

cloudera



Contents

1.	About Cloudera	3
1.1.	Cloudera Director.....	3
1.2.	Cloudera Manager	4
2.	Objective	4
3.	Getting Started.....	5
3.1.	Accessing Cloudera Backend cluster details	5
3.2.	Accessing Cloudera Manager from Cloudera Director Web UI	11
3.3.	Hue	17
3.4.	Apache Spark (Run Spark App)	20
3.5.	Viewing Jobs in UI	23
3.6.	Hive	25
3.7.	Impala	27
3.8.	Uploading Roadshow data to ADLS:	29
4.	Power BI integration with Data Lake Store and Impala (Optional).....	31
4.1.	Integrating with Data Lake Store	31
4.2.	Integrating with Impala.....	40
5.	Reference	43
5.1.	Restart Cloudera Management Service	43
5.2.	Error Messages While Running the Spark Job	46

1. About Cloudera

Cloudera is an open-source Apache Hadoop distribution, CDH (Cloudera Distribution Including Apache Hadoop) targets enterprise-class deployments of that technology.

Cloudera provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in your enterprise. Cloudera products and solutions enable you to deploy and manage Apache Hadoop and related projects, manipulate and analyze your data, and keep that data secure and protected.

Cloudera develops a Hadoop platform that integrates the most popular Apache Hadoop open source software within one place. Hadoop is an ecosystem, and setting a cluster manually is a pain. Going through each node, deploying the configuration though the cluster, deploying your services, and restarting them on a wide cluster is a major drawback of distributed system and require lot of automation for administration. Cloudera developed a big data Hadoop distribution that handles installation and updates on a cluster in few clicks.

Cloudera also develop their own projects such as Impala or Kudu that improve hadoop integration and responsiveness in the industry.

1.1. Cloudera Director

Cloudera Director enables reliable self-service for using CDH and Cloudera Enterprise Data Hub in the cloud.

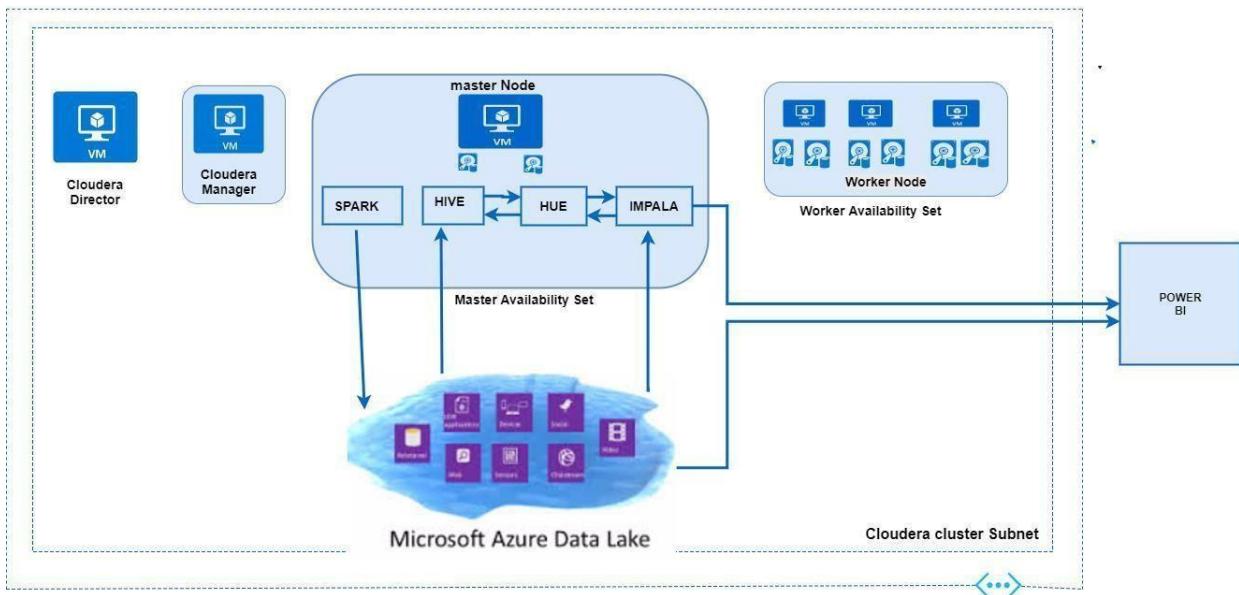
Cloudera Director provides a single-pane-of-glass administration experience for central IT to reduce costs and deliver agility, and for end-users to easily provision and scale clusters. Advanced users can interact with Cloudera Director programmatically through the REST API or the CLI to maximize time-to-value for an enterprise data hub in cloud environments.

Cloudera Director is designed for both long running and transient clusters. With long running clusters, you deploy one or more clusters that you can scale up or down to adjust to demand. With transient clusters, you can launch a cluster, schedule any jobs, and shut the cluster down after the jobs complete.

The Cloudera Director server is designed to run in a centralized setup, managing multiple Cloudera Manager instances and CDH clusters, with multiple users and user accounts. The server works well for launching and managing large numbers of clusters in a production environment.

1.2. Cloudera Manager

Cloudera Manager is a sophisticated application used to deploy, manage, monitor, and diagnose issues with your CDH deployments. Cloudera Manager provides the Admin Console, a web-based user interface that makes administration of your enterprise data simple and straightforward. It also includes the Cloudera Manager API, which you can use to obtain cluster health information and metrics, as well as configure Cloudera Manager.



