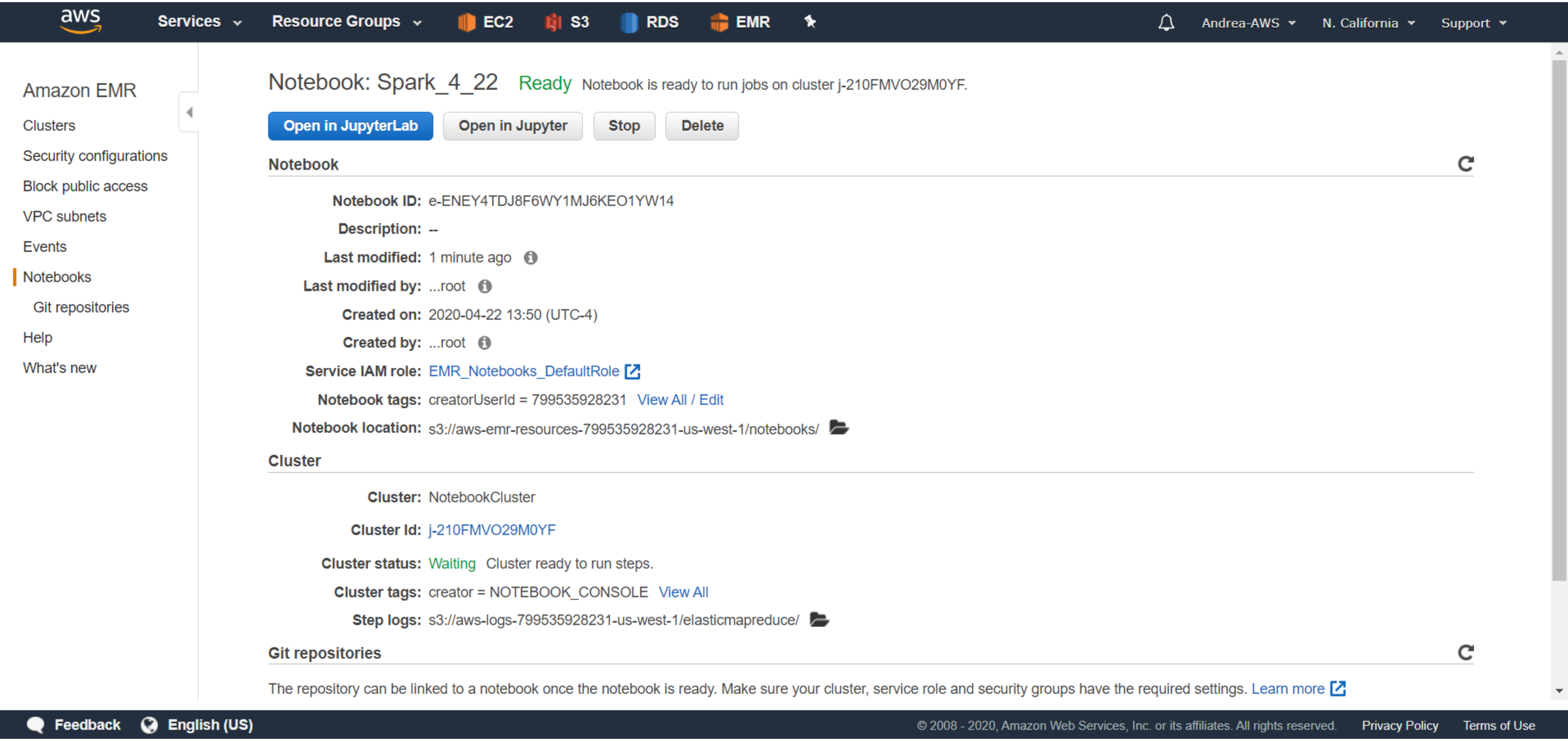
https://github.com/andreavillanes/AWS_EMR

Un poco sobre mi…

- PhD en Computer Science NC STATE UNIVERSITY
- Assistant Professor Institute for ADVANCED ANALYTICS aws certified Cloud Practitioner
- Master in Science of Analytics NC STATE UNIVERSITY
- Gracias DMC por la organizacion! DMC Perú

hadoop

APACHE
Spark™



Notebook: Spark_4_22 Ready Notebook is ready to run jobs on cluster j-210FMVO29M0YF.

Notebook

Notebook ID:	e-ENEY4TDJ8F6WY1MJ6KEO1YW14
Description:	--
Last modified:	1 minute ago
Last modified by:	...root
Created on:	2020-04-22 13:50 (UTC-4)
Created by:	...root
Service IAM role:	EMR_Notebooks_DefaultRole
Notebook tags:	creatorUserId = 799535928231 View All / Edit
Notebook location:	s3://aws-emr-resources-799535928231-us-west-1/notebooks/

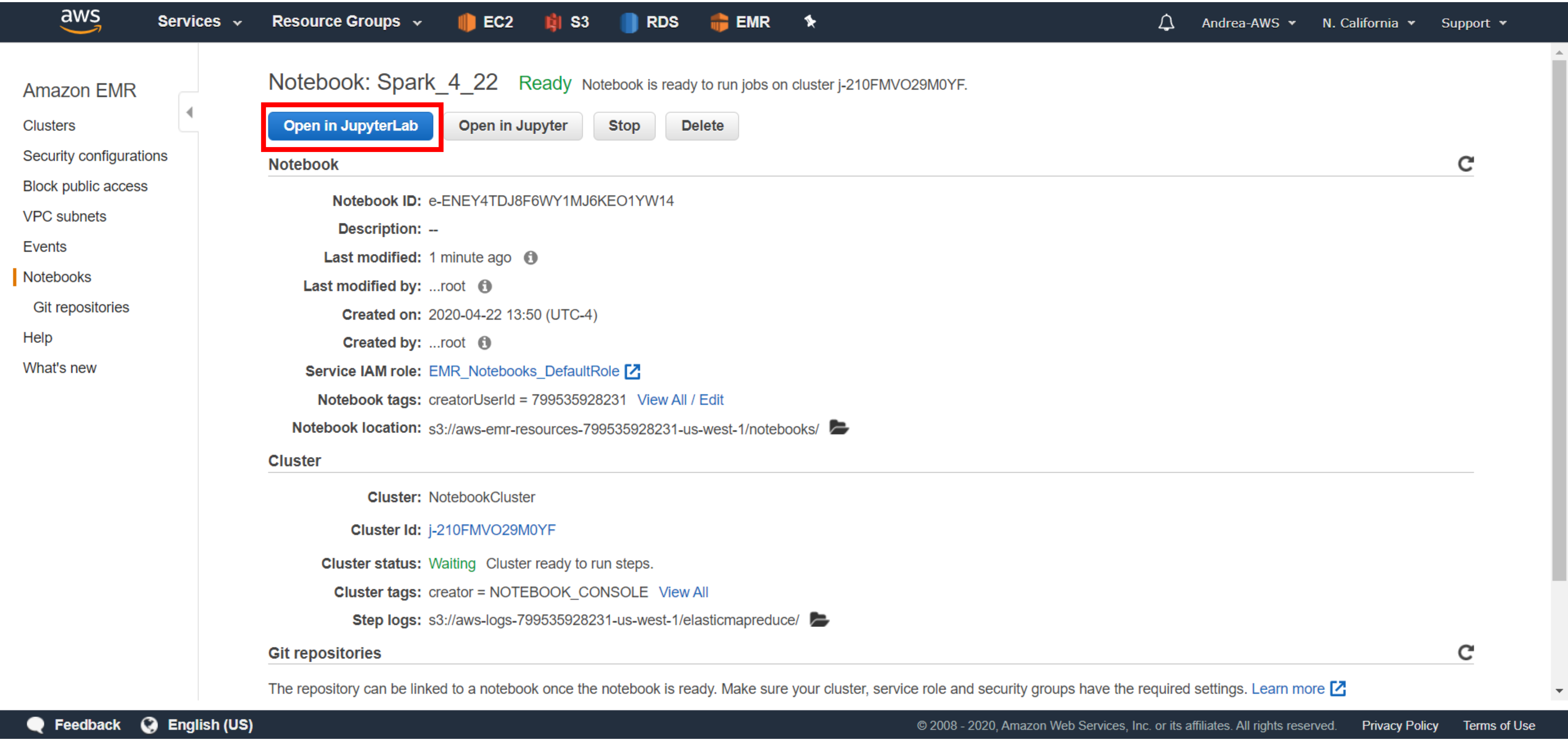
Cluster

Cluster:	NotebookCluster
Cluster Id:	j-210FMVO29M0YF
Cluster status:	Waiting Cluster ready to run steps.
Cluster tags:	creator = NOTEBOOK_CONSOLE View All
Step logs:	s3://aws-logs-799535928231-us-west-1/elasticmapreduce/

Git repositories

The repository can be linked to a notebook once the notebook is ready. Make sure your cluster, service role and security groups have the required settings. Learn more

Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



Amazon EMR
- Clusters
- Security configurations
- Block public access
- VPC subnets
- Events
- Notebooks
  - Git repositories
- Help
- What's new

Notebook: Spark_4_22 Ready Notebook is ready to run jobs on cluster j-210FMVO29M0YF.

