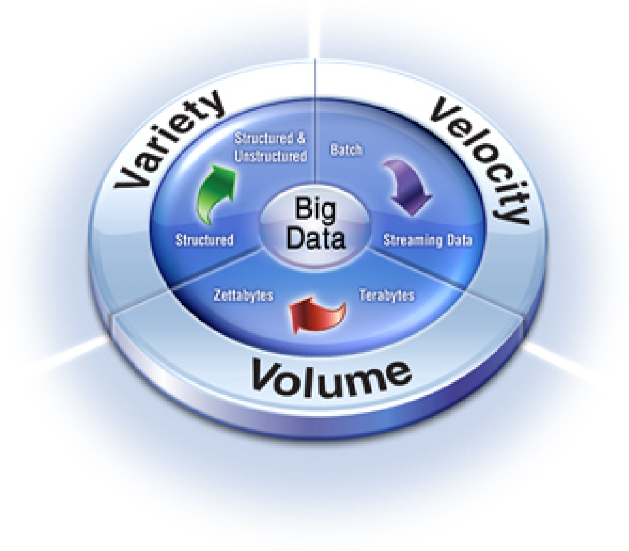
Big Data and Hadoop on IBM Cloud



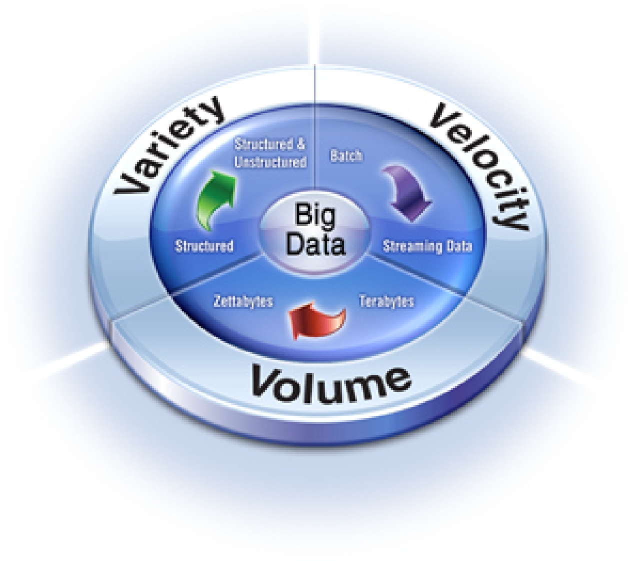
Overview

The speed and forms in which Data is generated makes it difficult to manage and process through traditional DataBases. Here comes Big Data technologies like Hadoop. Big Data is important for predictions, analysis and to get better decision making.

Step-by-step

1. Big data presents big opportunities

Extract insight from a high volume, variety and velocity of data in a timely and cost-effective manner



Variety: Manage and benefit from diverse data types and data structures

Velocity: Analyze streaming data and large volumes of persistent data

Volume: Scale from terabytes to zettabytes

Traditional Approach (BI) vs Big Data Approach

- **Traditional Approach (BI):** [Structured & Repeatable Analysis] Business Users Determine what question to ask then IT Structures the data to answer that question
- **Big Data Approach:** [Iterative & Exploratory Analysis] IT Delivers a platform to enable creative discovery then Business Explores what questions could be asked

What is Hadoop?

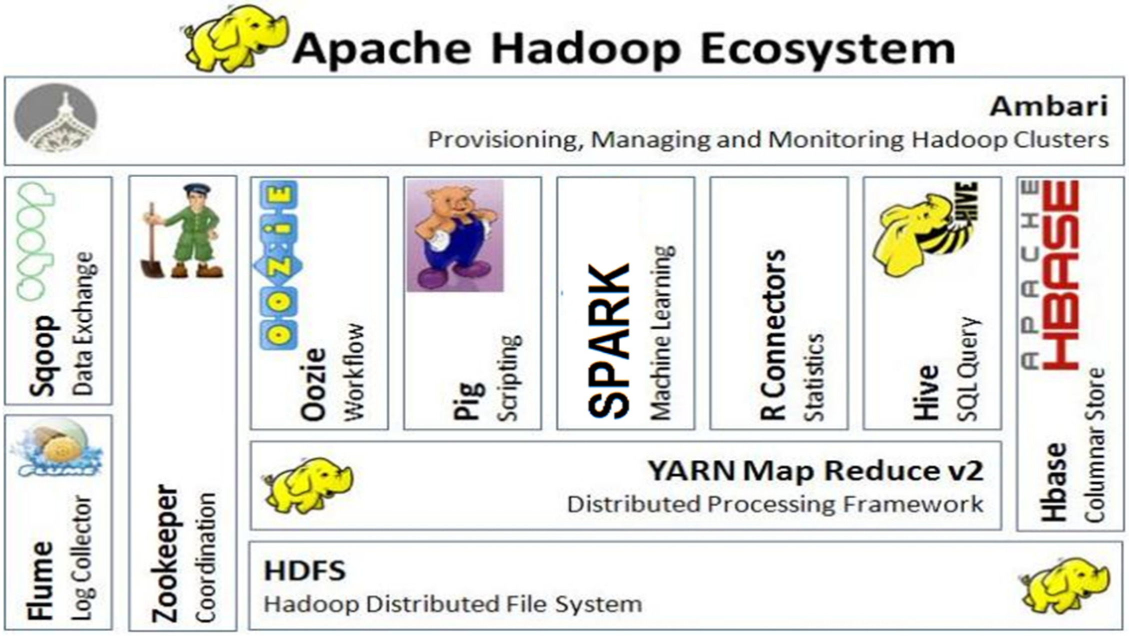
Hadoop is an open source distributed processing framework that manages data processing and storage for big data applications running in clustered systems. It is at the center of a growing ecosystem of big data technologies that are primarily used to support advanced analytics initiatives, including predictive analytics, data mining and machine learning applications. Hadoop can handle various forms of structured and unstructured data, giving users more flexibility for collecting, processing and analyzing data than relational databases and data warehouses provide.

Apache Hadoop is developed as part of an open source project.
Commercial distributions of Hadoop are currently offered by four primary vendors of big data platforms:
– Cloudera

– Hortonworks
– Amazon Web Services (AWS)
– MapR Technologies.

In addition, IBM, Google, Microsoft and other vendors offer cloud-based managed services that are built on top of Hadoop

IBM, Microsoft's Azure HDInsight, and Pivotal (a Dell Technologies subsidiary) are based on the Hortonworks platform.
while Intel use Cloudera.



The 4 Modules of Hadoop

1. Distributed File-System HDFS: allows data to be stored in an easily accessible format, across a large number of linked storage devices

2. MapReduce: MapReduce do two basic operations – reading data from the database, putting it into a format suitable for analysis (map), and performing mathematical operations i.e counting the number of males aged 30+ in a customer database (reduce).

3. Hadoop Common: provides the tools (in Java) needed for the user's computer systems (Windows, Unix or whatever) to read data stored under the Hadoop file system.

4. YARN: manages resources of the systems storing the data and running the analysis.

Advantages and disadvantages of Hadoop

Hadoop is good for:

- processing massive amounts of data through parallelism
- handling a variety of data (structured, unstructured, semi-structured)
- using inexpensive commodity hardware

Hadoop is not good for:

- processing transactions (random access)
- when work cannot be parallelized
- Fast access to data
- processing lots of small files
- intensive calculations with small amounts of data

What hardware is not used for Hadoop?

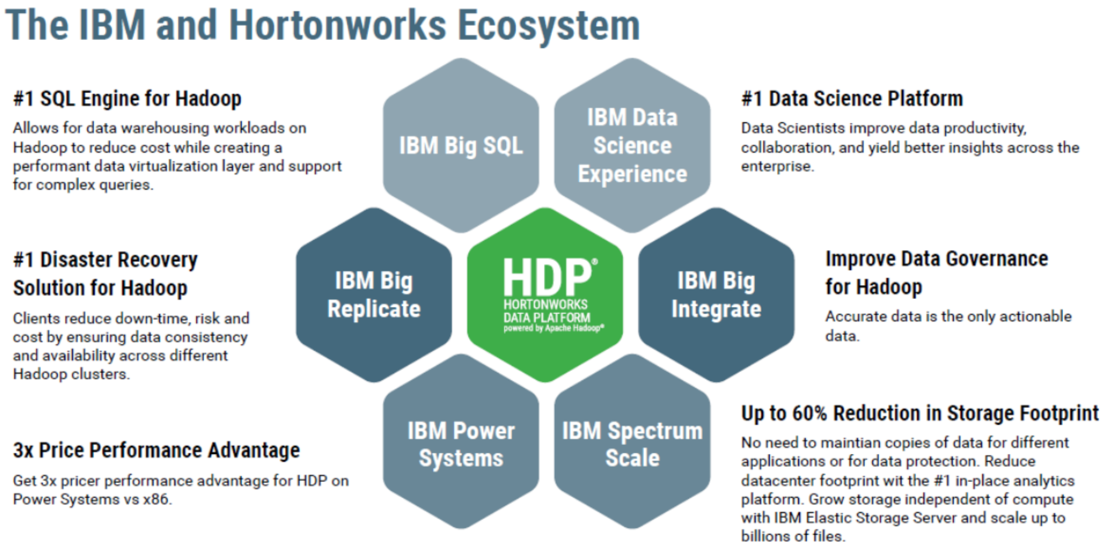
- RAID
- Linux Logical Volume Manager (LVM)
- Solid-state disk (SSD)

Big data tools associated with Hadoop

Apache Flume: a tool used to collect, aggregate and move huge amounts of streaming data into HDFS
Apache HBase: a distributed database that is often paired with Hadoop
Apache Hive: an SQL-on-Hadoop tool that provides data summarization, query and analysis
Apache Oozie: a server-based workflow scheduling system to manage Hadoop jobs
Apache Phoenix: an SQL-based massively parallel processing (MPP) database engine that uses HBase as its data store
Apache Pig: a high-level platform for creating programs that run on Hadoop clusters
Apache Sqoop: a tool to help transfer bulk data between Hadoop and structured data stores, such as relational databases
Apache ZooKeeper: a configuration, synchronization and naming registry service for large distributed systems.
apache Solr: enterprise search engine
Apache Spark: Spark is an alternative in-memory framework to MapReduce, Supports streaming, interactive queries and machine learning.
Kafka: distributed publish-subscribe messaging system and a robust queue that can handle a high volume of data.