2. Objective

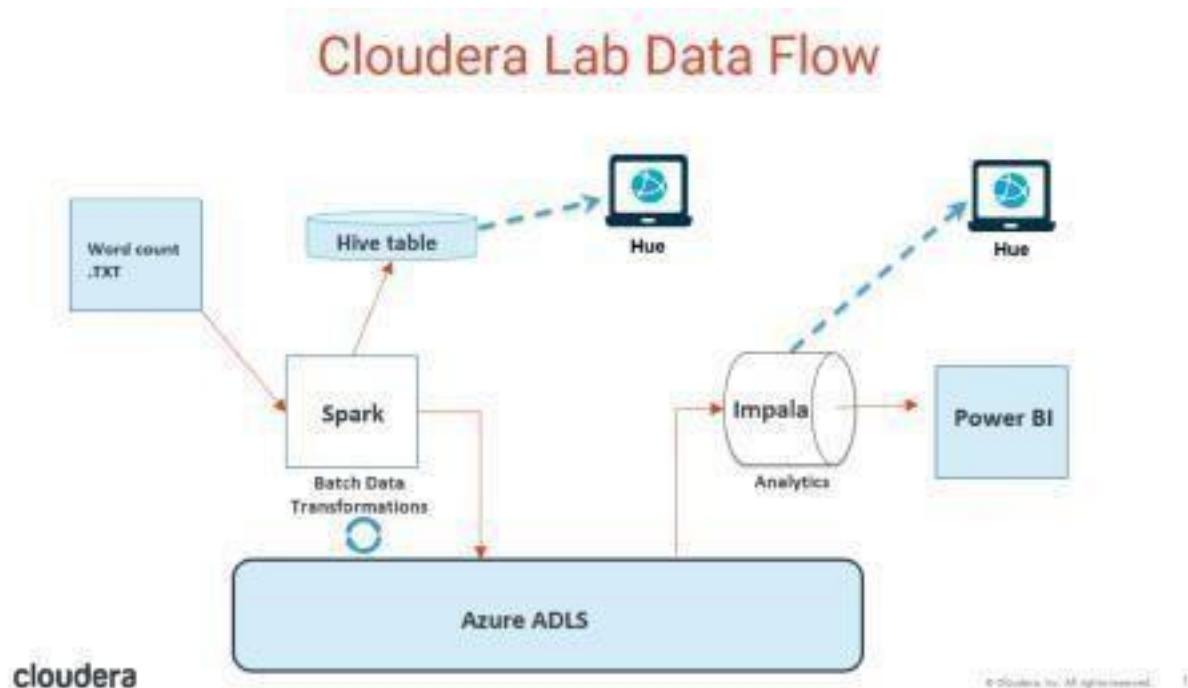
NOTE: As this test drive provides access to the full Cloudera Director platform, deployment can sometimes take up to **45 minutes**. While you wait, please feel free to review helpful content in this manual and on Cloudera's [Azure Marketplace product page](#), or on the Cloudera [website](#). Please also consider watching the demo video showcased on the test drive launch page on the Azure Marketplace web site.

The test drive provisions Cloudera Director, the environment, Cloudera Manager, and a cluster consisting of 1 master node and 3 worker nodes. The test drive also integrates with Azure Data Lake Store.

The use case scenario for this test drive is to provide users with a test Azure Data Lake Store and:

1. Run the **WordCount** app with Hadoop/Spark on ADLS.
2. Create a Hive table on the output, and query Hive from Hue.
3. Run query using Impala from Hue or Power BI.

The following diagram shows how the data in this test case flows from a .TXT file via Hue to ADLS, processed by Spark.



3. Getting Started

3.1. Accessing Cloudera Backend cluster details

Please login to the Azure portal and go to the Cloudera Director HOL Azure resource group allocated to you. Copy the DNS URLs for the **Cloudera Director**, **Manager** and **Master** nodes.

1. Go to the Resource Groups section and search by name for the Resource Group provided to you.

The screenshot shows the Microsoft Azure Resource groups interface. On the left, there's a navigation sidebar with options like 'New', 'Dashboard', 'Resource groups' (which is selected and highlighted with an orange border), 'All resources', 'Virtual machines', 'Storage accounts', 'Virtual networks', 'Load balancers', 'Availability sets', 'Network interfaces', 'Public IP addresses', 'Network security grou...', 'Azure Active Directory', 'App Services', and 'SQL databases'. The main area is titled 'Resource groups' and shows a list for 'SYSGAIN INC.' with one item: 'srikala-cloudera'. A search bar at the top right says 'Search resources'. Below the list, it says 'Subscriptions: 1 of 2 selected – Don't see a subscription? Switch directories' and lists 'srikala-cloudera' and 'Sysgain-CloudTry-Dev'. The table has columns for 'NAME' and 'TYPE'.

NAME	TYPE
srikala-cloudera	

2. Go to the virtual machine starting with "**cldr**" for the **Cloudera Director DNS Name**.

srikala-cloudera Resource group

Search (Ctrl+ /)

Add Columns Delete resource group Refresh Move

Essentials

Subscription name (change) Sysgain-CloudTry-Dev Deployments 8 Succeeded

Subscription ID 7eab3893-bd71-4690-84a5-47624df0b0e5

Filter by name... All types All locations Group by type

28 items

NAME	TYPE	LOCATION
6ce17224agysknqksa	Storage account	East US
e76396adlokjgcsca	Storage account	East US
cdedge-4f171cc5-63f9-43a5-8883-a148fc9...	Virtual machine	East US
cdmstr-6ce17224-c45b-4adf-831e-7344d3...	Virtual machine	East US
cdwork-39104616-0f77-4214-81c4-85586a7...	Virtual machine	East US
cdwork-58cb74de-22bc-4e64-a357-664f4a...	Virtual machine	East US
cdwork-e76396ad-27a0-4179-b193-91ee24...	Virtual machine	East US
cdr2jh	Virtual machine	East US

VIRTUAL MACHINE

- Click on the Cloudera Director virtual machine to get the DNS name. (See below)

cdr2jh Virtual machine

Search (Ctrl+ /)

Connect Start Restart Stop Move Delete Refresh

Resource group (change) srikala-cloudera

Status Running

Location East US

Subscription (change) Sysgain-CloudTry-Dev

Subscription ID 7eab3893-bd71-4690-84a5-47624df0b0e5

Computer name	cdr2jh
Operating system	Linux
Size	Standard DS12 v2 (4 vcpus, 28 GB memory)
Public IP address	40.76.23.71
Virtual network/subnet	clouderavnet/clouderasubnet
DNS name	cdr2jh.eastus.cloudapp.azure.com

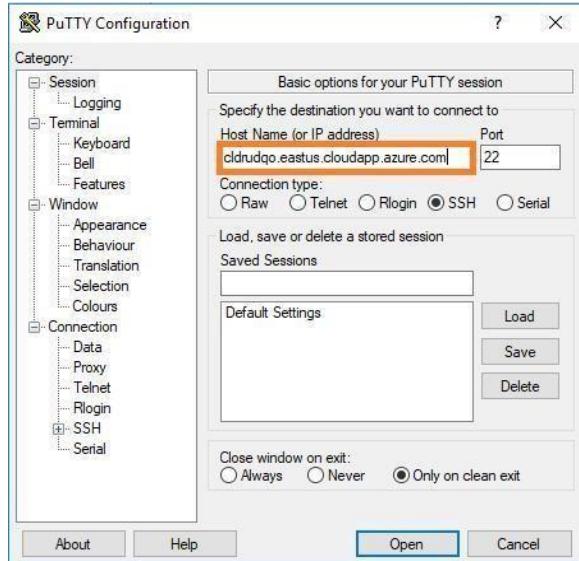
- Go to the virtual machine starting with "cdedge" for the Cloudera Manager DNS name.