Notebook

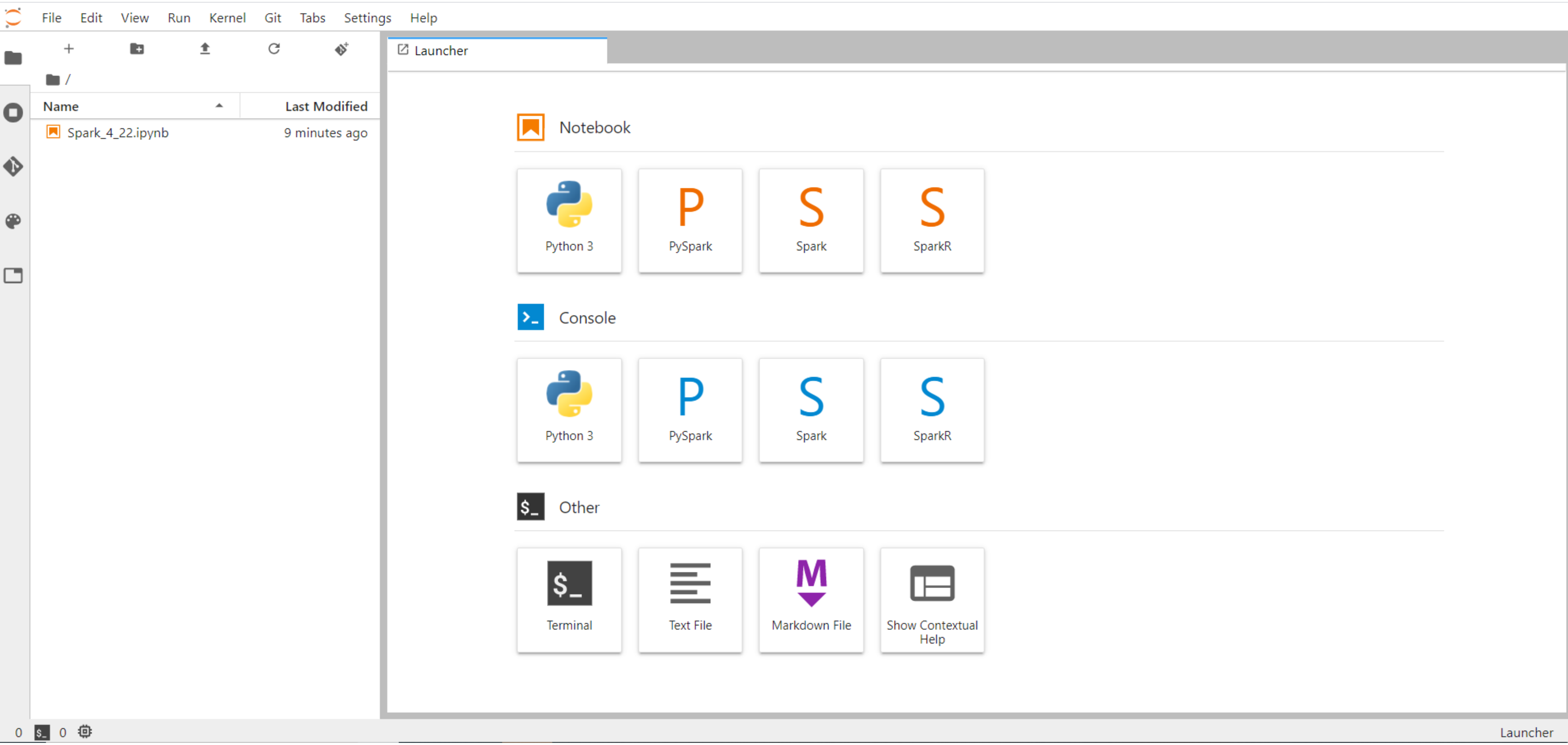
Notebook ID:	e-ENEY4TDJ8F6WY1MJ6KEO1YW14
Description:	--
Last modified:	1 minute ago
Last modified by:	...root
Created on:	2020-04-22 13:50 (UTC-4)
Created by:	...root
Service IAM role:	EMR_Notebooks_DefaultRole
Notebook tags:	creatorUserId = 799535928231 View All / Edit
Notebook location:	s3://aws-emr-resources-799535928231-us-west-1/notebooks/

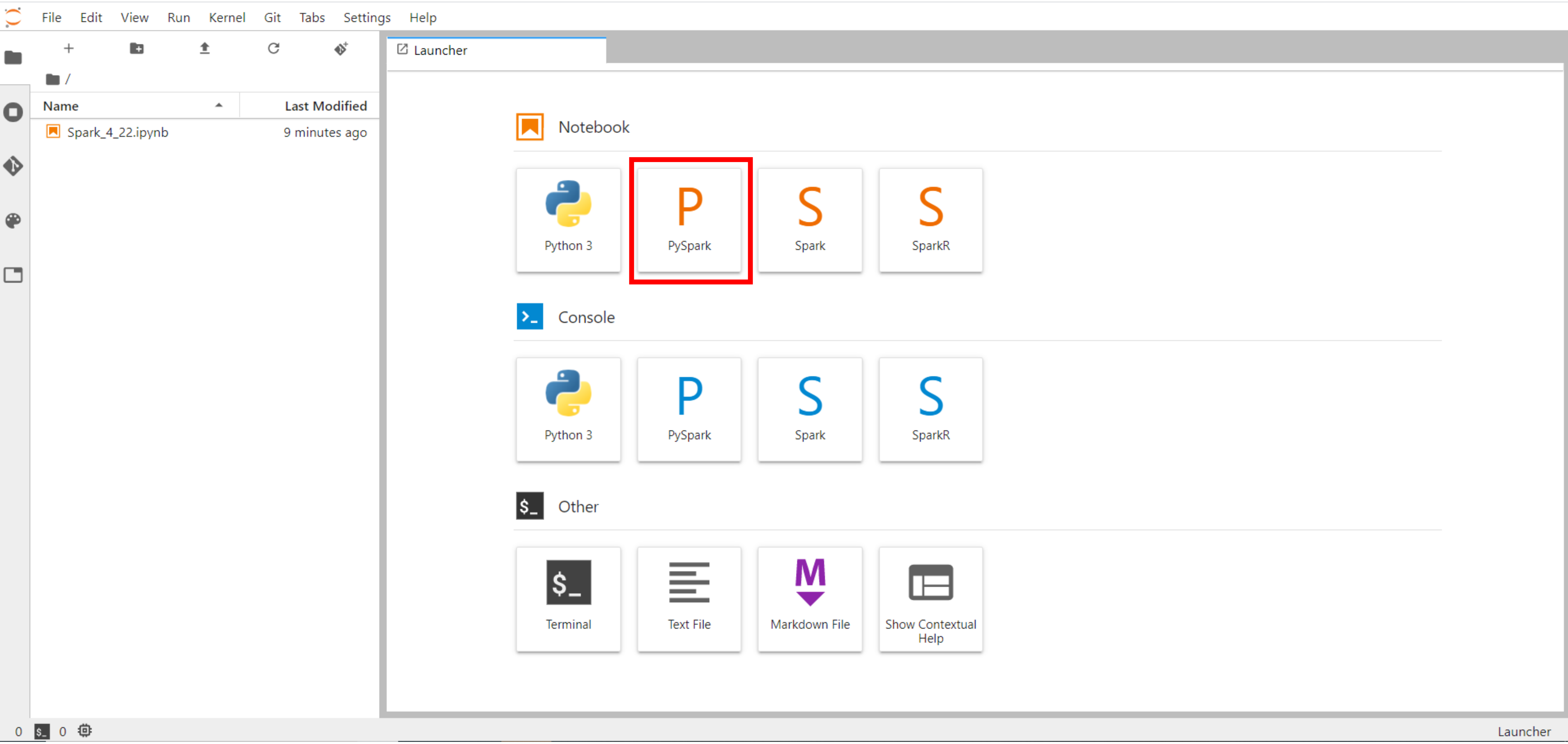
Cluster

Cluster:	NotebookCluster
Cluster Id:	j-210FMVO29M0YF
Cluster status:	Waiting Cluster ready to run steps.
Cluster tags:	creator = NOTEBOOK_CONSOLE View All
Step logs:	s3://aws-logs-799535928231-us-west-1/elasticmapreduce/

Git repositories

The repository can be linked to a notebook once the notebook is ready. Make sure your cluster, service role and security groups have the required settings. Learn more


File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
/
Name
Last Modified
Spark_4_22.ipynb
9 minutes ago
Launcher
Notebook
Python 3
PySpark
Spark
SparkR
Console
Python 3
PySpark
Spark
SparkR
Other
Terminal
Text File
Markdown File
Show Contextual Help
0
0
Launcher


File
Edit
View
Run
Kernel
Git
Tabs
Settings
Help
/
Name
Last Modified
Spark_4_22.ipynb
9 minutes ago
Launcher
Notebook
Python 3
PySpark
Spark
SparkR
Console
Python 3
PySpark
Spark
SparkR
Other
Terminal
Text File
Markdown File
Show Contextual Help
0
0
Launcher