GUI tools to manage Hadoop

- **Ambari:** developed by HortonWorks.
- **HUE:** developed by Cloudera



The IBM and Hortonworks Ecosystem

#1 SQL Engine for Hadoop

Allows for data warehousing workloads on Hadoop to reduce cost while creating a performant data virtualization layer and support for complex queries.

#1 Disaster Recovery Solution for Hadoop

Clients reduce down-time, risk and cost by ensuring data consistency and availability across different Hadoop clusters.

3x Price Performance Advantage

Get 3x pricer performance advantage for HDP on Power Systems vs x86.

#1 Data Science Platform

Data Scientists improve data productivity, collaboration, and yield better insights across the enterprise.

Improve Data Governance for Hadoop

Accurate data is the only actionable data.

Up to 60% Reduction in Storage Footprint

No need to maintian copies of data for different applications or for data protection. Reduce datacenter footprint wit the #1 in-place analytics platform. Grow storage independent of compute with IBM Elastic Storage Server and scale up to billions of files.

IBM Big SQL
IBM Data Science Experience
IBM Big Replicate
HDP® HORTONWORKS DATA PLATFORM powered by Apache Hadoop®
IBM Big Integrate
IBM Power Systems
IBM Spectrum Scale

What is IBM Big SQL?

– Industry-standard SQL query interface for BigInsights data

– New Hadoop query engine derived from decades of IBM R&D investment in RDBMS technology, including database parallelism and query optimization

Why Big SQL?

– Easy on-ramp to Hadoop for SQL professionals

– Support familiar SQL tools / applications (via JDBC and ODBC drivers)

What operations are supported

– Create tables / views. Store data in DFS, HBase, or Hive warehouse

– Load data into tables (from local files, remote files, RDBMSs(

– Query data (project, restrict, join, union, wide range of sub-queries, and built-in functions, UDFs, etc.(

– GRANT / REVOKE privileges, create roles, create column masks and row permissions

– Transparently join / union data between Hadoop and RDBMSs in single query

– Collect statistics and inspect detailed data access plan

– Establish workload management controls

– Monitor Big SQL usage

IBM Big SQL Main Features

1- Comprehensive, standard SQL

– SELECT: joins, unions, aggregates, subqueries

– GRANT/REVOKE, INSERT … INTO

– PL/SQL

– Stored procs, user-defined functions

– IBM data server JDBC and ODBC drivers

2- Optimization and performance

– IBM MPP engine (C++) replaces Java MapReduce layer

– Continuous running daemons (no start up latency)

– Message passing allow data to flow between nodes without persisting intermediate results

– In-memory operations with ability to spill to disk (useful for aggregations that exceed available RAM)

– Cost-based query optimization with 140+ rewrite rules

3- Various storage formats supported

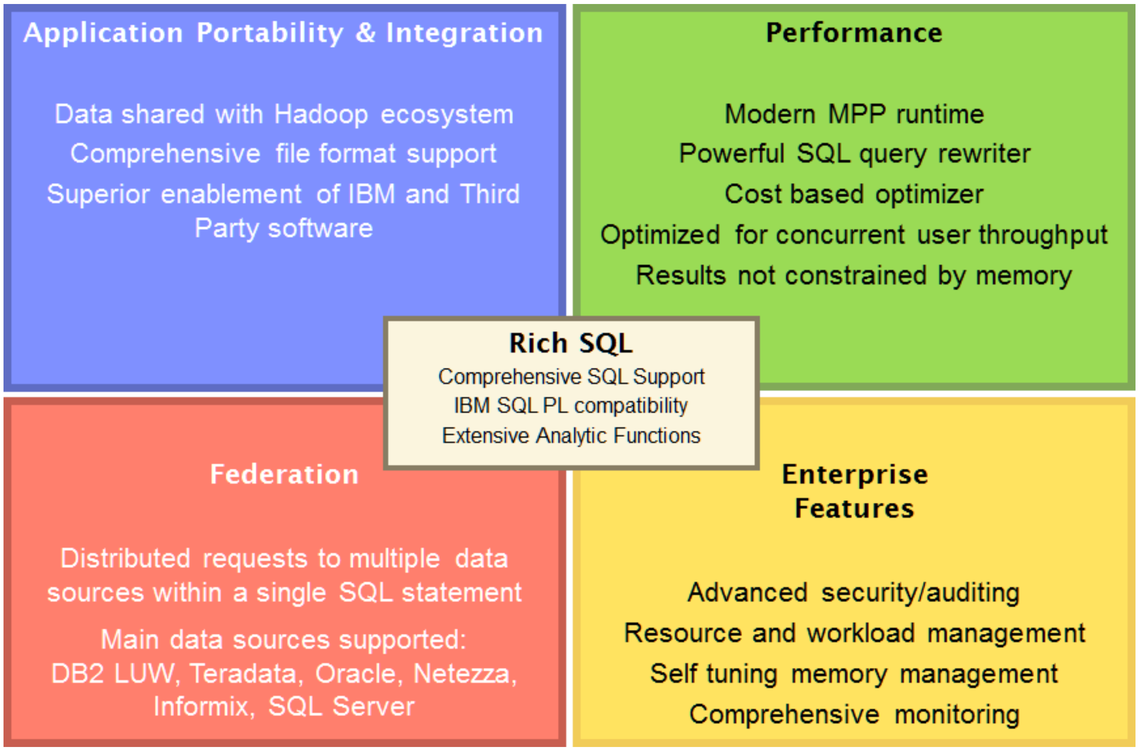
– Text (delimited), Sequence, RCFile, ORC, Avro, Parquet

– Data persisted in DFS, Hive, HBase

– No IBM proprietary format required

4- Integration with RDBMSs via LOAD, query federation

Why choose Big SQL instead of Hive and other vendors?



Application Portability & Integration

Data shared with Hadoop ecosystem

Comprehensive file format support

Superior enablement of IBM and Third Party software

Performance

Modern MPP runtime

Powerful SQL query rewriter

Cost based optimizer

Optimized for concurrent user throughput

Results not constrained by memory

Rich SQL

Comprehensive SQL Support

IBM SQL PL compatibility

Extensive Analytic Functions

Federation

Distributed requests to multiple data sources within a single SQL statement

Main data sources supported: DB2 LUW, Teradata, Oracle, Netezza, Informix, SQL Server

Enterprise Features

Advanced security/auditing

Resource and workload management

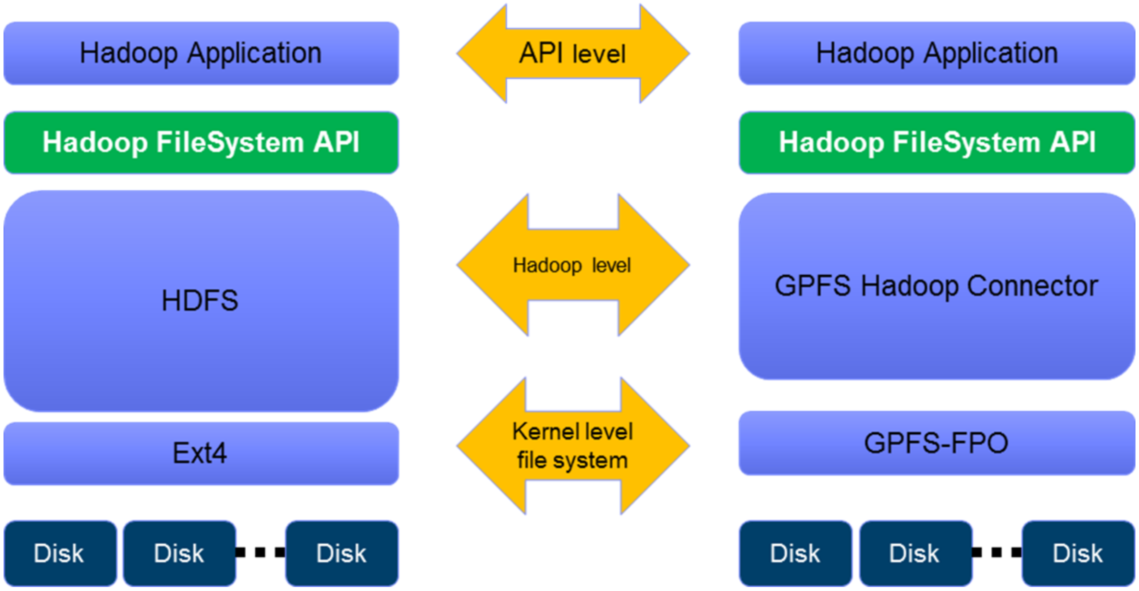
Self tuning memory management

Comprehensive monitoring

IBM Spectrum Scale (Also know as “GPFS FPO”)

What is expected from IBM Spectrum Scale ?

- high scale, high performance, high availability, data integrity
- same data accessible from different computers
- logical isolation: filesets are separate filesystems inside a filesystem
- physical isolation: filesets can be put in separate storage pools
- enterprise features (quotas, security, ACLs, snapshots, etc.)