- Click on the Cloudera Manager virtual machine to get the DNS name. (See below)

- Go to the virtual machine starting with "**cdmstr**" for the Cloudera Master DNS name.

7. Click on the Cloudera Master virtual machine to get the DNS name. (See below)

8. You must also access the Cloudera backend cluster details to get the Node Details. This is explained below.
9. Log in to the Cloudera Director VM using the **Cloudera Director FQDN** address gathered from the previous steps, and use an SSH tool like PuTTY (or Terminal on Mac), which we'll refer to in this walkthrough. ([Download PuTTY here](#)) E.g. **cldrhyic.eastus.cloudapp.azure.com**



10. Once connected, login to the Cloudera Director VM using the **Director Username** and then the **Director Password** from the provided test drive access credentials.

(**Note:** Passwords are hidden when typed or pasted in Linux terminals)

A screenshot of a terminal window titled 'cloudera@cldrudqo:~'. The window shows the command 'login as: cloudera' followed by 'Using keyboard-interactive authentication.' and a password prompt 'Password:'. The password field is redacted with black text.

11. All the Cloudera Backend cluster details are present in the **NodeDetails** file. **Copy the NodeDetails into a text file or Word document for reference**, these details will be used later.

To open the NodeDetails file use the following command.

```
cat NodeDetails
```

The NodeDetails file contains Node and URI details used by the Cloudera test drive environment. These are gathered using a script which pulls required data using the API calls.

```
cloudera@cldrjd3y:~$ login as: cloudera
Using keyboard-interactive authentication.
Password:
[cloudera@cldrjd3y ~]$ cat NodeDetails
Cloudera Director Node private IPAddress: 10.3.0.4
Cloudera Manager Node private IPAddress: 10.3.0.5
Cloudera Master Node private IPAddress: 10.3.0.9
Cloudera Hue Web UI URL: http://10.3.0.9:8888
Cloudera Hue Web UI Username/Password: admin/admin
Your Datalake Directory for the testdrive: demotdjd3y
Your Datalake Endpoint for the testdrive: adl://cddatalakejd3y.azuredatalakestore.net
Your Datalakename: cddatalakejd3y
Your ClientID: [REDACTED]
Your Clientsecret: [REDACTED]
Your TenantID: dcf9e4d3-f44a-4c28-be12-8245c0d35668
Your SubscriptionID: deb67cbe-e165-4aad-bfad-497f54b02674
Your Output Data files on Datalake for the testdrive: adl://cddatalakejd3y.azuredatalakestore.net/demotdjd3y/WordCount
[cloudera@cldrjd3y ~]$
```

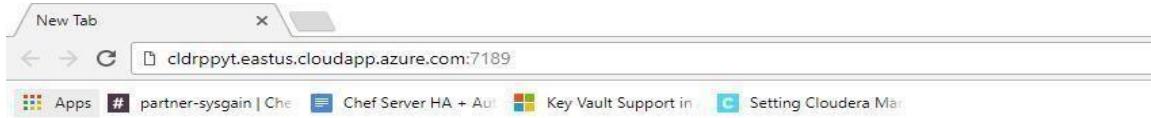
3.2. Accessing Cloudera Manager from Cloudera Director Web UI

After deploying a cluster, you can manage it using Cloudera Manager.

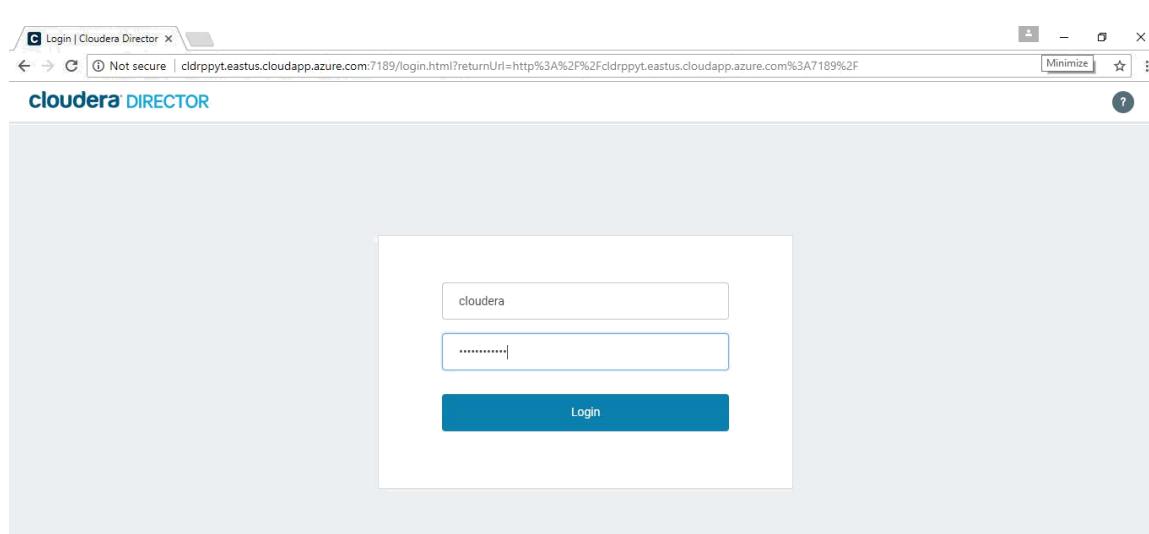
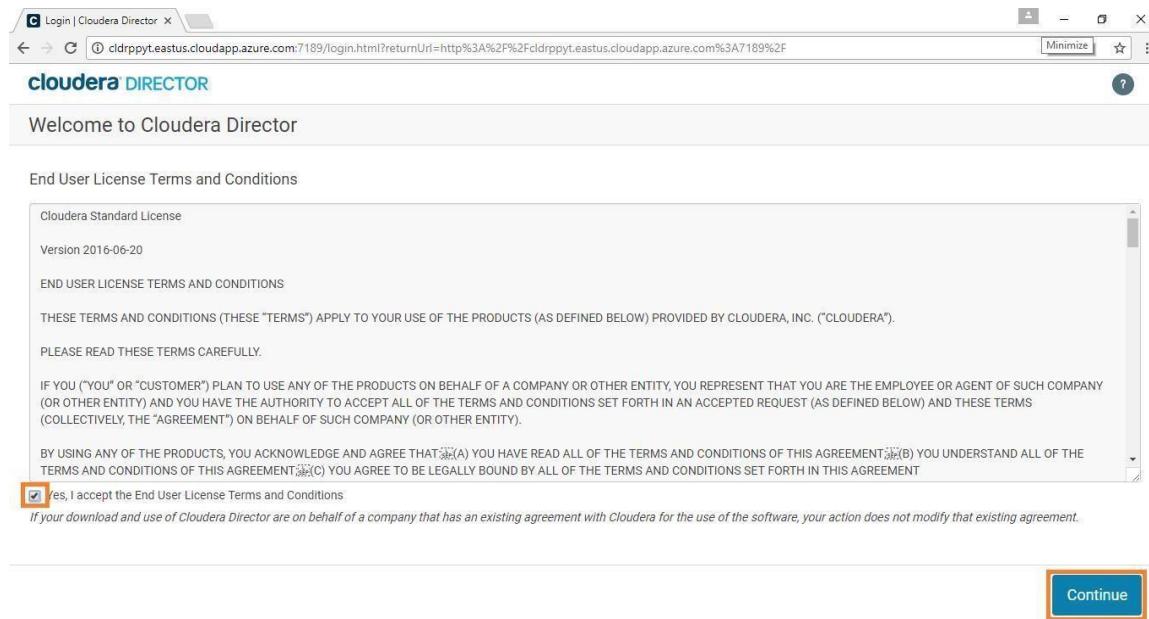
12