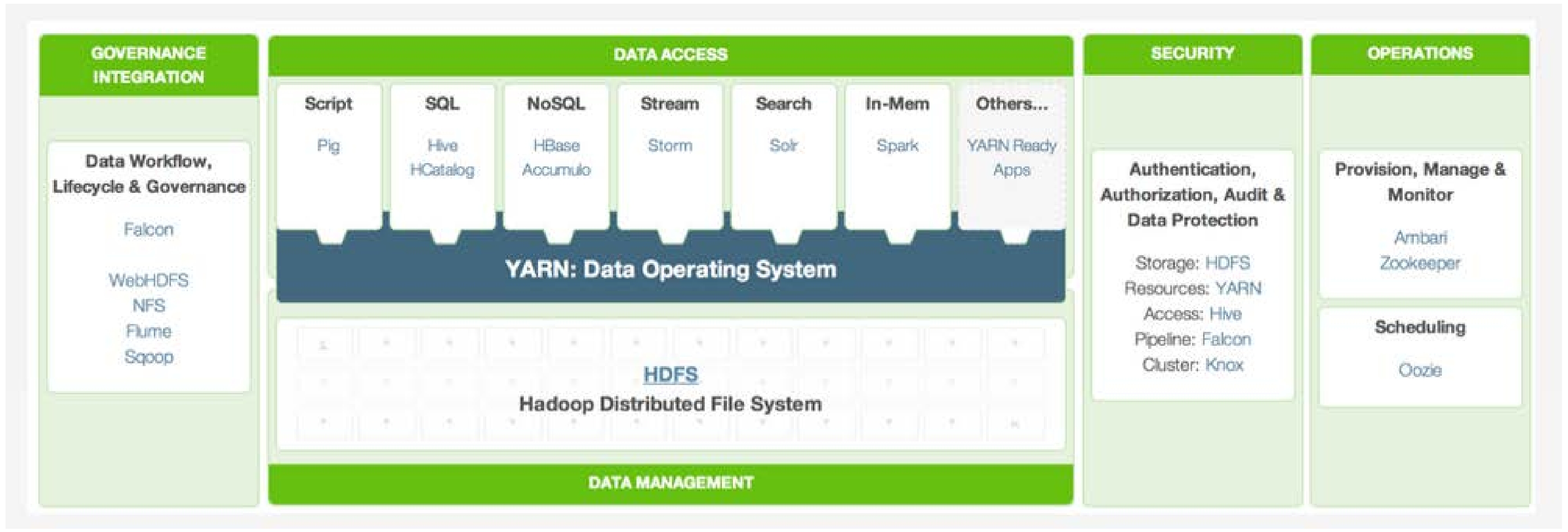
Agenda

- Apache Hadoop
- Apache Spark
- Distribuciones Comerciales de Hadoop
- AWS Elastic Map Reduce (EMR)
- PySpark
- Spark SQL
- Spark MLlib
- Como abrir una cuenta de AWS gratis?

Motivacion para Hadoop

- Velocidad
- Variedad
- Volumen
- Data tiene valor
- Dos problemas que tenemos que resolver:
  1. Como podemos almacenar grandes cantidades de datos a un costo razonable?
  2. Como podemos analizar la data que hemos almacenado?

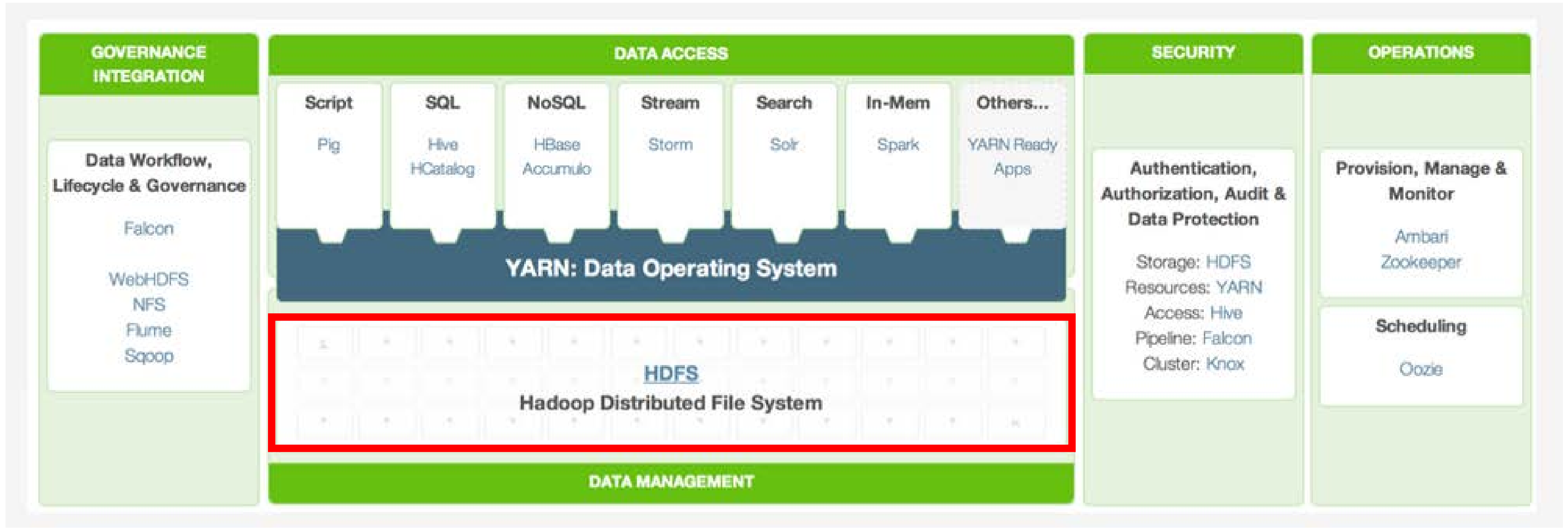
Apache Hadoop



Beneficios:

1. Escalibilidad
2. Tolerancia a las fallas

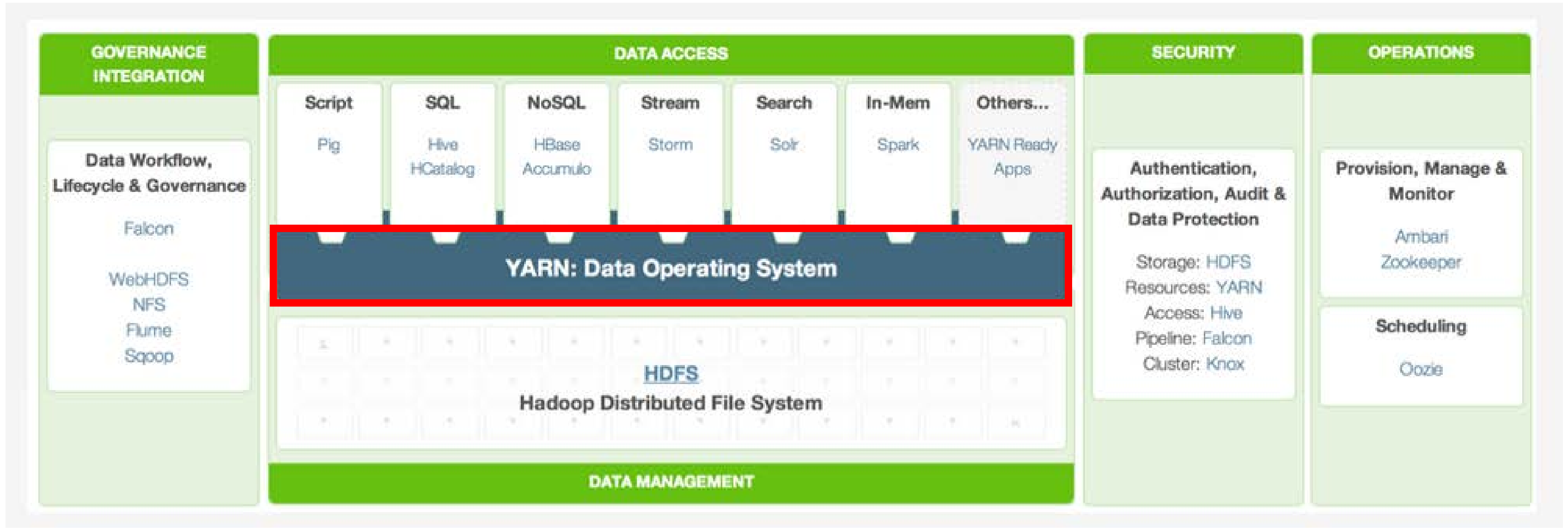
Apache Hadoop



Beneficios:

1. Escalibilidad
2. Tolerancia a las fallas

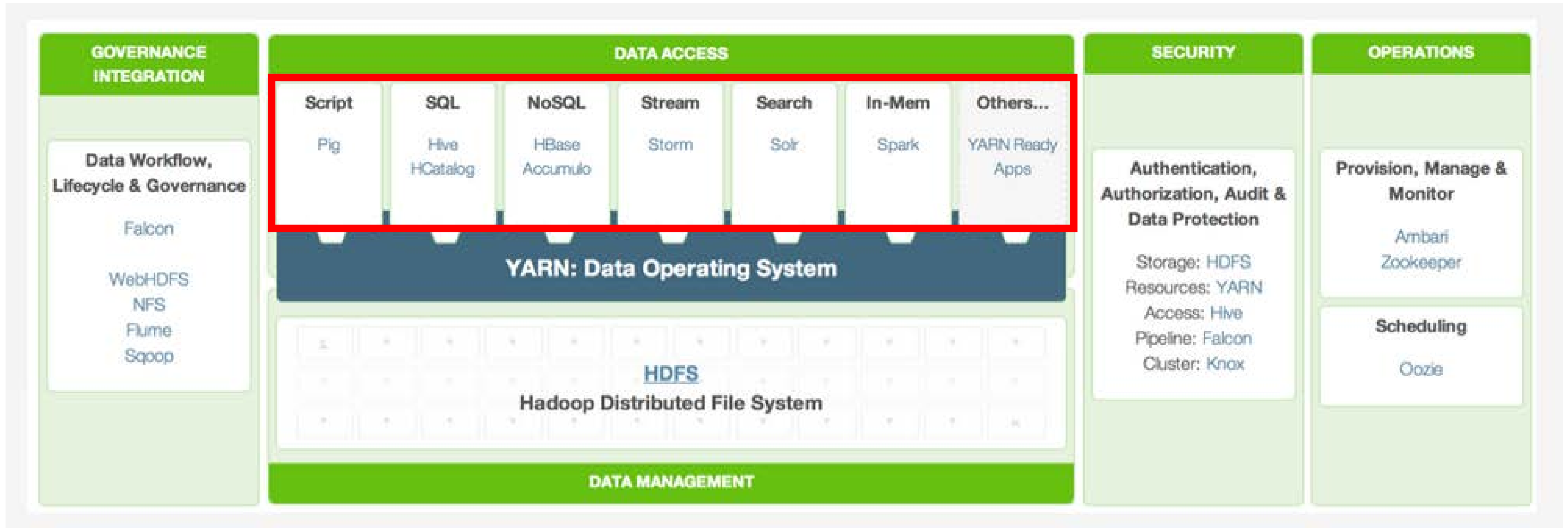
Apache Hadoop



Beneficios:

1. Escalibilidad
2. Tolerancia a las fallas

Apache Hadoop



Beneficios:

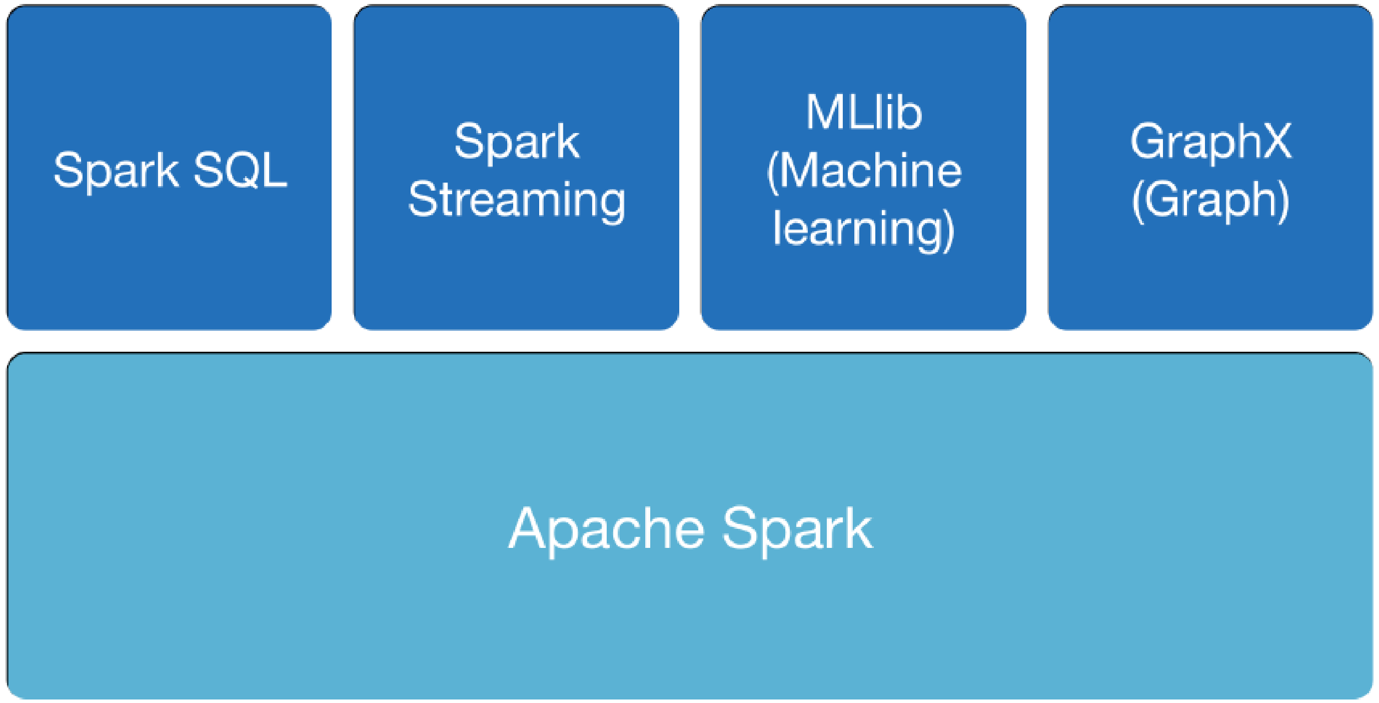
1. Escalibilidad
2. Tolerancia a las fallas

Motivacion para Spark

- Focus en operaciones interactivas e iterativas
- Utiliza procesamiento en memoria
- Los hace ideal para aplicaciones de data science (data mining, machine learning)
- Extensive API support for Java, Scala, R and Python
- **Runs Everywhere:** Spark runs on Hadoop, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, and AWS S3.

Apache Spark Components

- Combine SQL, streaming, and complex analytics.



Preguntas?

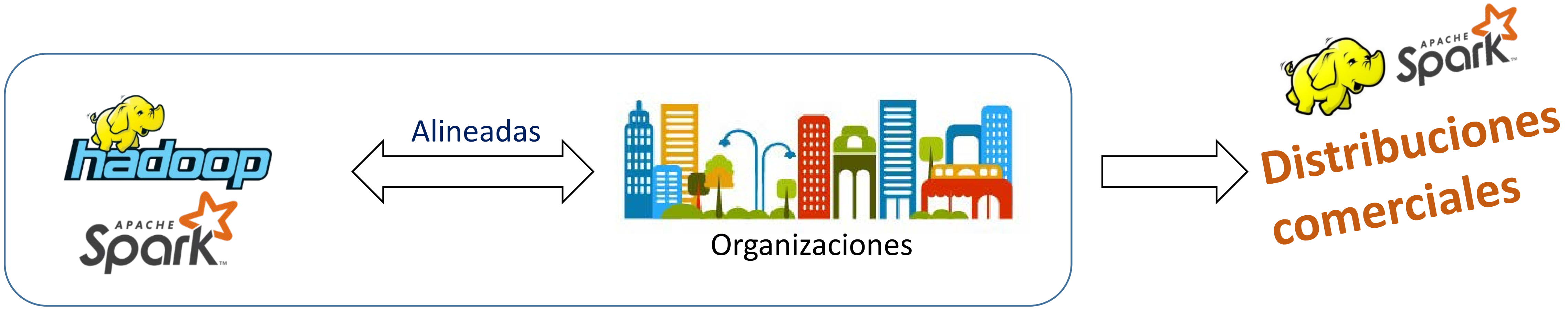
Agenda

- Apache Hadoop
- Apache Spark
- Distribuciones Comerciales de Hadoop
- AWS Elastic Map Reduce (EMR)
- PySpark
- Spark SQL
- Spark MLlib
- Como abrir una cuenta de AWS gratis?

Apache Hadoop y Apache Spark son proyectos open source

$\Longrightarrow$ open source

Necesitamos…



Necesitamos que los requerimientos para Hadoop esten alineados con las necesidades de las empresas, y eso resulta en el nacimiento de distribuciones comerciales.