HDFS vs GPFS Commands

1) Copy File

– **HDFS:** hadoop fs -copyFromLocal /local/source/path /hdfs/target/path

– **GPFS:** cp /source/path /target/path

2) Move File

– **HDFS:** hadoop fs -mv path1/ path2/

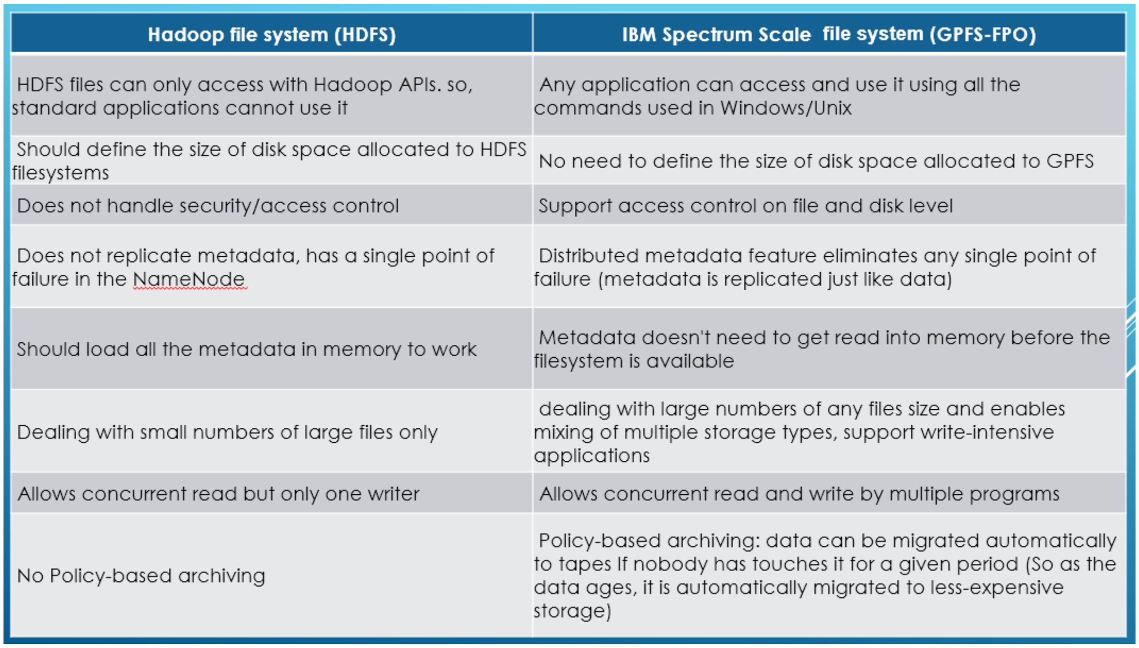
– **GPFS:** mv path1/ path2/

3) Compare Files

– **HDFS:** diff < (hadoop fs -cat file1) < (hadoop fs -cat file2)

– **GPFS:** diff file1 file2

HDFS vs IBM Spectrum Scale

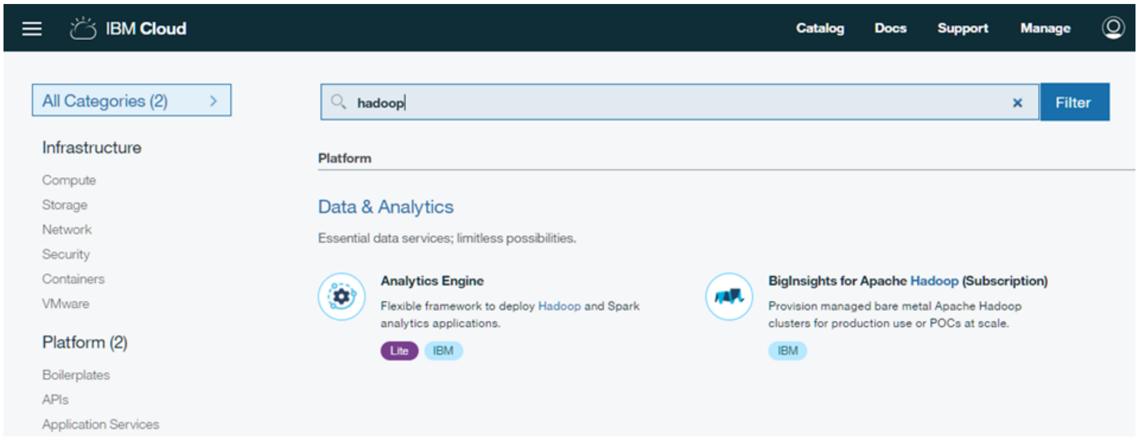


Hadoop file system (HDFS)	IBM Spectrum Scale file system (GPFS-FPO)
HDFS files can only access with Hadoop APIs. so, standard applications cannot use it	Any application can access and use it using all the commands used in Windows/Unix
Should define the size of disk space allocated to HDFS filesystems	No need to define the size of disk space allocated to GPFS
Does not handle security/access control	Support access control on file and disk level
Does not replicate metadata, has a single point of failure in the NameNode	Distributed metadata feature eliminates any single point of failure (metadata is replicated just like data)
Should load all the metadata in memory to work	Metadata doesn't need to get read into memory before the filesystem is available
Dealing with small numbers of large files only	dealing with large numbers of any files size and enables mixing of multiple storage types, support write-intensive applications
Allows concurrent read but only one writer	Allows concurrent read and write by multiple programs
No Policy-based archiving	Policy-based archiving: data can be migrated automatically to tapes If nobody has touches it for a given period (So as the data ages, it is automatically migrated to less-expensive storage)

2. How to work with Hadoop on IBM cloud?

Login to IBM Cloud (BlueMix) https://console.bluemix.net/ , and search for Hadoop

You will get two results (Lite = Free), and (Subscription=with Cost) –> choose “Analytics Engine”



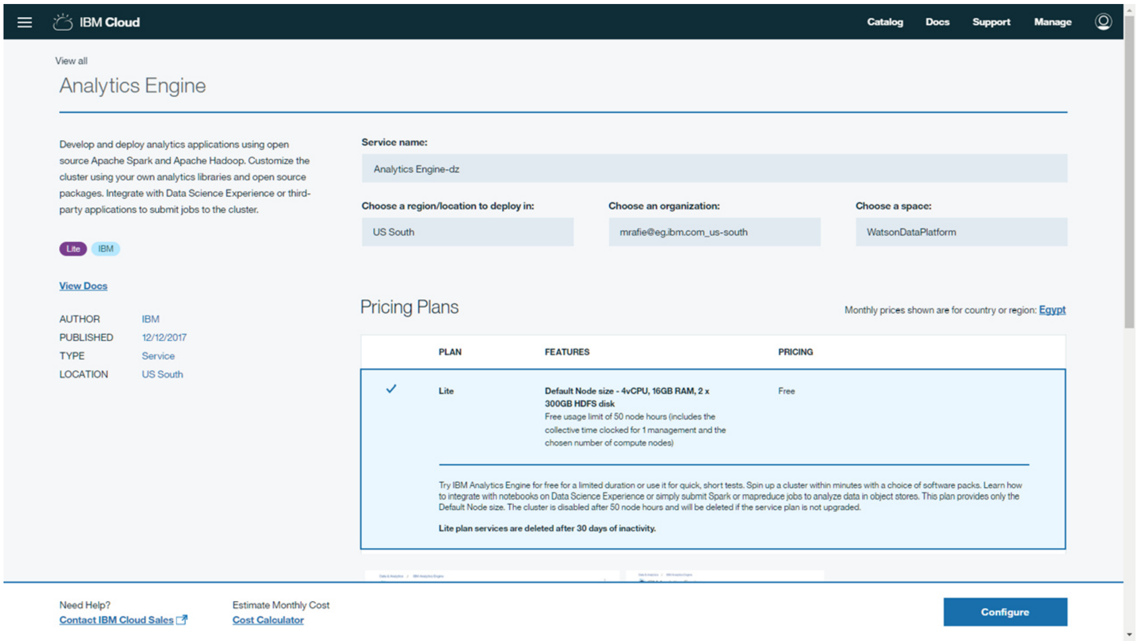
Essential data services; limitless possibilities.

Analytics Engine

Flexible framework to deploy Hadoop and Spark analytics applications.

BigInsights for Apache Hadoop (Subscription)

Provision managed bare metal Apache Hadoop clusters for production use or POCs at scale.



Analytics Engine

Develop and deploy analytics applications using open source Apache Spark and Apache Hadoop. Customize the cluster using your own analytics libraries and open source packages. Integrate with Data Science Experience or third-party applications to submit jobs to the cluster.

View Docs

AUTHOR	IBM
PUBLISHED	12/12/2017
TYPE	Service
LOCATION	US South

Service name: Analytics Engine-dz

Choose a region/location to deploy in: US South

Choose an organization: mrafie@eg.ibm.com_us-south

Choose a space: WatsonDataPlatform

Pricing Plans

Monthly prices shown are for country or region: Egypt

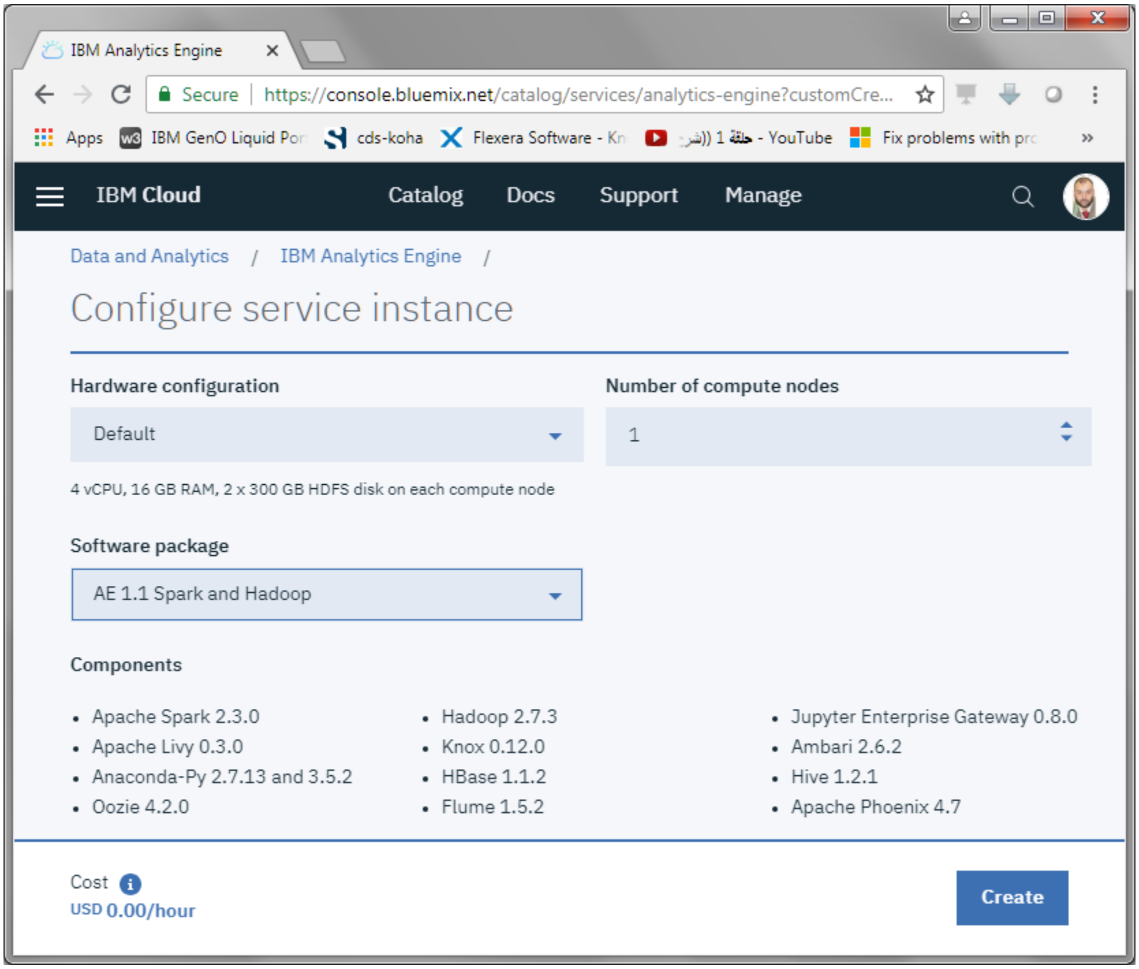
	PLAN	FEATURES	PRICING
✓	Lite	**Default Node size - 4vCPU, 16GB RAM, 2 x 300GB HDFS disk** Free usage limit of 50 node hours (includes the collective time clocked for 1 management and the chosen number of compute nodes)	Free