1. Access the Cloudera Director Web UI using the **Cloudera Director Access URL** provided in the Access Information. Enter it into a web browser.

Eg: **cldrhyic.eastus.cloudapp.azure.com:7189**



2. **Accept the End User License Terms and Conditions** and click on **Continue**.



3. Login to the Cloudera Director web console using **CD-WEB UI Username** and **Password** from the Access Information.

4. The Cloudera Director console should open. You should now be able to browse through to see the Environment, Cloudera Manager and the Cluster details.

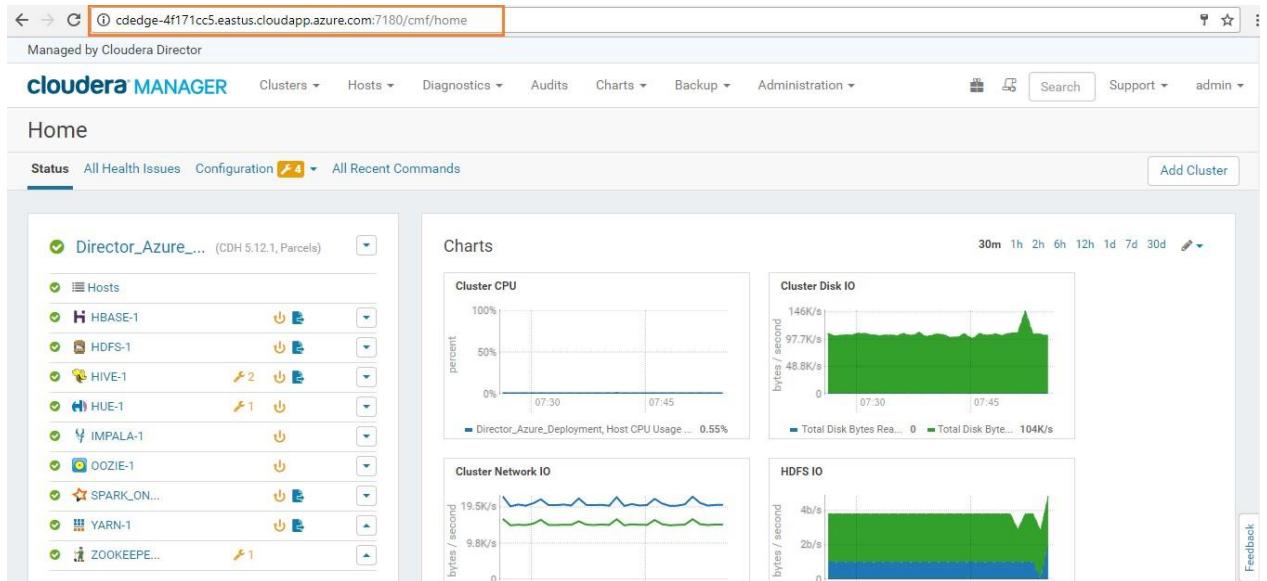
The screenshot shows the Cloudera Director dashboard. On the left, there's a sidebar with 'Filters' for Cluster Status (Ready: 1) and Cluster Health (Good: 1). The main area displays a table with columns: Cluster name, Environment, Services, Hosts, HDFS Used, Physical Memory Capacity, Host CPU Usage, and Non-HDFS Used. A cluster named 'Director_Azure_Deployment' is selected, highlighted with a red box. Below the table, a message states 'All services are healthy (9)'.