Ejemplos de Hadoop vendors que venden una Distribucion de Hadoop

HORTONWORKS®
POWERING THE FUTURE OF DATA™

cloudera

MAPR

Microsoft Azure

aws

Google Cloud Platform

Distribuciones de Hadoop

Hortonworks Data Platform (HDP)

Cloudera CDH

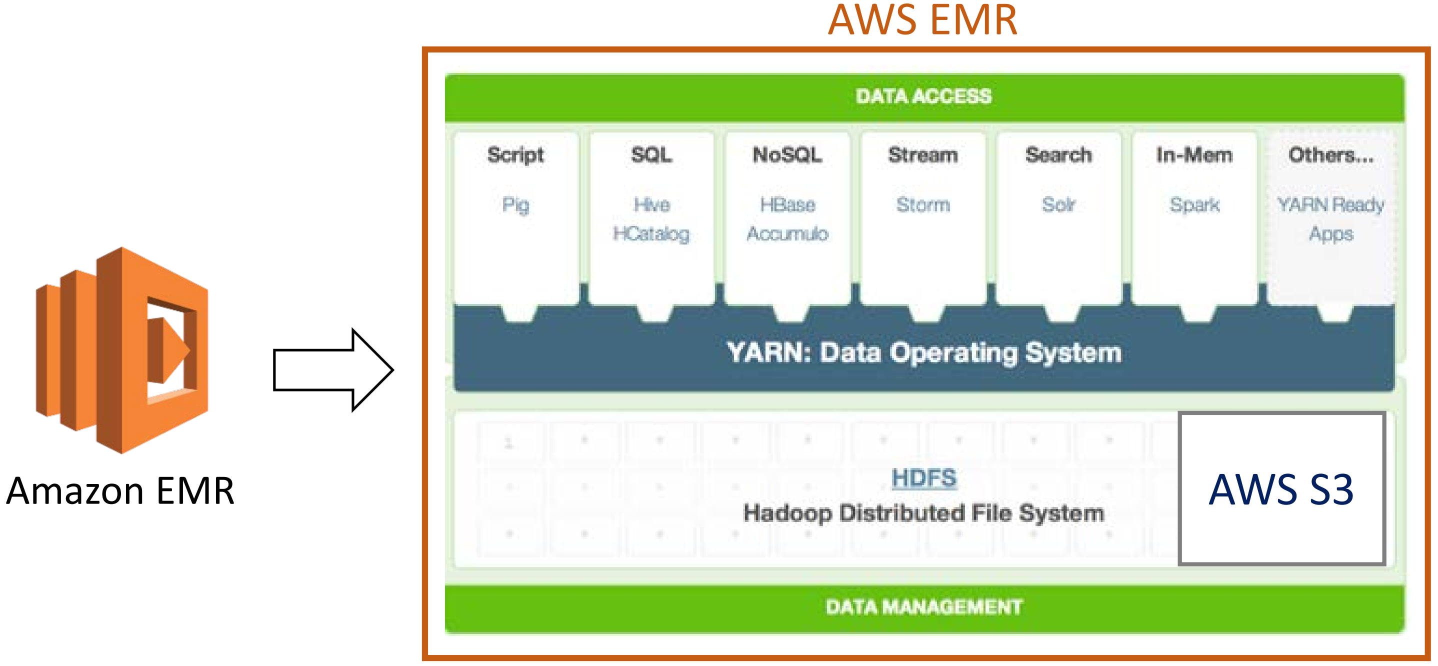
MapR Distribution

Microsoft's Azure HDInsight

Amazon Elastic MapReduce (EMR)

Cloud Dataproc

AWS Hadoop Distribution: Amazon Elastic MapReduce (EMR)



- Amazon EMR includes EMRFS, a connector allowing Hadoop to use **S3 as a storage layer**.
- **HDFS** is automatically installed with Hadoop on your EMR cluster, and **you can use HDFS along with Amazon S3 to store your input and output data.**
- Amazon EMR configures Hadoop to use **HDFS for intermediate data** created during MapReduce jobs, even if your input data is located in Amazon S3.

Amazon EMR programmatically installs and configures applications in the Hadoop project, including Hadoop MapReduce (YARN), and HDFS, across the nodes in your cluster.

Amazon Simple Storage Service (S3)

Amazon S3

- S3: **object storage service**.
- Objects are stored in **buckets**.
- Natively online, **HTTP access.**
- Every object in Amazon S3 can be **uniquely addressed through the combination of the web service endpoint, bucket name, key, and optionally, a version.**
- **Store and retrieve any amount of data,** any time, from anywhere on the web.
- Amazon S3 is **highly scalable, reliable, low cost, and designed for durability**.
- Data **can be stored as-is**: there is no need to convert it to a predefined schema.

Preguntas?

Agenda

- Apache Hadoop
- Apache Spark
- Distribuciones Comerciales de Hadoop
- AWS Elastic Map Reduce (EMR)
- PySpark
- Spark SQL
- Spark MLlib
- Como abrir una cuenta de AWS gratis?

PySpark

Python + Spark = PySpark

- PySpark is the collaboration of Apache Spark and Python.
- **Apache Spark** is an open-source cluster-computing framework, built around speed, ease of use, and streaming analytics.
- **Python** is a general-purpose, high-level programming language.

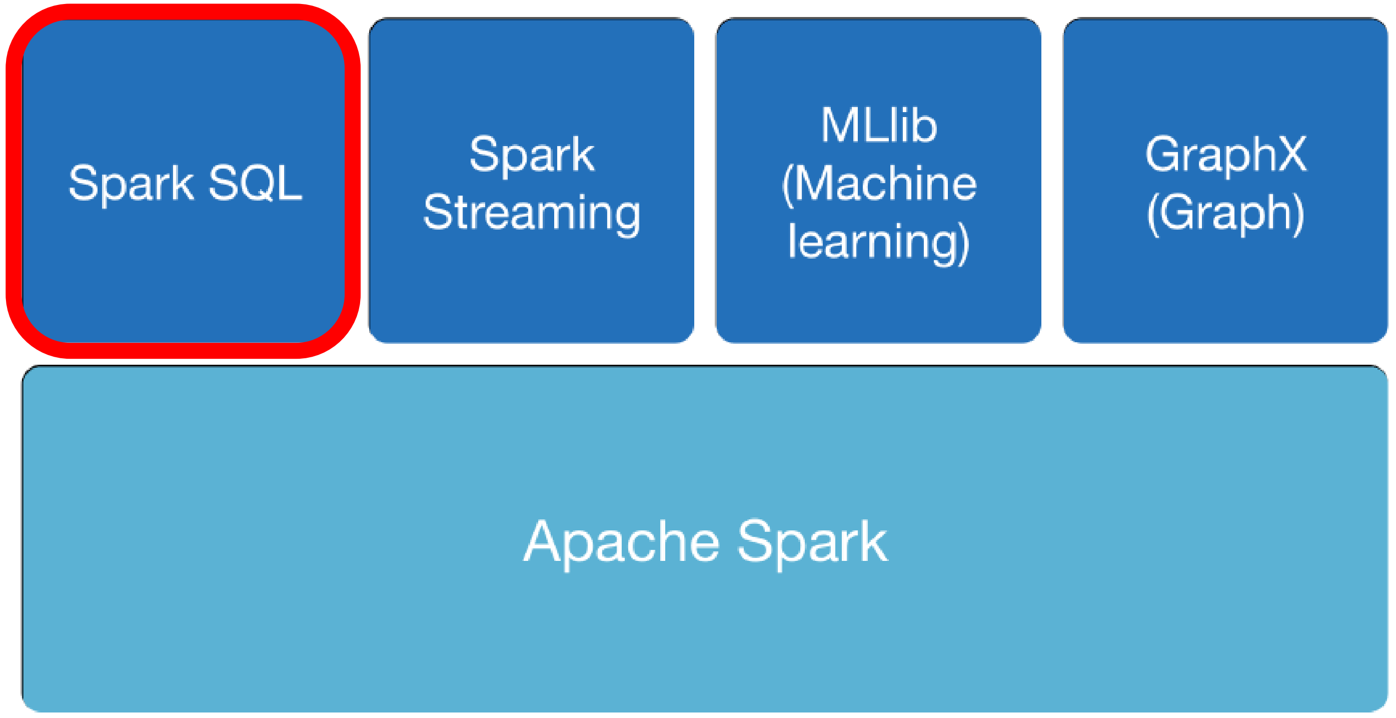
```
df = spark.read.json("logs.json")
df.where("age > 21")
  .select("name.first").show()
```

Spark's Python DataFrame API

Spark SQL

Apache Spark Components

- Combine SQL, streaming, and complex analytics.



Apache Spark SQL

- Spark SQL is a Spark's module for working with **structured data.**
- There are several ways to interact with Spark SQL including SQL, the DataFrames API and the Datasets API. However, PySpark only implements the DataFrames API.
- A DataFrame is a distributed collection of data organized into named columns. It is **conceptually equivalent to a table in a relational database or a data frame in R/Python**, but with richer optimizations under the hood. DataFrames can be constructed from a wide array of sources such as: structured data files, tables in Hive, external databases, or existing RDDs.
- Executes SQL queries written using either a basic SQL syntax

PySpark & Spark SQL Cheat Sheet

Spark SQL is Apache Spark's module for working with structured data.

Initializing SparkSession

A SparkSession can be used create DataFrame, register DataFrame as views, execute SQL over views, and read csv, json, txt and parquet files.