Try IBM Analytics Engine for free for a limited duration or use it for quick, short tests. Spin up a cluster within minutes with a choice of software packs. Learn how to integrate with notebooks on Data Science Experience or simply submit Spark or mapreduce jobs to analyze data in object stores. This plan provides only the Default Node size. The cluster is disabled after 50 node hours and will be deleted if the service plan is not upgraded.

Lite plan services are deleted after 30 days of inactivity.

Need Help?
Contact IBM Cloud Sales

Estimate Monthly Cost
Cost Calculator



Data and Analytics / IBM Analytics Engine /

Configure service instance

Hardware configuration: Default

4 vCPU, 16 GB RAM, 2 x 300 GB HDFS disk on each compute node

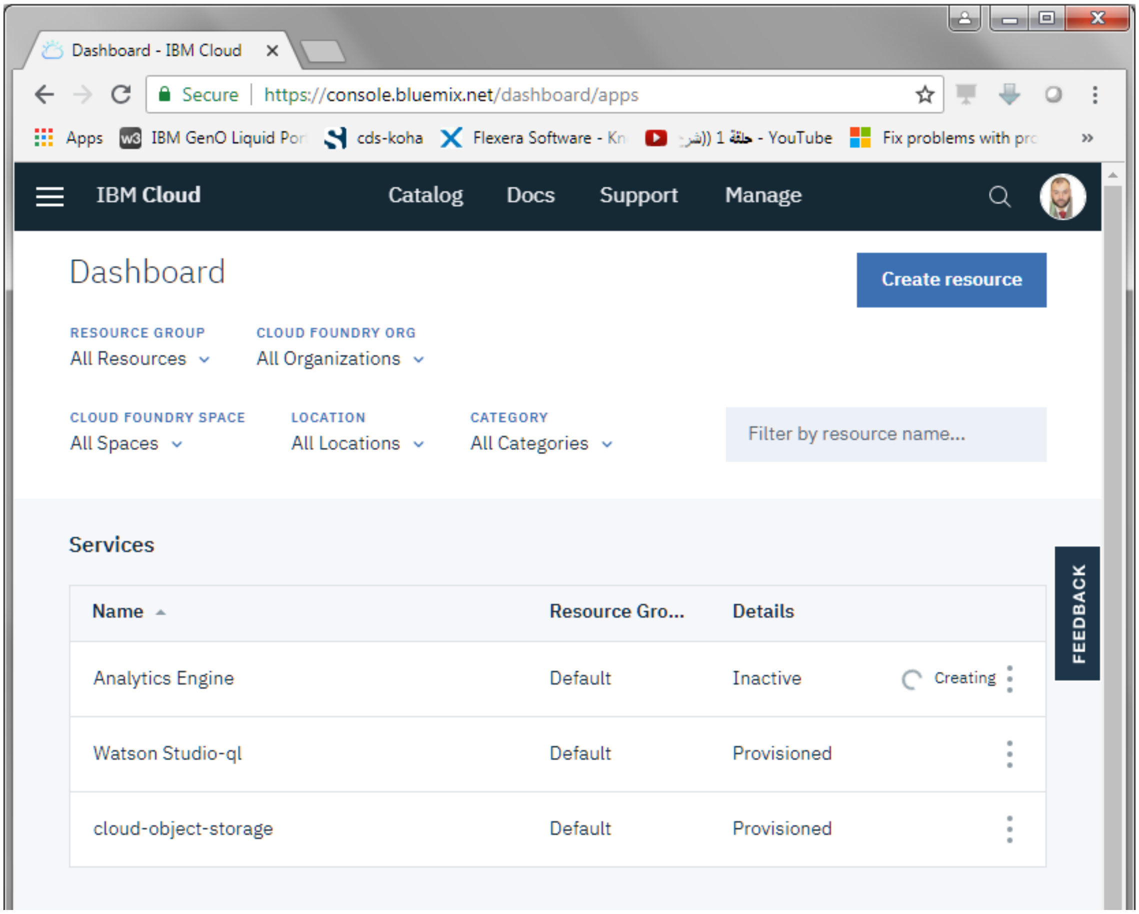
Number of compute nodes: 1

Software package: AE 1.1 Spark and Hadoop

Components

- Apache Spark 2.3.0
- Apache Livy 0.3.0
- Anaconda-Py 2.7.13 and 3.5.2
- Oozie 4.2.0
- Hadoop 2.7.3
- Knox 0.12.0
- HBase 1.1.2
- Flume 1.5.2
- Jupyter Enterprise Gateway 0.8.0
- Ambari 2.6.2
- Hive 1.2.1
- Apache Phoenix 4.7

Cost
USD 0.00/hour



Dashboard

RESOURCE GROUP: All Resources

CLOUD FOUNDRY ORG: All Organizations

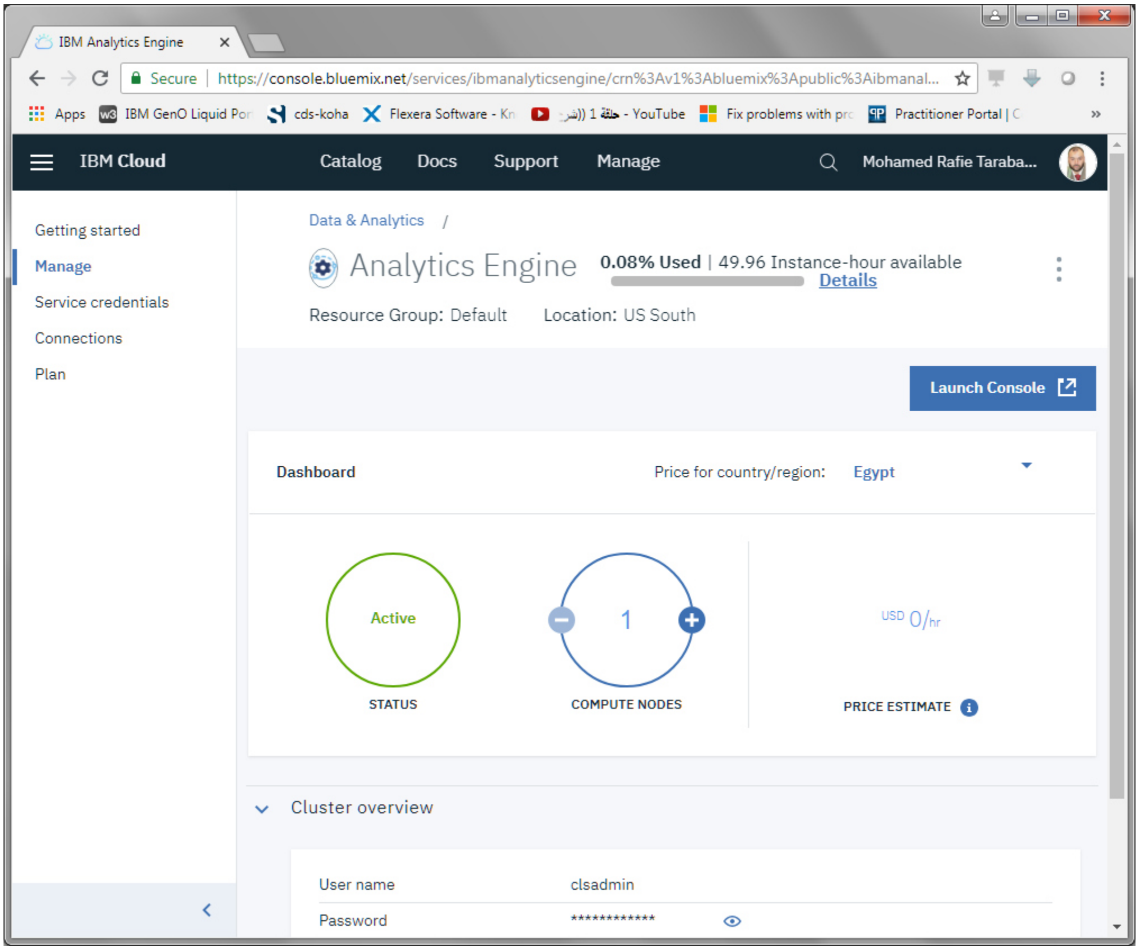
CLOUD FOUNDRY SPACE: All Spaces

LOCATION: All Locations

CATEGORY: All Categories

Services

Name	Resource Gro...	Details	
Analytics Engine	Default	Inactive	Creating
Watson Studio-ql	Default	Provisioned	
cloud-object-storage	Default	Provisioned	