5. Use the Cloudera Manager FQDN address, along with the **port** number, and paste it in new browser tab.

EX: cdedge-4f171cc5.eastus.cloudapp.azure.com:7180

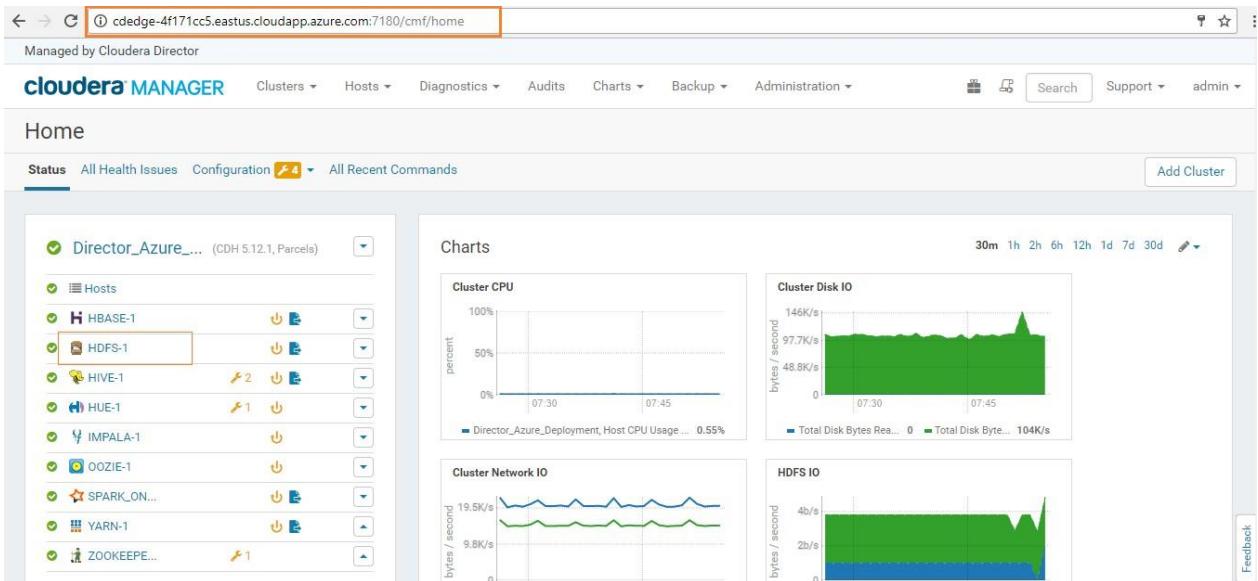
The screenshot shows the Cloudera Manager login page. The URL in the browser is 'Not secure | cdedge-4f171cc5.eastus.cloudapp.azure.com:7180/cmf/login'. The page has a 'cloudera MANAGER' header and a 'Support Portal' link. The main content is a login form with fields for 'Username' and 'Password', a 'Remember me' checkbox, and a 'Log In' button.

6. Login to the Cloudera Manager Console using **CM-WEB UI Username** and **CM-WEB UI Password** from the Access Information.



Note: The next step is to Restart Stale Services. We must do this to get the Azure Service Principle updated to the configuration file `site-core.xml`, which is required to integrate with Azure Data Lake Store.

7. In Cloudera Manager, click on the **HDFS-1** service to **Restart Stale Services**.



8. Click on the **Restart Stale Services** icon as shown in the below screenshot.

Managed by Cloudera Director

cloudera MANAGER

HDFS-1 (Director_Azure_Deployment) Actions

30 minutes preceding Sep 20, 7:56 AM CDT

Status Instances Configuration Commands File Browser Charts Library Cache Statistics Audits NameNode Web UI Quick Links

HDFS Summary

- Configured Capacity: 1.7 GiB / 20.8 TiB

Health Tests

Show 7 Good

Status Summary

DataNode: 3 Good Health

Charts

- HDFS Capacity**: Shows capacity over time from 07:30 to 07:45. Legend: Configured Capacity (20.8T), HDFS Used (1.7G), Non-HDFS Used (0).
- Total Blocks Written Across DataNodes**: Shows blocks written per second over the same period. Legend: HDFS-1, Total Blocks Written Across DataNodes (0.05).
- Total Transceivers Across DataNodes**: Shows transceivers over time.
- Transceivers Across DataNodes**: Shows transceivers over time.

- Click on the **Restart Stale Services** button so the cluster can read the new configuration information.

Managed by Cloudera Director

cloudera MANAGER

Stale Configurations (Director_Azure_Deployment)

Filters Clear All

FILE		File: core-site.xml
File: core-site.xml	2	... @@ -129,6 +129,34 @@
File: creds.localceks	0	129 129 <property>
File: hadoop-conf/core-site.xml	1	130 138 <name>hadoop.http.logs.enabled</name>
File: hbase-conf/core-site.xml	0	131 131 <value>true</value>
File: hive-conf/core-site.xml	0	132 132 </property>
File: yarn-conf/core-site.xml	0	133 + <property>
		134 + <name>dfs.adls.oauth2.client.id</name>
		135 + <value>d49e2d2d-e97e-4f3e-aa00-45e202305782</value>
		136 + </property>
		137 + <property>
		138 + <name>dfs.adls.oauth2.refresh.url</name>
		139 + <value>https://login.windows.net/dcf9e4d3-f44a-4c28-be12-8245c0d35668/oauth2/token</value>
		140 + </property>
		141 + <property>
		142 + <name>dfs.adls.oauth2.credential</name>
		143 + <value>+u9pF9/ZepqBKUkLoLcAUV8vQmB9xwhi+RZT7Am/Ys=</value>
SERVICE	Clear	144 + </property>
HBASE-1	2	145 + <property>
HDFS-1	3	146 + <name>dfs.adls.oauth2.access.token.provider.type</name>
HIVE-1	4	147 + <value>ClientCredential</value>
HUE-1	4	148 + </property>
IMPALA-1	2	149 + <property>
OOZIE-1	3	150 + <name>fs.adl.impl</name>
SPARK_ON_YARN-1	2	151 + <value>org.apache.hadoop.fs.adl.AdlFileSystem</value>
		152 + </property>
		153 + <property>

Activate Windows
Go to Settings to activate Windows
Restart Stale Services

- Click on the **Restart Now** button.

The screenshot shows the Cloudera Manager interface for restarting stale services. At the top, it says "Managed by Cloudera Director" and "cloudera MANAGER". Below that, there are two buttons: "Restart Stale Services" and "Review Changes". A note states: "All services running with outdated configurations in the cluster and their dependencies will be restarted." There is a checkbox for "Re-deploy client configuration". At the bottom right, there is a "Feedback" link and a "Activate Windows" button with a "Restart Now" link.

11. Wait until all requested services are restarted. Once all the services are restarted, click on the **Finish** button.