In EMR Notebooks, SparkSession is automatically created for you. The SparkSession is accessible through a variable called *spark*.
If you want to verify your Spark version:

```
>>> spark.version
```

Creating DataFrames

From Spark Data Sources

CSV

```
>>> df = spark.read.csv("s3a://bucket_name/airlines.csv", inferSchema = True, header=True)
```

JSON

```
>>> df2 = spark.read.json("s3a://bucket_name/customer.json")
>>> df2.show( )
>>> df3 = spark.read.load("s3a://bucket_name/people.json", format="json")
```

Parquet files

```
 >>> df4 = spark.read.load("s3a://bucket_name/users.parquet")
```

TXT files

```
>>> df5 = spark.read.text("s3a://bucket_name/people.txt")
```

View the DataFrame

Show() - Displays the top 20 rows of DataFrame in a tabular form.

```
>>> df.show()
```

Show(n) - Displays the top *n* rows of DataFrame in a tabular form.

```
>>> df.show(n)
```

Inspect Data

`>>> df.describe().show()`	Compute summary statistics
`>>> df.columns`	Return the columns of df
`>>> df.count()`	Count the number of rows in df
`>>> df.distinct().count()`	Count the number of distinct rows in df
`>>> df.printSchema()`	Print the schema of df

Queries

```
>>> from pyspark.sql import functions as F
```

Select

`>>>  df.select ("firstName") .show( )`	**Show all entries in *firstName* column**

`>>> df.select("firstName","lastName")  \`
`        .show( )`
	Show all entries in *firstName*, and *lastName*

`>>> df.select (df ["firstName"] ,df ["age"]+ 1)`	**Show all entries in *firstName* and *age*, add *1* to the entries of *age***
`>>> df.select(df['age'] > 24).show( )`	**Show all entries where *age >24***

When

`>>> df.select ("firstName",`
`                F.when(df.age > 30, 1) \`
`               .otherwise(0)) \`
`        .show( )`
	Show *firstName* and 0 or 1 depending on age > 30

`>>> df[df.firstName.isin ("Jane","Boris")]`
`                    .show( )`
	Show *firstName* if in the given options

Like

` >>> df.select("firstName",`
`               df.lastName.like ("Smith")) \`
`         .show( )`
	Show *firstName*, and *lastName* if *lastName* is like Smith

Startswith – Endswith

`>>> df.select("firstName",`
`               df.lastName \`
`                .startswith("Sm"))  \`
`        .show( )`
	Show *firstName*, and *lastName* if *lastName* starts with *Sm*

`>>> df.select(df.lastName.endswith("th"))  \`
`        .show( )`
	Show last names ending in *th*

Substring

`>>> df.select(df.firstName.substr(1,  3)) \`
`                          .alias("name"))  \`
`        .collect( )`
	Return substrings of *firstName*

Between

`>>> df.select (df.age.between(22, 24)) \`
`        .show()`
	Show *age* if values between 22 and 24

Adding Columns

```
>>> from pyspark.sql.functions import log

>>> df2 = df.withColumn("new_column", log("rating"))
>>> df2.show()
```

Duplicate Values

```
>>> df = df.dropDuplicates( )
```

GroupBy

`>>> df.groupBy("age")  \`
`        .count( )   \`
`        .show( )`
	Group by age, count the members in the groups

Filter

`>>> df.filter(df["age"]>24).show( )`	**Filter entries of age, only keep those records of which the values are > 24**

Missing & Replacing Values

`>>> df.na.fill(50).show( )`	**Replace null values**
`>>> df.na.drop().show()`	**Return new df omitting rows with null values**

`>>> df.na \`
`         .replace(10, 20) \`
`         .show()`
	Return new df replacing one value with another

Running SQL Queries Programmatically

Registering DataFrames as Views

```
>>> df.createOrReplaceTempView("customer")
```

Query Views

```
>>> sqlDF = spark.sql("SELECT * FROM customer").show()
```

Output

Data Structure

`>>> rdd1 = df.rdd`	**Convert *df* into an RDD**
`>>> df.toPandas()`	**Return the contents of *df* as Pandas *DataFrame***

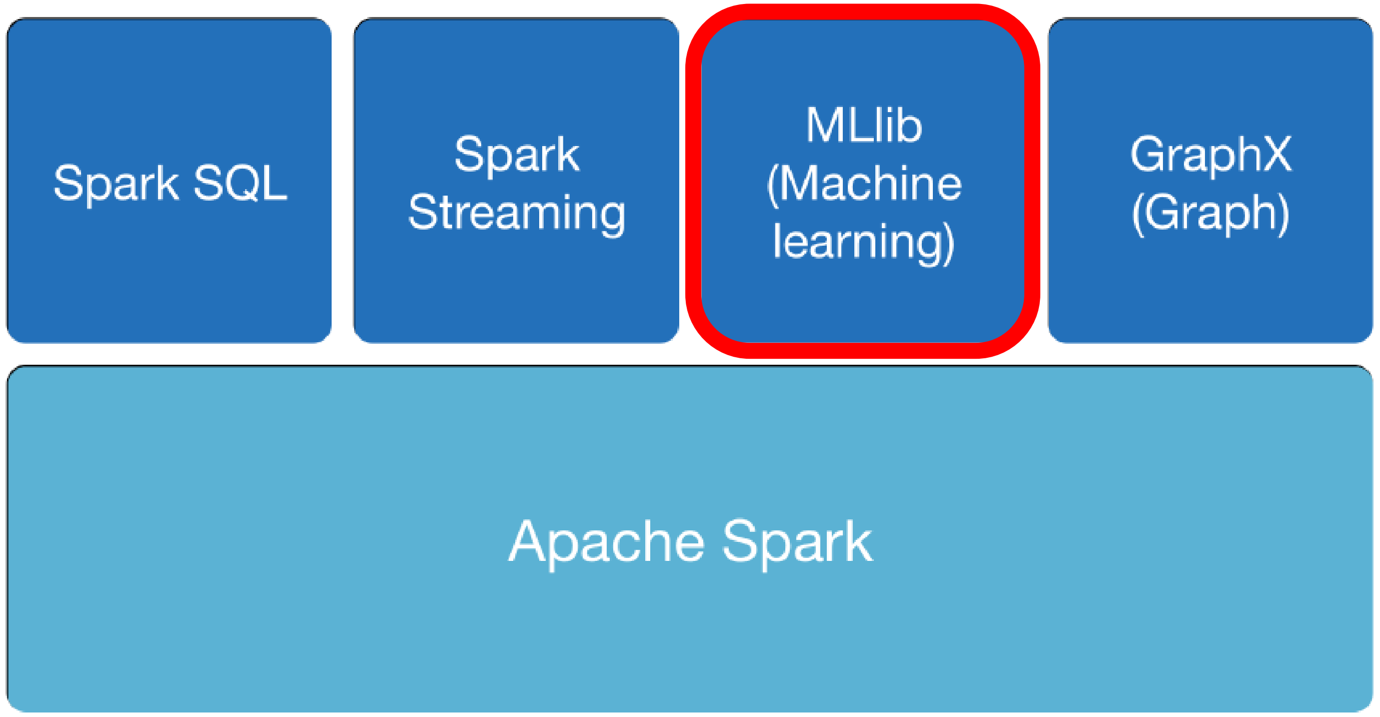
Write & Save to Files

```
>>> df.select("firstName", "city") .write.save("nameAndCity.parquet")
>>> df.select("firstName", "age") .write  .save("namesAndAges.json",format="json")
```

Spark MLlib

Apache Spark Components

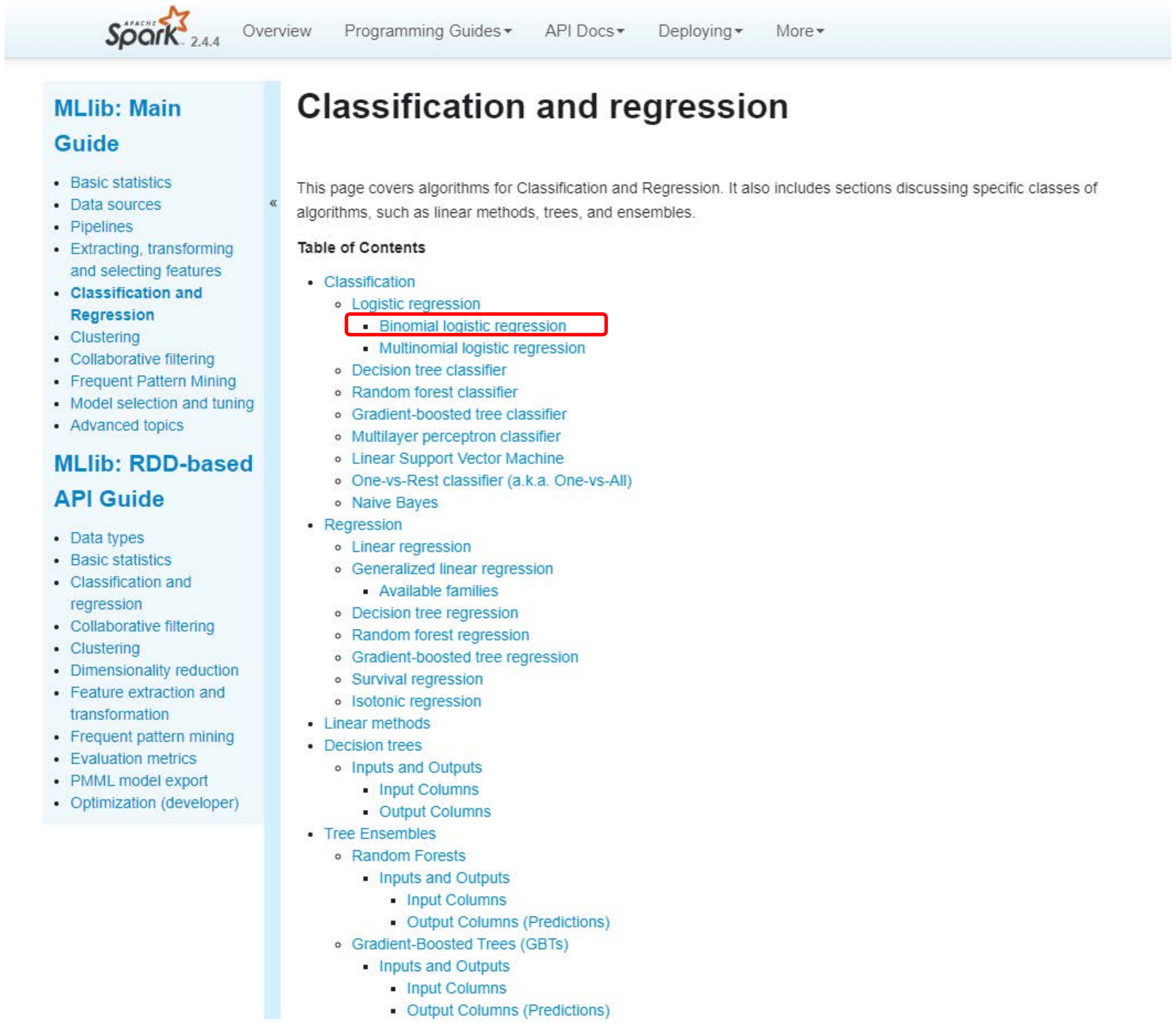
- Combine SQL, streaming, and complex analytics.