0.08% Used | 49.96 Instance-hour available

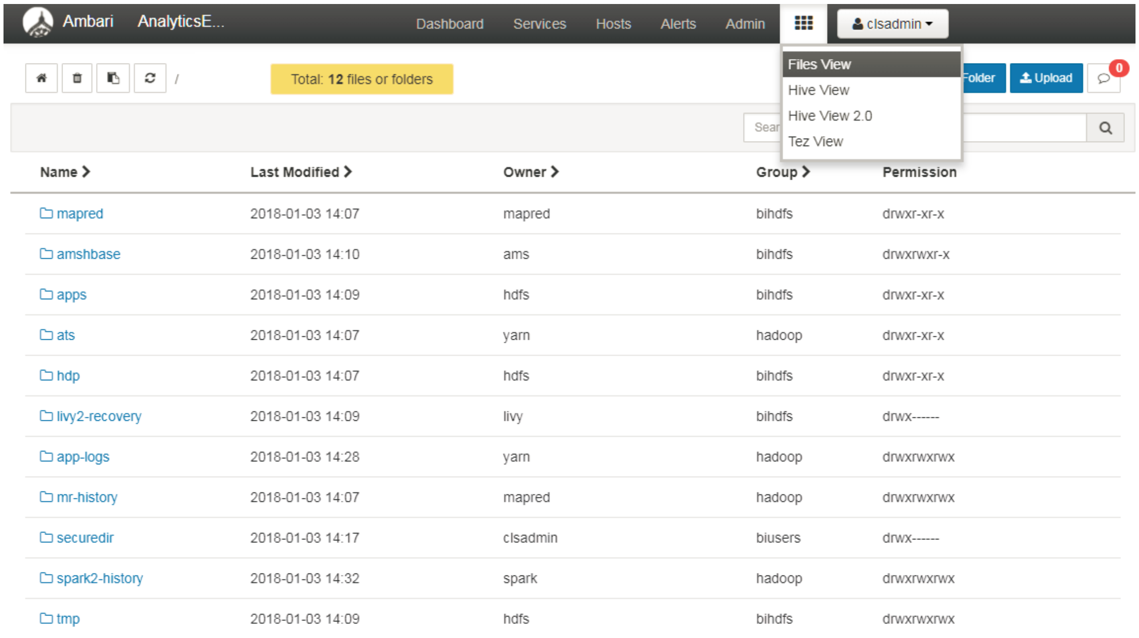
Resource Group: Default   Location: US South

Price for country/region: Egypt

User name	clsadmin
Password	************

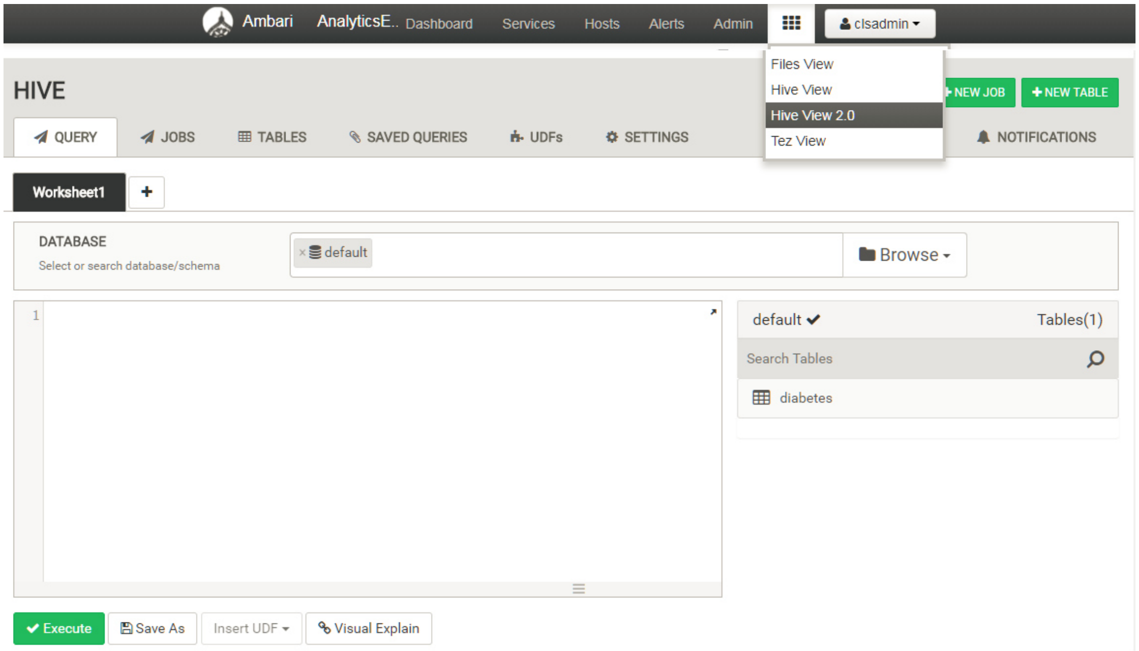
Ambari (GUI tools to manage Hadoop)

Ambari View is developed by HortonWorks.
Ambari is a GUI tool you can use to create(install) manage the entire hadoop cluster. You can keep on expanding by adding nodes and monitor the health, space utilization etc through Ambari.
Ambari views are more to help users to use the installed components/services like hive, pig, capacity scheduler to see the cluster-load and manage YARN workload management, provisioning cluster resources, manage files etc.



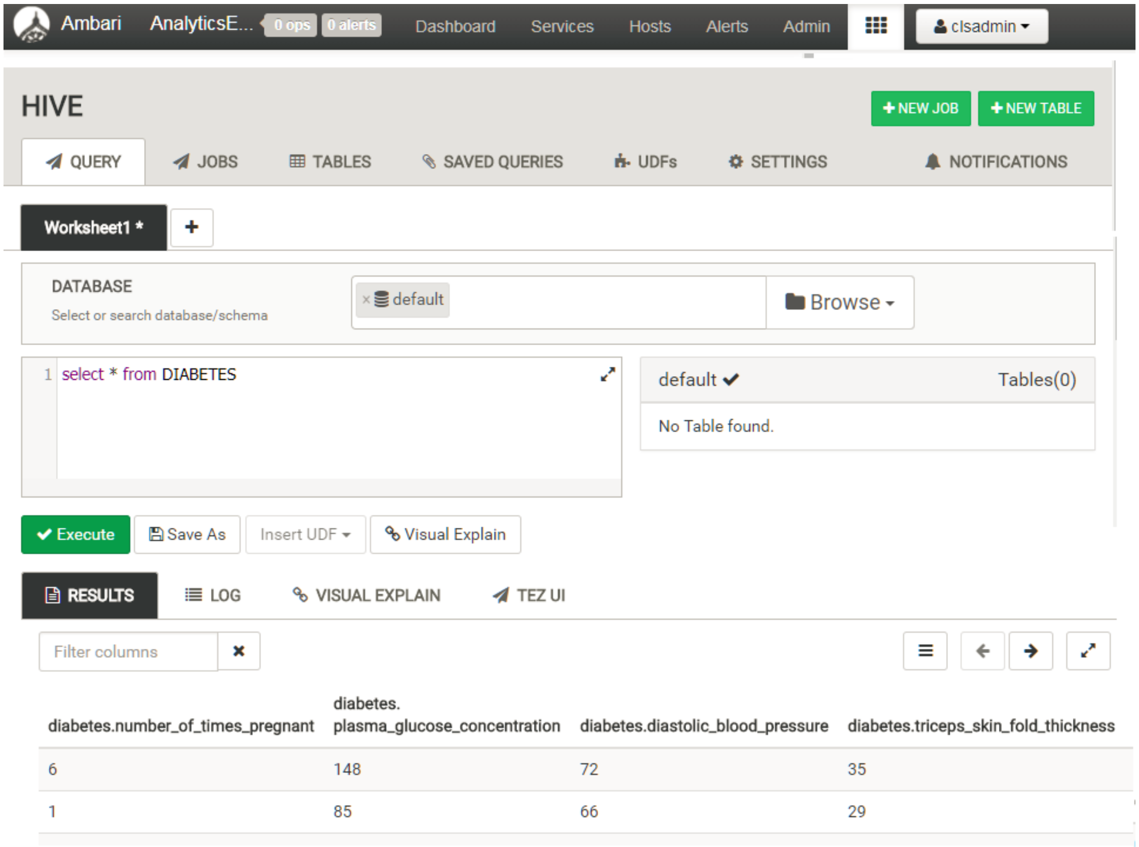
Total: 12 files or folders

Name	Last Modified	Owner	Group	Permission
mapred	2018-01-03 14:07	mapred	bihdfs	drwxr-xr-x
amshbase	2018-01-03 14:10	ams	bihdfs	drwxrwxr-x
apps	2018-01-03 14:09	hdfs	bihdfs	drwxr-xr-x
ats	2018-01-03 14:07	yarn	hadoop	drwxr-xr-x
hdp	2018-01-03 14:07	hdfs	bihdfs	drwxr-xr-x
livy2-recovery	2018-01-03 14:09	livy	bihdfs	drwx------
app-logs	2018-01-03 14:28	yarn	hadoop	drwxrwxrwx
mr-history	2018-01-03 14:07	mapred	hadoop	drwxrwxrwx
securedir	2018-01-03 14:17	clsadmin	biusers	drwx------
spark2-history	2018-01-03 14:32	spark	hadoop	drwxrwxrwx
tmp	2018-01-03 14:09	hdfs	bihdfs	drwxrwxrwx