The screenshot shows the "Command Details Step" of the restart wizard. It displays a table of completed steps:

Step	Description	Time	Duration
1	Execute global command Wait for configuration staleness computation	Sep 20, 7:58:19 AM	35ms
2	Execute command Restart on cluster Director_Azure_Deployment	Sep 20, 7:58:19 AM	3.3m

Below the table, it says "All services successfully restarted." There are buttons for "Back", "1", "2", and "Finish". A "Feedback" link and an "Activate Windows" button with a "Restart Now" link are also present.

12. Now we have the **Cloudera Director** ready, with **Cloudera Manager** and **Cluster** (1 master and 3 workers).

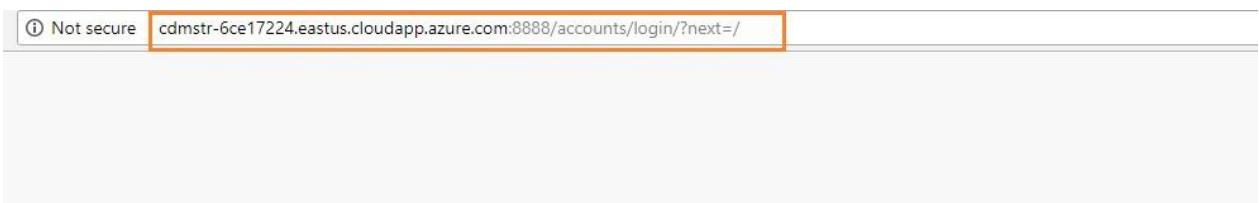
Note: Please visit section **5.1** in the **Reference** section later in this guide for additional details and help for any error messages you may encounter.

3.3. Hue

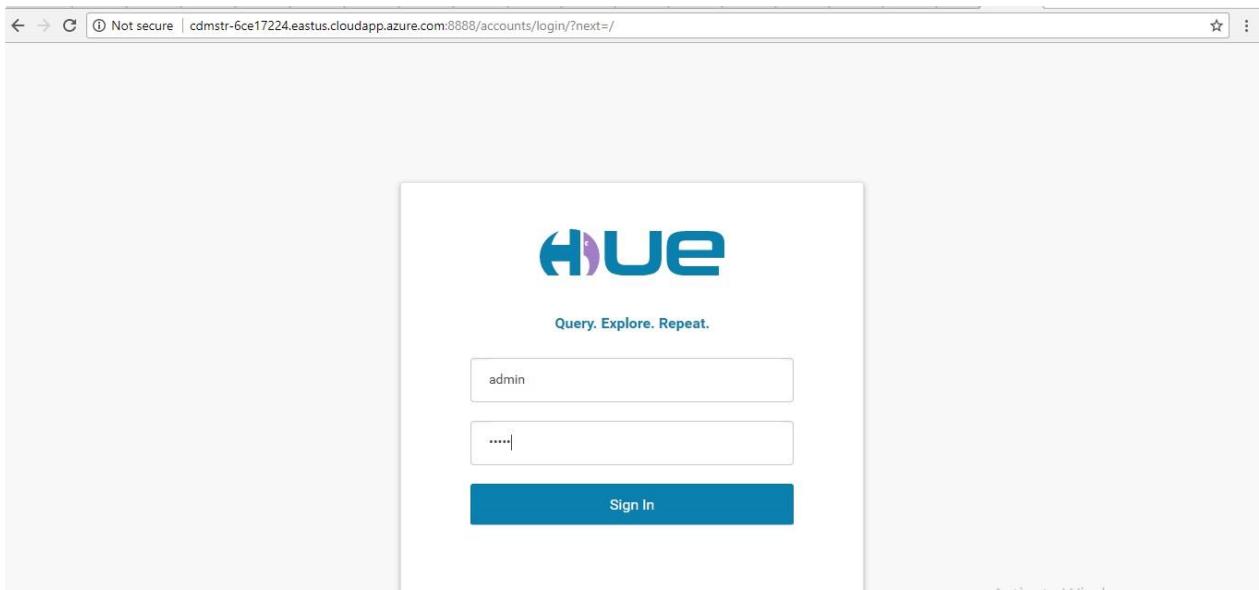
Hue is a set of web applications that enable you to interact with a CDH cluster. Hue applications let you browse HDFS and manage a Hive metastore. They also let you run Hive and Cloudera Impala queries, HBase and Sqoop commands, Pig scripts, MapReduce jobs, and Oozie workflows.

1. Copy the **Cloudera Hue Web URL** using the cloudera master DNS server url with port 8888 as shown in below example and paste it in browser – which opens the Hue console.

Example: <http://cdmstr-6ce17224.eastus.cloudapp.azure.com:8888>



2. Create a Hue Account by giving **Cloudera Hue Web UI Username/Password** from the **NodeDetails** file. (**Username/Password: admin/admin**)



3. You will login into the Hue dashboard. CDH 5.12 has a new Hue UI. We recommend switching to Hue 3 from the admin tab (see screenshot below).

The screenshot shows the Hue web interface for Impala. The top navigation bar includes a back arrow, forward arrow, refresh button, and a URL bar with the address `cdmstr-6ce17224.eastus.cloudapp.azure.com:8888/hue/editor/?type=impala`. Below the header is the Hue logo and a search bar labeled "Search data and saved documents...". The main area is titled "Impala" with sub-titles "Add a name..." and "Add a description...". It features a text input field with placeholder text "Example: SELECT * FROM tablename, or press CTRL + space" and a toolbar with icons for copy, paste, cut, and settings. On the left, there's a sidebar with "Databases" and "Tables" sections. The "Tables" section shows one entry: "default" with "(1)" and a "T" icon. Below the table list are "Query History" and "Saved Queries" tabs. On the right, a sidebar titled "Assistant" contains links for "My Profile", "Manage Users", "Switch to Hue 3" (which is highlighted with a red box), "Help", "Welcome Tour", "Check Configuration", and "Sign out".

4. On the right side of the page, click on the **HDFS browser** icon, as shown in the below screenshot.

This screenshot shows the same Hue interface as the previous one, but with a specific element highlighted. The "HDFS browser" icon, which is a folder icon with a plus sign, is highlighted with a red box. The rest of the interface is identical to the first screenshot, showing the "Impala" editor screen.

5. Copy the data of **inputfile** from the below link. Give any name to the file (Eg: '**data**' or '**input**'), then save it in **.txt** format.