Apache Spark MLlib

- MLlib is developed as part of the Apache Spark project. It thus gets tested and updated with each Spark release.
- MLlib is Spark’s scalable machine learning library consisting of common learning algorithms and utilities, including **classification, regression, clustering, collaborative filtering, dimensionality reduction**, and more.
- List of algorithms implemented in MLlib: http://spark.apache.org/docs/latest/ml-guide.html

Spark Documentation: http://spark.apache.org/docs/latest/ml-guide.html



MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- **Classification and Regression**
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

MLlib: RDD-based API Guide

- Data types
- Basic statistics
- Classification and regression
- Collaborative filtering
- Clustering
- Dimensionality reduction
- Feature extraction and transformation
- Frequent pattern mining
- Evaluation metrics
- PMML model export
- Optimization (developer)

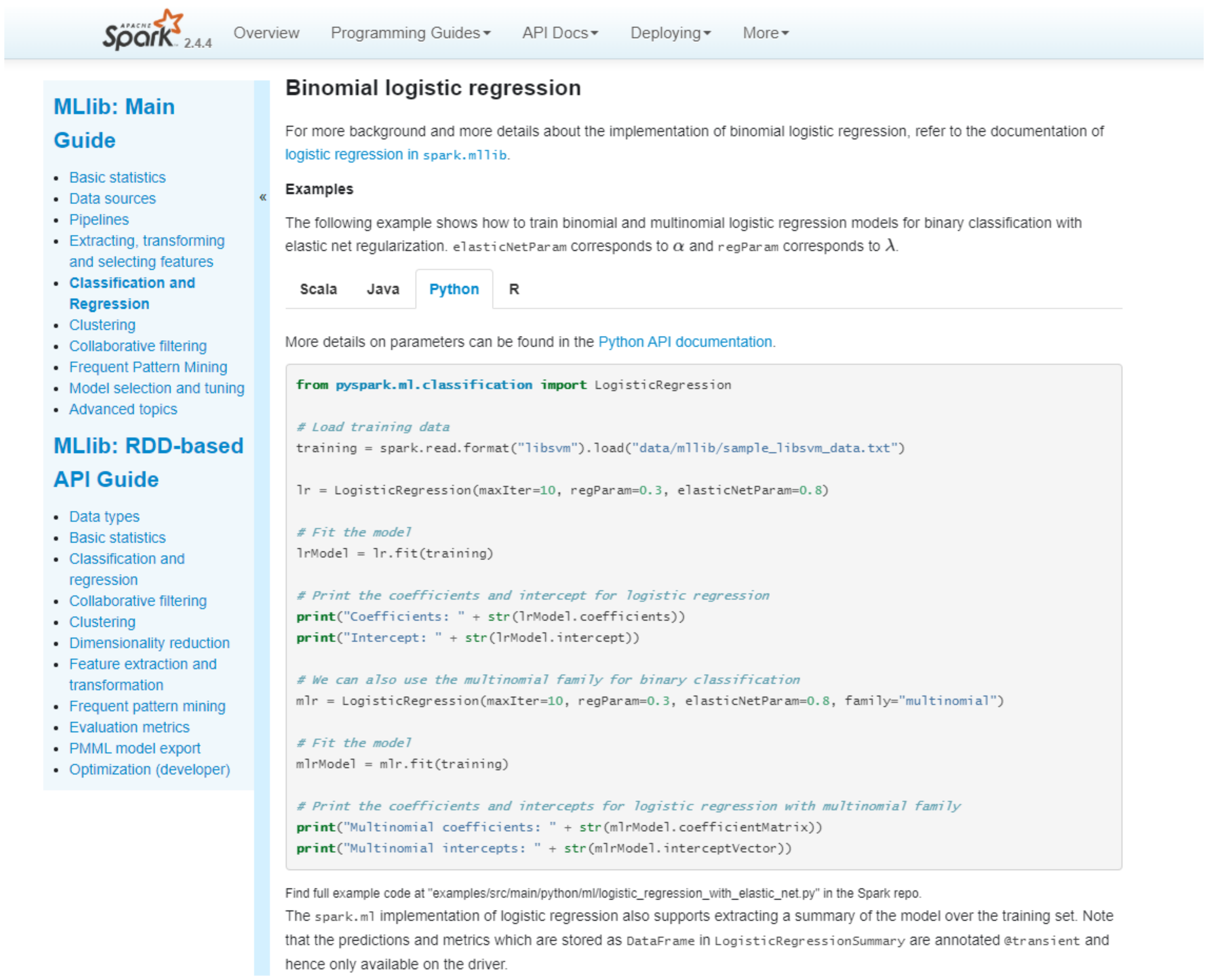
Classification and regression

This page covers algorithms for Classification and Regression. It also includes sections discussing specific classes of algorithms, such as linear methods, trees, and ensembles.

Table of Contents

- Classification
  - Logistic regression
    - Binomial logistic regression
    - Multinomial logistic regression
  - Decision tree classifier
  - Random forest classifier
  - Gradient-boosted tree classifier
  - Multilayer perceptron classifier
  - Linear Support Vector Machine
  - One-vs-Rest classifier (a.k.a. One-vs-All)
  - Naive Bayes
- Regression
  - Linear regression
  - Generalized linear regression
    - Available families
  - Decision tree regression
  - Random forest regression
  - Gradient-boosted tree regression
  - Survival regression
  - Isotonic regression
- Linear methods
- Decision trees
  - Inputs and Outputs
    - Input Columns
    - Output Columns
- Tree Ensembles
  - Random Forests
    - Inputs and Outputs
      - Input Columns
      - Output Columns (Predictions)
  - Gradient-Boosted Trees (GBTs)
    - Inputs and Outputs
      - Input Columns
      - Output Columns (Predictions)

Spark Documentation: http://spark.apache.org/docs/latest/ml-guide.html



MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- **Classification and Regression**
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

MLlib: RDD-based API Guide

- Data types
- Basic statistics
- Classification and regression
- Collaborative filtering
- Clustering
- Dimensionality reduction
- Feature extraction and transformation
- Frequent pattern mining
- Evaluation metrics
- PMML model export
- Optimization (developer)

Binomial logistic regression

For more background and more details about the implementation of binomial logistic regression, refer to the documentation of logistic regression in `spark.mllib`.

Examples

The following example shows how to train binomial and multinomial logistic regression models for binary classification with elastic net regularization. `elasticNetParam` corresponds to α and `regParam` corresponds to λ.

More details on parameters can be found in the Python API documentation.

```python
from pyspark.ml.classification import LogisticRegression

# Load training data
training = spark.read.format("libsvm").load("data/mllib/sample_libsvm_data.txt")

lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)

# Fit the model
lrModel = lr.fit(training)

# Print the coefficients and intercept for logistic regression
print("Coefficients: " + str(lrModel.coefficients))
print("Intercept: " + str(lrModel.intercept))

# We can also use the multinomial family for binary classification
mlr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8, family="multinomial")

# Fit the model
mlrModel = mlr.fit(training)

# Print the coefficients and intercepts for logistic regression with multinomial family
print("Multinomial coefficients: " + str(mlrModel.coefficientMatrix))
print("Multinomial intercepts: " + str(mlrModel.interceptVector))
```

Find full example code at "examples/src/main/python/ml/logistic_regression_with_elastic_net.py" in the Spark repo.