DATABASE
Select or search database/schema
default

default ✓ Tables(1)
Search Tables
diabetes



```
select * from DIABETES
```

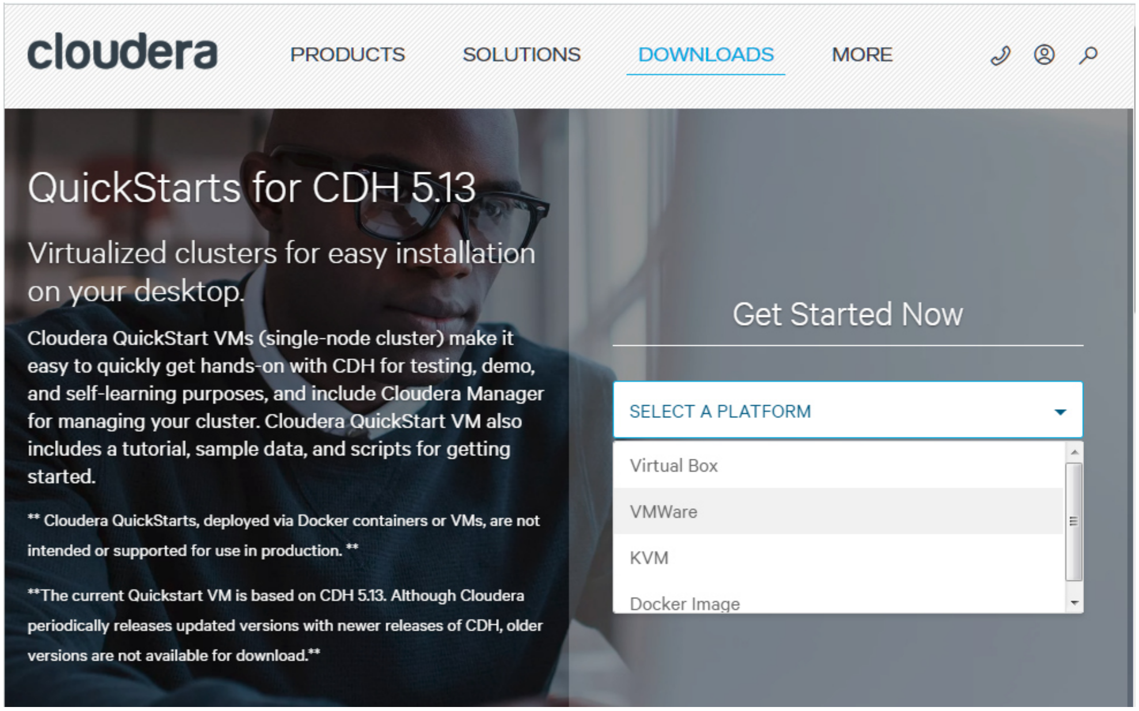
default ✓ Tables(0)

No Table found.

diabetes.number_of_times_pregnant	diabetes. plasma_glucose_concentration	diabetes.diastolic_blood_pressure	diabetes.triceps_skin_fold_thickness
6	148	72	35
1	85	66	29

3. How to work with Hadoop using Cloudera VM?

Download VM from https://www.cloudera.com/downloads/quickstart_vms/5-13.html



QuickStarts for CDH 5.13

Virtualized clusters for easy installation on your desktop.

Cloudera QuickStart VMs (single-node cluster) make it easy to quickly get hands-on with CDH for testing, demo, and self-learning purposes, and include Cloudera Manager for managing your cluster. Cloudera QuickStart VM also includes a tutorial, sample data, and scripts for getting started.

** Cloudera QuickStarts, deployed via Docker containers or VMs, are not intended or supported for use in production. **

The current Quickstart VM is based on CDH 5.13. Although Cloudera periodically releases updated versions with newer releases of CDH, older versions are not available for download.

How to use HUE

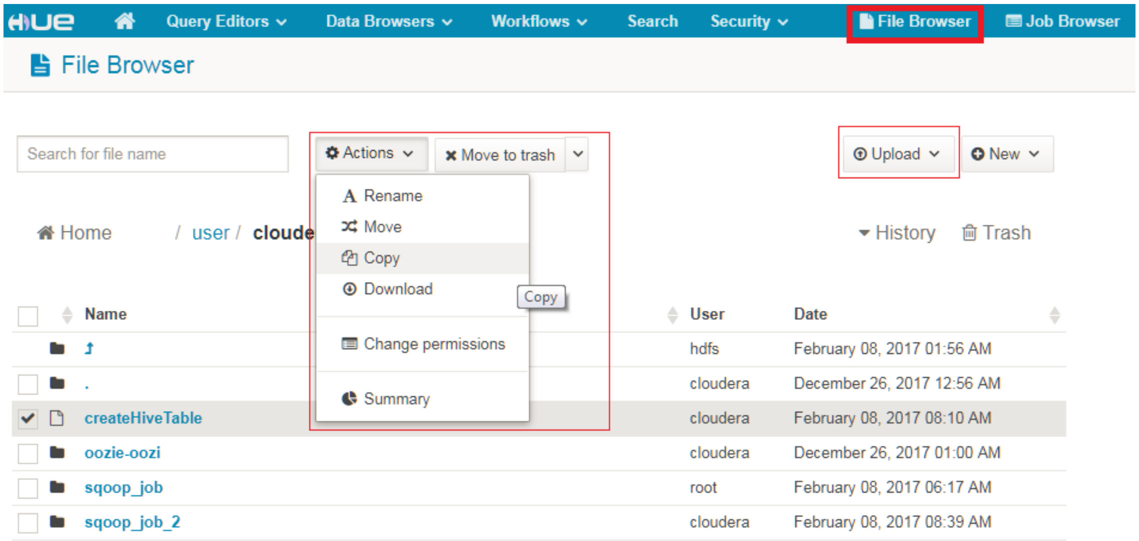
Example 1

Copy file from HD to HDFS

Using command line

hadoop fs -put /HD PATH/temperature.csv /Hadoop Path/temp

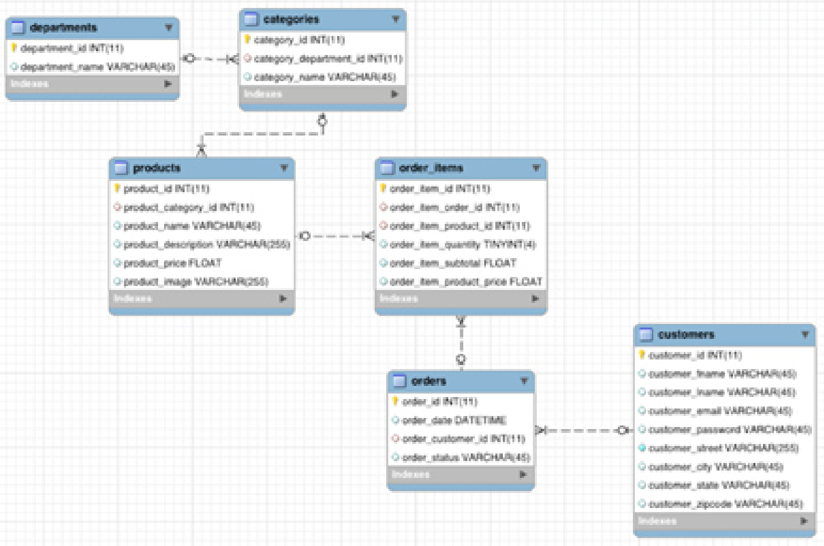
Using HUE GUI



	Name	User	Date
	↥	hdfs	February 08, 2017 01:56 AM
□	.	cloudera	December 26, 2017 12:56 AM
☑	createHiveTable	cloudera	February 08, 2017 08:10 AM
□	oozie-oozi	cloudera	December 26, 2017 01:00 AM
□	sqoop_job	root	February 08, 2017 06:17 AM
□	sqoop_job_2	cloudera	February 08, 2017 08:39 AM

Example 2

Use Scoop to move mySql DB table to Hadoop file system inside hive directory


departments
department_id INT(11)
department_name VARCHAR(45)
Indexes
categories
category_id INT(11)
category_department_id INT(11)
category_name VARCHAR(45)
Indexes
products
product_id INT(11)
product_category_id INT(11)
product_name VARCHAR(45)
product_description VARCHAR(255)
product_price FLOAT
product_image VARCHAR(255)
Indexes
order_items
order_item_id INT(11)
order_item_order_id INT(11)
order_item_product_id INT(11)
order_item_quantity TINYINT(4)
order_item_subtotal FLOAT
order_item_product_price FLOAT
Indexes
orders
order_id INT(11)
order_date DATETIME
order_customer_id INT(11)
order_status VARCHAR(45)
Indexes
customers
customer_id INT(11)
customer_fname VARCHAR(45)
customer_lname VARCHAR(45)
customer_email VARCHAR(45)
customer_password VARCHAR(45)
customer_street VARCHAR(255)
customer_city VARCHAR(45)
customer_state VARCHAR(45)
customer_zipcode VARCHAR(45)
Indexes

```
> sqoop import-all-tables \
-m 1\
–connect jdbc:mysql://localhost:3306/retail_db \
–username=retail_dba \
–password=cloudera \
–compression-codec=snappy \
–as-parquetfile \
–warehouse-dir=/user/hive/warehouse \
–hive-import
```

Parameters description

-m parameter: number of .parquet files
/usr/hive/warehouse is the default hive path
To view tables after move to HDFS > hadoop fs -ls /user/hive/warehouse/
To get the actual hive Tables path, use terminally type hive then run command set hive.metastore.warehouse.dir;

Example 3

Working with Hive tables

Task 1: To view current Hive tables

```
show tables;
```

Task 2: Run SQL command on Hive tables

```
select c.category_name, count(order_item_quantity) as count
from order_items oi
inner join products p on oi.order_item_product_id = p.product_id
inner join categories c on c.category_id = p.product_category_id
group by c.category_name
order by count desc
limit 10;
```

Task 3: Run SQL command on Hive tables

```
select p.product_id, p.product_name, r.revenue from products p inner join

(

select oi.order_item_product_id, sum(cast(oi.order_item_subtotal as float)) as revenue

from order_items oi inner join orders o

on oi.order_item_order_id = o.order_id where o.order_status <> ‘CANCELED’ and o.order_status <> ‘SUSPECTED_FRAUD’

group by order_item_product_id

) r

on p.product_id = r.order_item_product_id

order by r.revenue desc

limit 10;
```

Example 4

Copy temperature.csv file from HD to new HDFS directory “temp” then load this file inside new Hive table

```
hadoop fs -mkdir -p /user/cloudera/temp

hadoop fs -put /var/www/html/temperature.csv /user/cloudera/temp
```