<https://aztdrepo.blob.core.windows.net/clouderadirector/inputfile.txt>

6. Once ready, click on **Upload** on the Hue file browser page (see below).

Note: Please ensure the inputfile is uploaded to the path **/user/admin** (see below):

The screenshot shows the Hue File Browser interface. At the top, there's a navigation bar with links for 'Query Editors', 'Data Browsers', and 'Workflows'. Below the navigation bar is a search bar labeled 'Search for file name' and a toolbar with buttons for 'Actions', 'Move to trash', 'Upload' (which is highlighted with a red box), and 'New'. The current path is '/ user / admin'. On the right, there's a 'History' dropdown. The main area displays a table of files with columns for Name, Size, User, Group, Permissions, and Date. Two entries are listed: 'hdfs' (size 0, user hdfs, group supergroup, permissions drwxr-xr-x, date September 20, 2017 04:28 AM) and 'admin' (size 0, user admin, group admin, permissions drwxr-xr-x, date September 20, 2017 04:28 AM).

7. Select the saved .txt file to upload it.

This screenshot shows the same Hue File Browser interface after a file has been uploaded. The 'Upload' button in the toolbar is still highlighted. In the main file list, the 'admin' entry now has a small 'file' icon next to it, indicating it is a file. The other file, 'hdfs', remains unchanged.

This screenshot shows the Hue File Browser with an 'Upload' dialog box open over the file list. The dialog title is 'Upload to /user/admin'. It contains a text input field 'or drag and drop them here' and a 'Select files' button. The background file list shows the same two entries as before: 'hdfs' and 'admin'. The 'Upload' button in the toolbar is still highlighted.

Name	Size	User	Group	Permissions	Date
j		hdfs	supergroup	drwxr-xr-x	September 20, 2017 04:28 AM
.		admin	admin	drwxr-xr-x	September 20, 2017 06:07 AM
inputfile.txt	143 bytes	admin	admin	-rw-r--r-	September 20, 2017 06:07 AM

8. The .txt file is now uploaded to Hue. The Spark application will use this data as input and provide the output to ADLS.

3.4. Apache Spark (Run Spark App)

Spark is the open standard for flexible in-memory data processing that enables batch, realtime, and advanced analytics on the Apache Hadoop platform.

To use it properly, it is also a good idea to install "dos2unix". dos2unix is a program that converts DOS to UNIX text file format, ensuring everything will run in a Linux environment.

1. Login to the **Master VM** by typing in the below command in the open terminal session from before (**copy/paste may not work**):

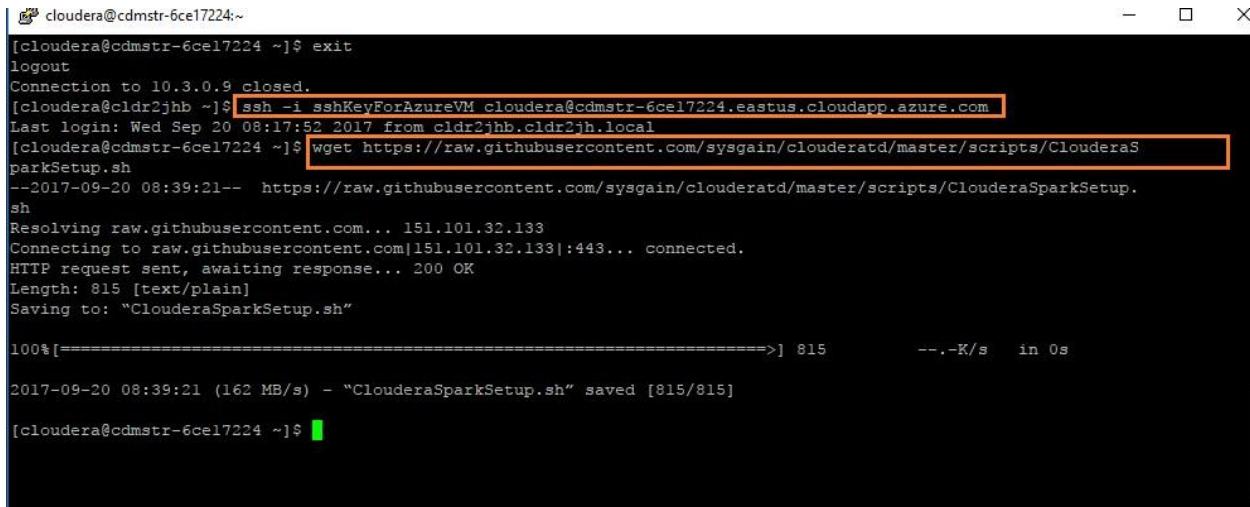
```
ssh -i sshKeyForAzureVM cloudera@<Master Node FQDN>
```

```
[cloudera@cldr2jh ~]$ ssh -i sshKeyForAzureVM cloudera@cdmstr-6ce17224.eastus.cloudapp.azure.com
Last login: Wed Sep 20 08:17:52 2017 from cldr2jh.cldr2jh.local
[cloudera@cdmstr-6ce17224 ~]$
```

2. **Download** the following script file using the below command.