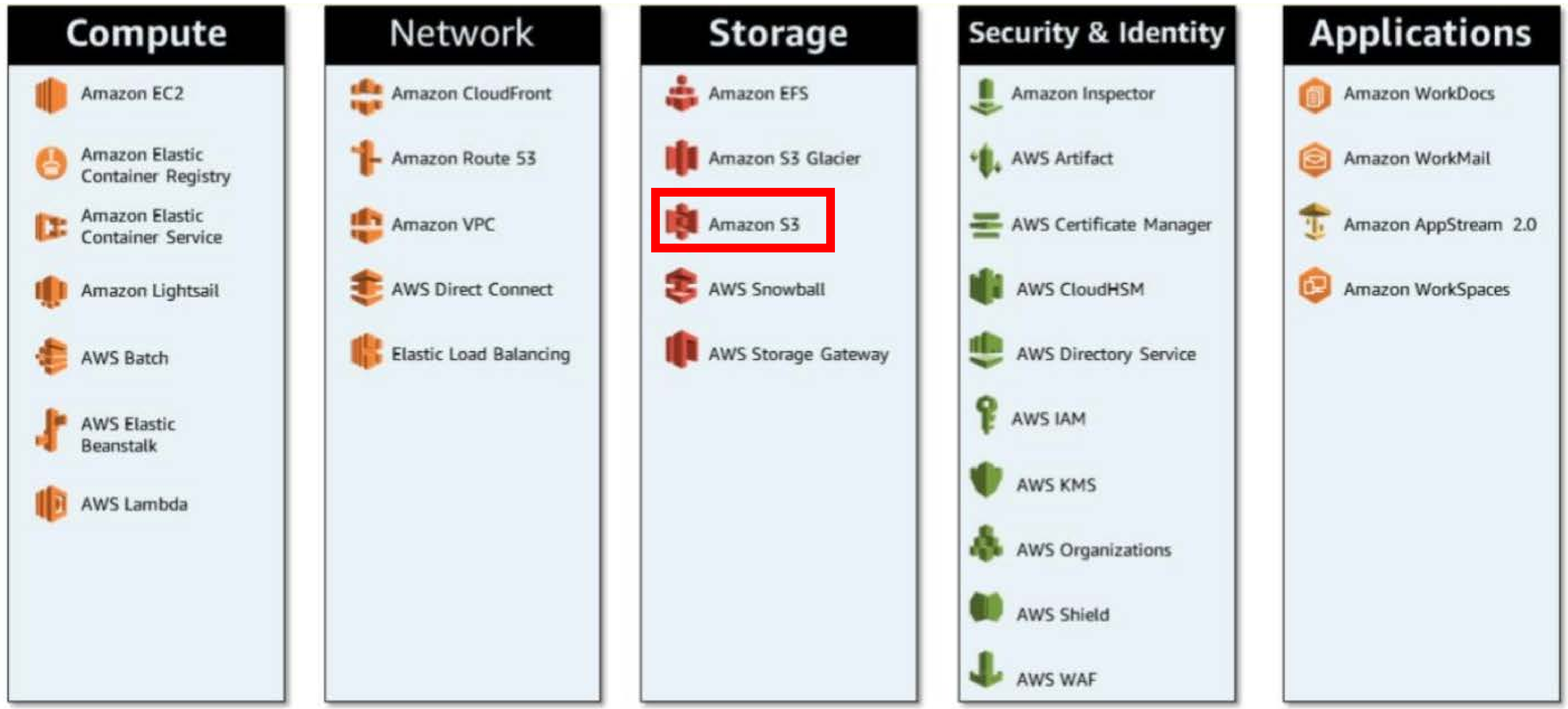
The `spark.ml` implementation of logistic regression also supports extracting a summary of the model over the training set. Note that the predictions and metrics which are stored as `DataFrame` in `LogisticRegressionSummary` are annotated `@transient` and hence only available on the driver.

Preguntas?

Agenda

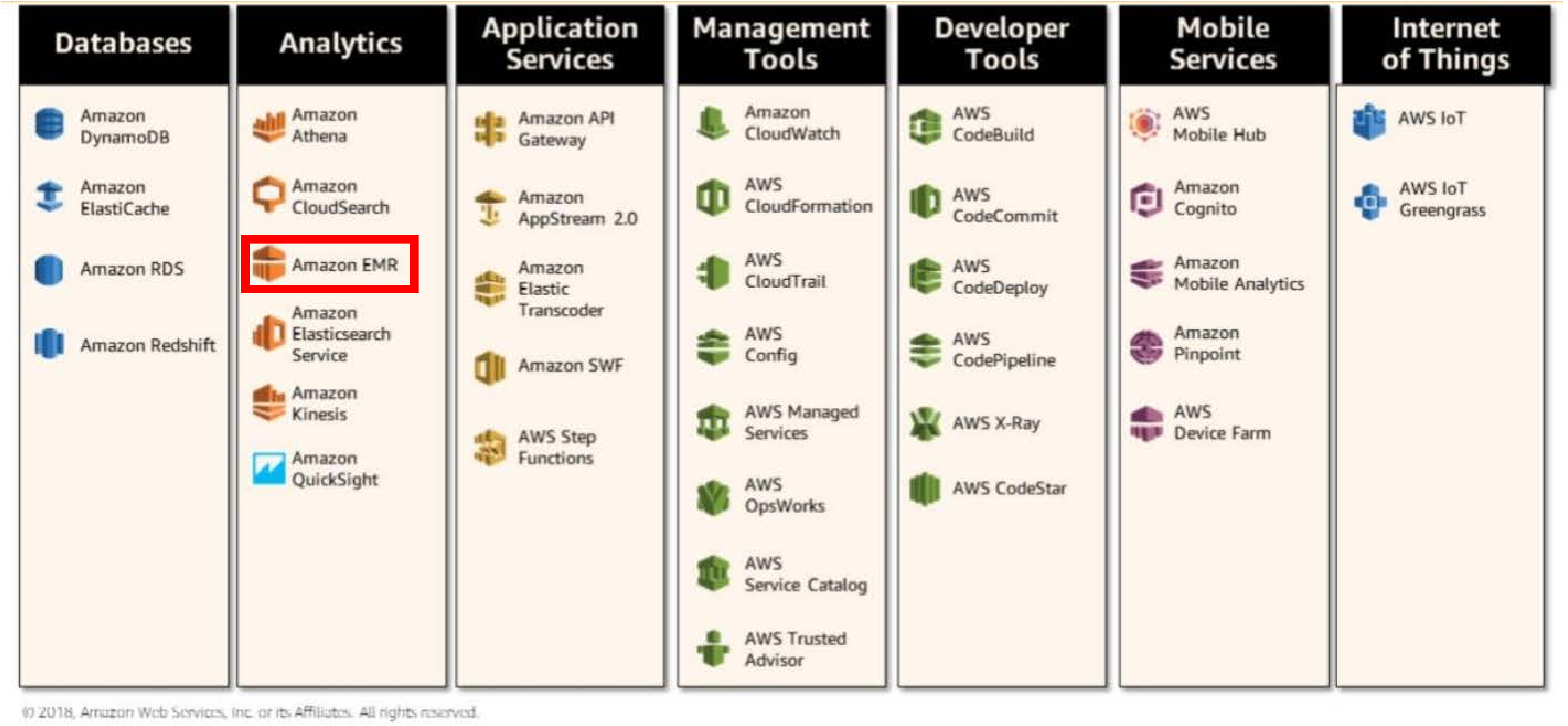
- Apache Hadoop
- Apache Spark
- Distribuciones Comerciales de Hadoop
- AWS Elastic Map Reduce (EMR)
- PySpark
- Spark SQL
- Spark MLlib
- Como abrir una cuenta de AWS gratis?

AWS Foundational Services



Compute	Network	Storage	Security & Identity	Applications
Amazon EC2	Amazon CloudFront	Amazon EFS	Amazon Inspector	Amazon WorkDocs
Amazon Elastic Container Registry	Amazon Route 53	Amazon S3 Glacier	AWS Artifact	Amazon WorkMail
Amazon Elastic Container Service	Amazon VPC	Amazon S3	AWS Certificate Manager	Amazon AppStream 2.0
Amazon Lightsail	AWS Direct Connect	AWS Snowball	AWS CloudHSM	Amazon WorkSpaces
AWS Batch	Elastic Load Balancing	AWS Storage Gateway	AWS Directory Service	
AWS Elastic Beanstalk			AWS IAM	
AWS Lambda			AWS KMS	
			AWS Organizations	
			AWS Shield	
			AWS WAF	

AWS Platform Services



Databases	Analytics	Application Services	Management Tools	Developer Tools	Mobile Services	Internet of Things
Amazon DynamoDB	Amazon Athena	Amazon API Gateway	Amazon CloudWatch	AWS CodeBuild	AWS Mobile Hub	AWS IoT
Amazon ElastiCache	Amazon CloudSearch	Amazon AppStream 2.0	AWS CloudFormation	AWS CodeCommit	Amazon Cognito	AWS IoT Greengrass
Amazon RDS	Amazon EMR	Amazon Elastic Transcoder	AWS CloudTrail	AWS CodeDeploy	Amazon Mobile Analytics	
Amazon Redshift	Amazon Elasticsearch Service	Amazon SWF	AWS Config	AWS CodePipeline	Amazon Pinpoint	
	Amazon Kinesis	AWS Step Functions	AWS Managed Services	AWS X-Ray	AWS Device Farm	
	Amazon QuickSight		AWS OpsWorks	AWS CodeStar		
			AWS Service Catalog			
			AWS Trusted Advisor			

Como abrir una cuenta de AWS gratis?

- AWS Free Tier Account → pide tarjeta de credito
- AWS Educate Account → si eres estudiante, puedes accede sin tarjeta de credito

Preguntas?