Create Hive table based on CVS file

```
hive> Create database weather;

CREATE EXTERNAL TABLE IF NOT EXISTS weather.temperature (

place STRING COMMENT ‘place’,

year INT COMMENT ‘Year’,

month STRING COMMENT ‘Month’,

temp FLOAT COMMENT ‘temperature’)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ‘,’
```

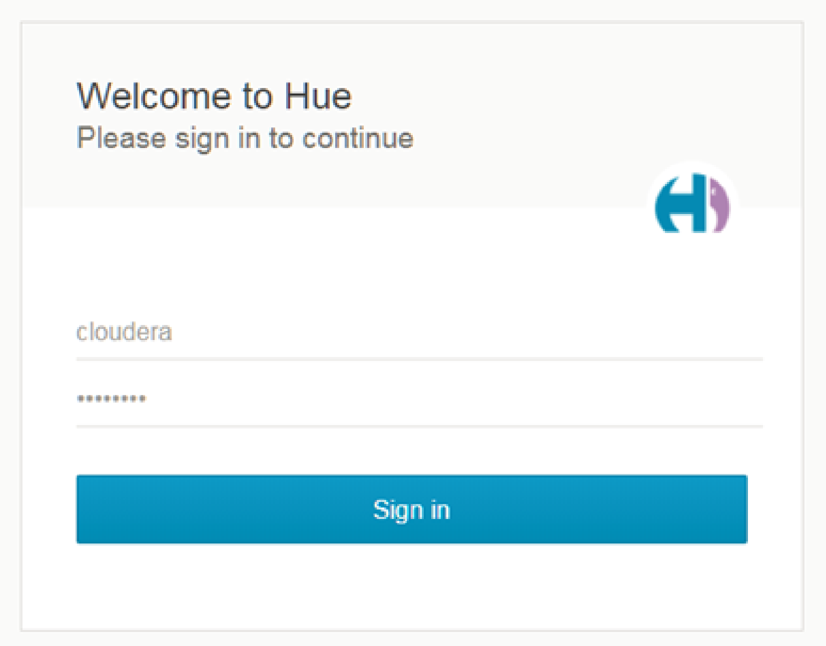
```
LINES TERMINATED BY '\n'

STORED AS TEXTFILE

LOCATION '/user/cloudera/temp/';
```

Example 5

Using HUE create a workflow using Oozie to move data from mySQL/CSV files to Hive

Step 1: get the virtual machine IP using ifconfig

Step 2: navigate to the http://IP:8888 , to get HUE login screen (cloudera/cloudera)



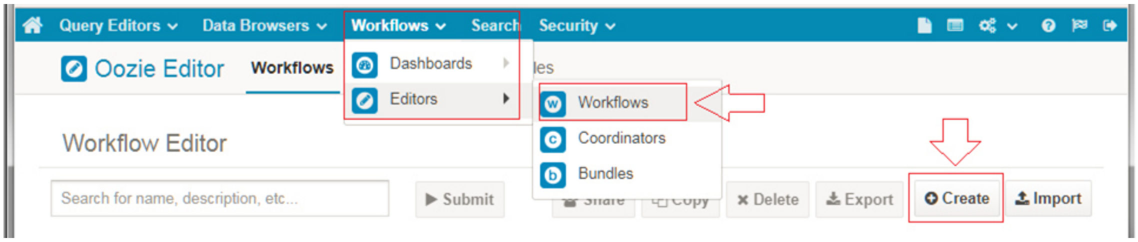
Welcome to Hue

Please sign in to continue

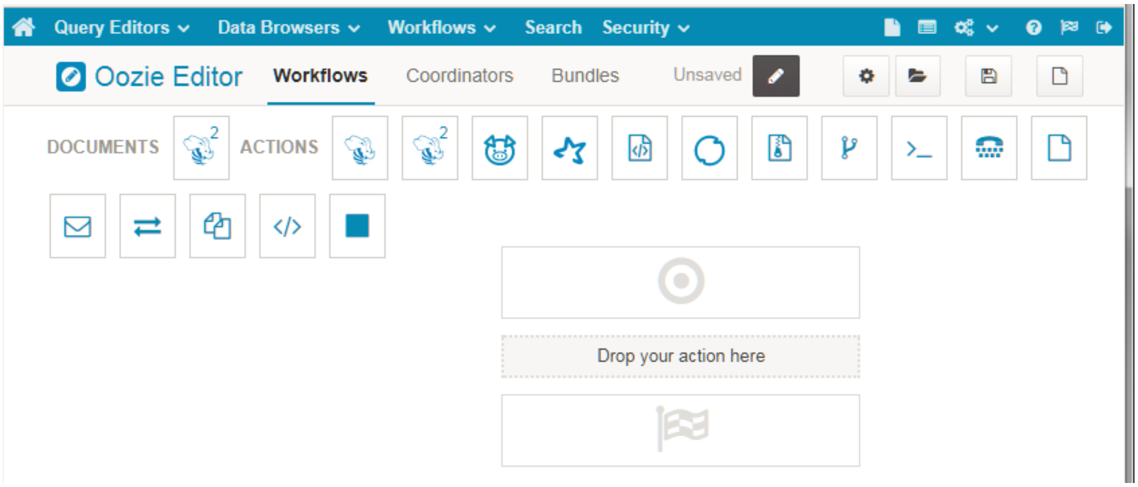
cloudera

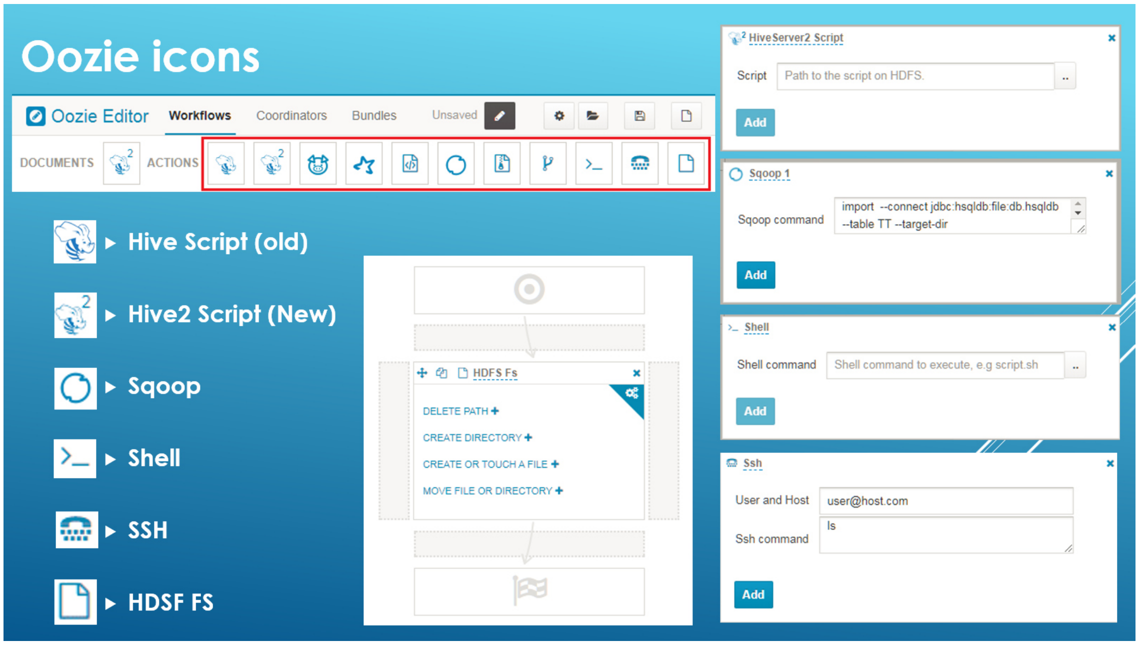
****

Step 3: Open Oozie: Workflows>Editors>Workflows> then click "create" button



then





Oozie icons

Hive Script (old)

Hive2 Script (New)

Sqoop

Shell

SSH

HDSF FS

HiveServer2 Script

Script: Path to the script on HDFS.

Add

Sqoop 1

Sqoop command: import --connect jdbc:hsqldb:file:db.hsqldb --table TT --target-dir

Add

Shell

Shell command: Shell command to execute, e.g script.sh

Add

Ssh

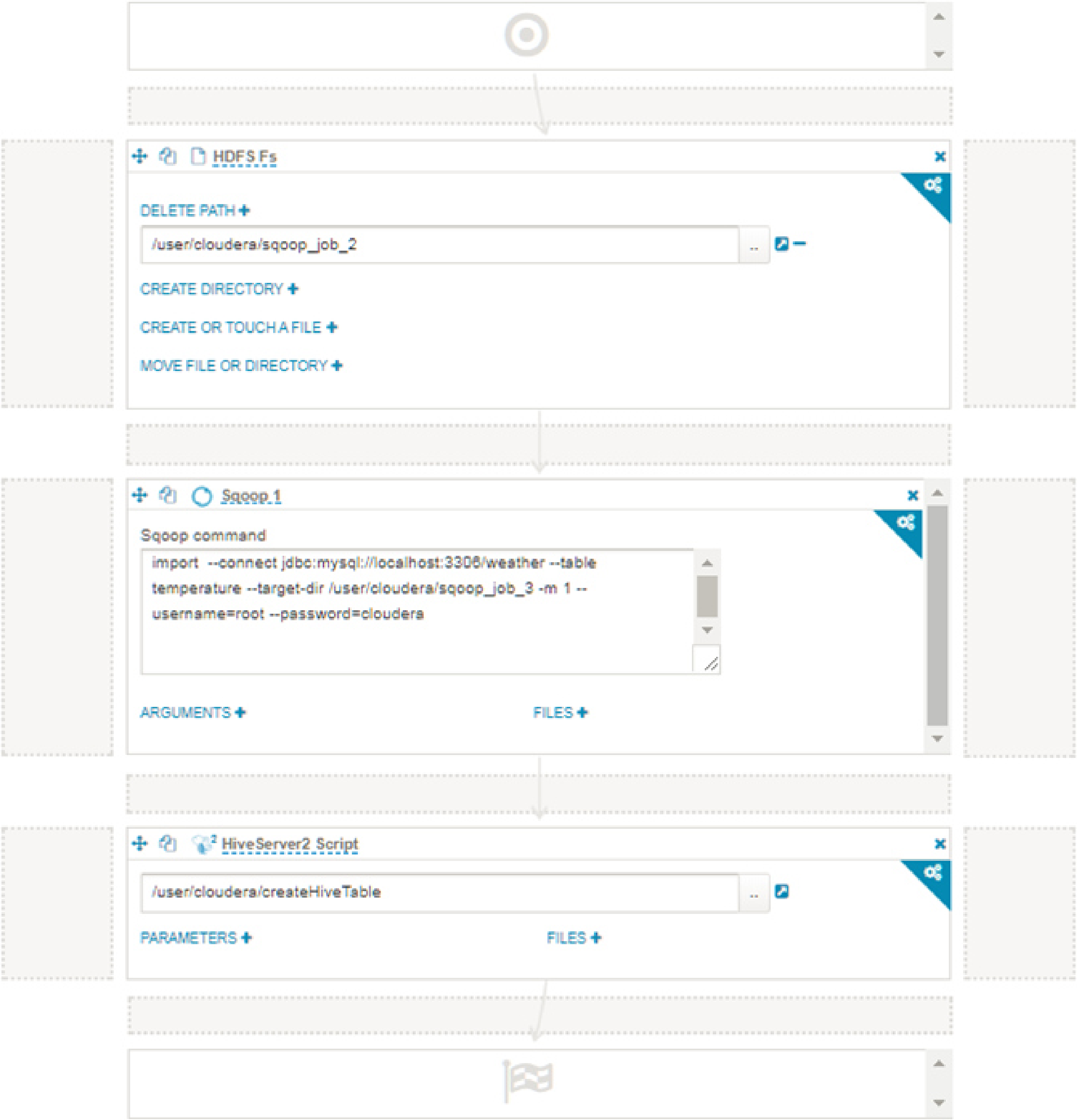
User and Host: user@host.com

Ssh command: ls

Add

Simple Oozie workflow

1. delete HDFS folder
2. Copy mySql table as text file to HDFS
3. Create Hive table based on this text file

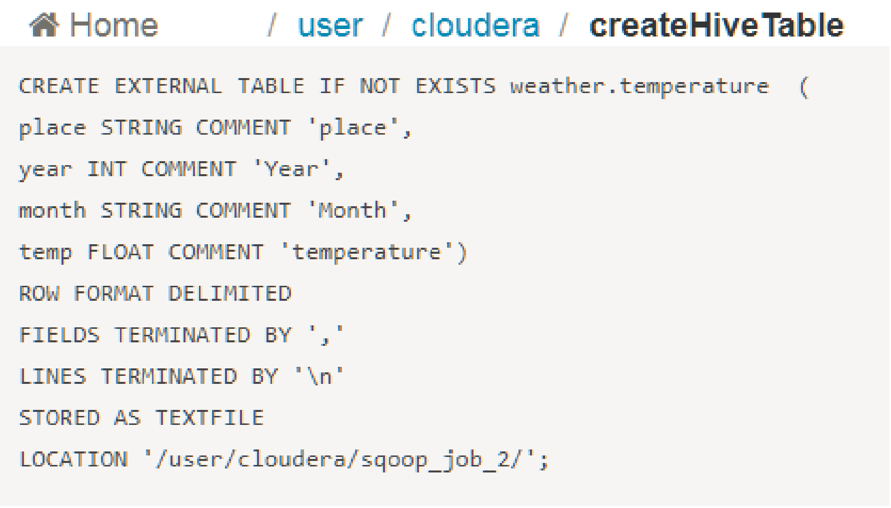


DELETE PATH: /user/cloudera/sqoop_job_2

Sqoop command

```
import  --connect jdbc:mysql://localhost:3306/weather --table
temperature --target-dir /user/cloudera/sqoop_job_3 -m 1 --
username=root --password=cloudera
```

/user/cloudera/createHiveTable

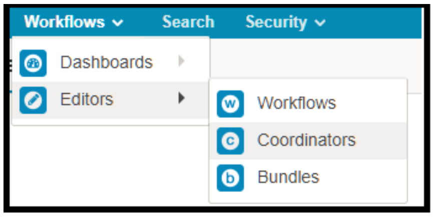


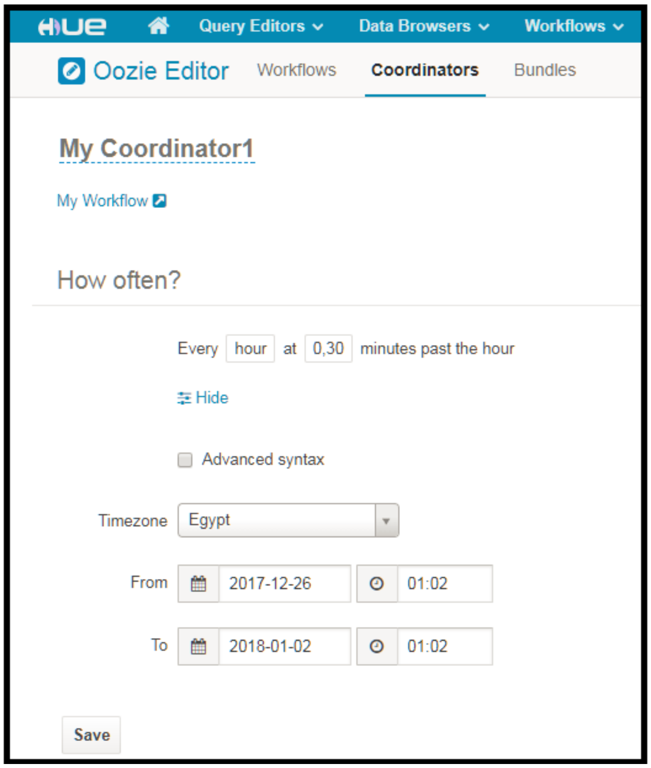
```
CREATE EXTERNAL TABLE IF NOT EXISTS weather.temperature  (
place STRING COMMENT 'place',
year INT COMMENT 'Year',
month STRING COMMENT 'Month',
temp FLOAT COMMENT 'temperature')
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/cloudera/sqoop_job_2/';
```

Example 6

Create schedual to run workflow

steps: Workflow > Editor > Coordinators





My Coordinator1

My Workflow

How often?

Every hour at 0,30 minutes past the hour

□ Advanced syntax

Timezone Egypt

From 2017-12-26 01:02

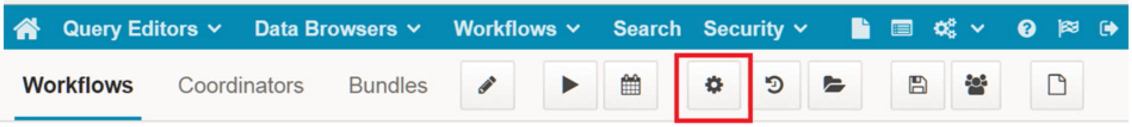
To 2018-01-02 01:02

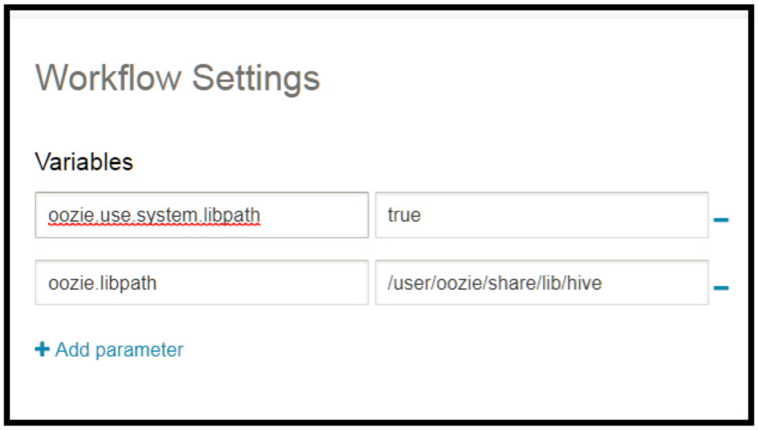
Notes:

Setup workflow settings

Workflow can contains some variables
To define new variable –> ${Variable}
Sometimes you need to define hive libpath in HUE to work with hive

ozzie.libpath : /user/oozie/share/lib/hive





Workflow Settings

Variables

oozie.use.system.libpath — true

oozie.libpath — /user/oozie/share/lib/hive

4. Data representation formats used for Big Data

Data representation formats used for Big Data, Common data representation formats used for big data include:

1. Row- or record-based encodings:

–Flat files / text files
–CSV and delimited files
–Avro / SequenceFile
–JSON
–Other formats: XML, YAML

2. Column-based storage formats:

–RC / ORC file
–Parquet

3. NoSQL Database

What is Parquet, RC/ORC file formats, and Avro?

1) Parquet

Parquet is a columnar storage format,

Allows compression schemes to be specified on a per-column level

Offer better write performance by storing metadata at the end of the file

Provides the best results in benchmark performance tests

2) RC/ORC file formats

developed to support Hive and use a columnar storage format

Provides basic statistics such as min, max, sum, and count, on columns

3) Avro

Avro data files are a compact, efficient binary format

5. What is NoSQL Databases?

What is NoSQL Databases?

NoSQL is a new way of handling variety of data. NoSQL DB can handle Millions of Queries per Sec while normal RDBMS can handle Thousands of Queries per Sec only, and both are follow CAP Theorem.

Types of NoSQL datastores:

• Key-value stores: MemCacheD, REDIS, and Riak

• Column stores: HBase and Cassandra

• Document stores: MongoDB, CouchDB, Cloudant, and MarkLogic

• Graph stores: Neo4j and Sesame

CAP Theorem

CAP Theorem states that in the presence of a network partition, one has to choose between consistency and availability.

- Consistency means Every read receives the most recent write or an error
- Availability means Every request receives a (non-error) response (without guarantee that it contains the most recent write)

How Famous Databases align with CAP Theorem?

- HBase, and MongoDB —> CP [give data Consistency but not Availability]
- Cassandra , CouchDB —> AP [give data Availability but not Consistency]
- Traditional Relational DBMS are CA [support Consistency and Availability but not network partition]