The script contains the spark app (**WordCount**). The application counts the number of occurrences of each letter in words which have more characters than a given threshold.

```
wget https://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/ClouderaSparkSetup.sh
```



```
[cloudera@cdmstr-6ce17224 ~]$ exit
[cloudera@cdmstr-6ce17224 ~]$ ssh -i sshKeyForAzureVM cloudera@cdmstr-6ce17224.eastus.cloudapp.azure.com
Connection to 10.3.0.9 closed.
[cloudera@cdmstr-6ce17224 ~]$ ssh -i sshKeyForAzureVM cloudera@cdmstr-6ce17224.eastus.cloudapp.azure.com
Last login: Wed Sep 20 08:17:52 2017 from cldr2jhb.cldr2jh.local
[cloudera@cdmstr-6ce17224 ~]$ wget https://raw.githubusercontent.com/sysgain/clouderatd/master/scripts/ClouderaSparkSetup.sh
--2017-09-20 08:39:21-- https://raw.githubusercontent.com/sysgain/clouderatd/master/scripts/ClouderaSparkSetup.sh
Resolving raw.githubusercontent.com... 151.101.32.133
Connecting to raw.githubusercontent.com|151.101.32.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 815 [text/plain]
Saving to: "ClouderaSparkSetup.sh"

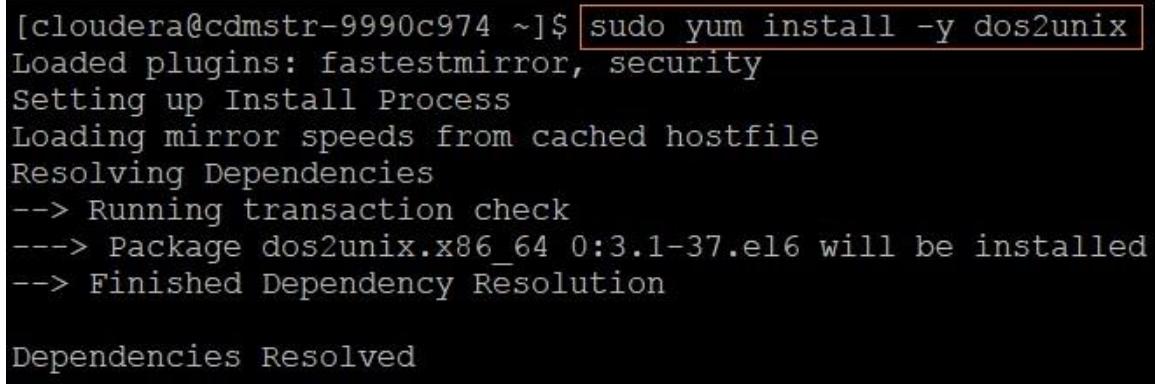
100%[=====] 815 --.-K/s in 0s

2017-09-20 08:39:21 (162 MB/s) - "ClouderaSparkSetup.sh" saved [815/815]

[cloudera@cdmstr-6ce17224 ~]$
```

3. To install **dos2unix**, run the following command:

```
sudo yum install -y dos2unix
```



```
[cloudera@cdmstr-9990c974 ~]$ sudo yum install -y dos2unix
Loaded plugins: fastestmirror, security
Setting up Install Process
Loading mirror speeds from cached hostfile
Resolving Dependencies
--> Running transaction check
--> Package dos2unix.x86_64 0:3.1-37.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved
```

4. To give permissions to **ClouderaSparkSetup.sh** file, run the following commands:

```
dos2unix /home/cloudera/ClouderaSparkSetup.sh
chmod 755 /home/cloudera/ClouderaSparkSetup.sh
```

```

cloudera@cdmstr-6ce17224:~$ Installed:
cloudera@cdmstr-6ce17224:~$ dos2unix /home/cloudera/ClouderaSparkSetup.sh
cloudera@cdmstr-6ce17224:~$ chmod 755 /home/cloudera/ClouderaSparkSetup.sh
cloudera@cdmstr-6ce17224:~$ sh ClouderaSparkSetup.sh demotdjhnb cdmstr-tce1/224.eastus.cloudapp.azure.com inputfile.txt adl://cddatalake2jhb.azuredatalakestore.net
--2017-09-20 08:42:52-- https://aztdrepo.blob.core.windows.net/clouderadirector/wordcount.jar
Resolving aztdrepo.blob.core.windows.net... 52.238.56.168
Connecting to aztdrepo.blob.core.windows.net|52.238.56.168|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6371588 (6.1M) [application/octet-stream]
Saving to: "/home/cloudera/wordcount.jar"

100%[=====] 6,371,588 5.05M/s in 1.2s

2017-09-20 08:42:54 (5.05 MB/s) - "/home/cloudera/wordcount.jar" saved [6371588/6371588]

17/09/20 08:42:55 INFO spark.SparkContext: Running Spark version 1.6.0
17/09/20 08:42:56 INFO spark.SecurityManager: Changing view acls to: cloudera
17/09/20 08:42:56 INFO spark.SecurityManager: Changing modify acls to: cloudera
17/09/20 08:42:56 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(cloudera); users with modify permissions: Set(cloudera)

```

5. Run the following command to execute the **ClouderaSparkSetup.sh** script:

```
sh ClouderaSparkSetup.sh <DataLake Directory> <Master Node FQDN>
<inputfile.txt> <DataLake Endpoint for the testdrive>
```

Note: Replace the above values from **NodeDetails** and give the Name of the input file that you have just uploaded in Hue in the place of **<inputfile.txt>**

Example:

```
sh ClouderaSparkSetup.sh demotdah6k cdmstr-6ce17224.eastus.cloudapp.azure.com inputfile.txt
adl://cddatalakeah6k.azuredatalakestore.net
```

```

cloudera@cdmstr-6ce17224:~$ Installed:
cloudera@cdmstr-6ce17224:~$ dos2unix /home/cloudera/ClouderaSparkSetup.sh
cloudera@cdmstr-6ce17224:~$ chmod 755 /home/cloudera/ClouderaSparkSetup.sh
cloudera@cdmstr-6ce17224:~$ sh ClouderaSparkSetup.sh demotdjhnb cdmstr-tce1/224.eastus.cloudapp.azure.com inputfile.txt adl://cddatalake2jhb.azuredatalakestore.net
--2017-09-20 08:42:52-- https://aztdrepo.blob.core.windows.net/clouderadirector/wordcount.jar
Resolving aztdrepo.blob.core.windows.net... 52.238.56.168
Connecting to aztdrepo.blob.core.windows.net|52.238.56.168|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6371588 (6.1M) [application/octet-stream]
Saving to: "/home/cloudera/wordcount.jar"

100%[=====] 6,371,588 5.05M/s in 1.2s

2017-09-20 08:42:54 (5.05 MB/s) - "/home/cloudera/wordcount.jar" saved [6371588/6371588]

17/09/20 08:42:55 INFO spark.SparkContext: Running Spark version 1.6.0
17/09/20 08:42:56 INFO spark.SecurityManager: Changing view acls to: cloudera
17/09/20 08:42:56 INFO spark.SecurityManager: Changing modify acls to: cloudera
17/09/20 08:42:56 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(cloudera); users with modify permissions: Set(cloudera)

```

6. By executing the above script, the data has been stored to ADLS using Spark application.

Note: Please visit section **5.2** in the **Reference** section for additional details and help for any error messages you may encounter.

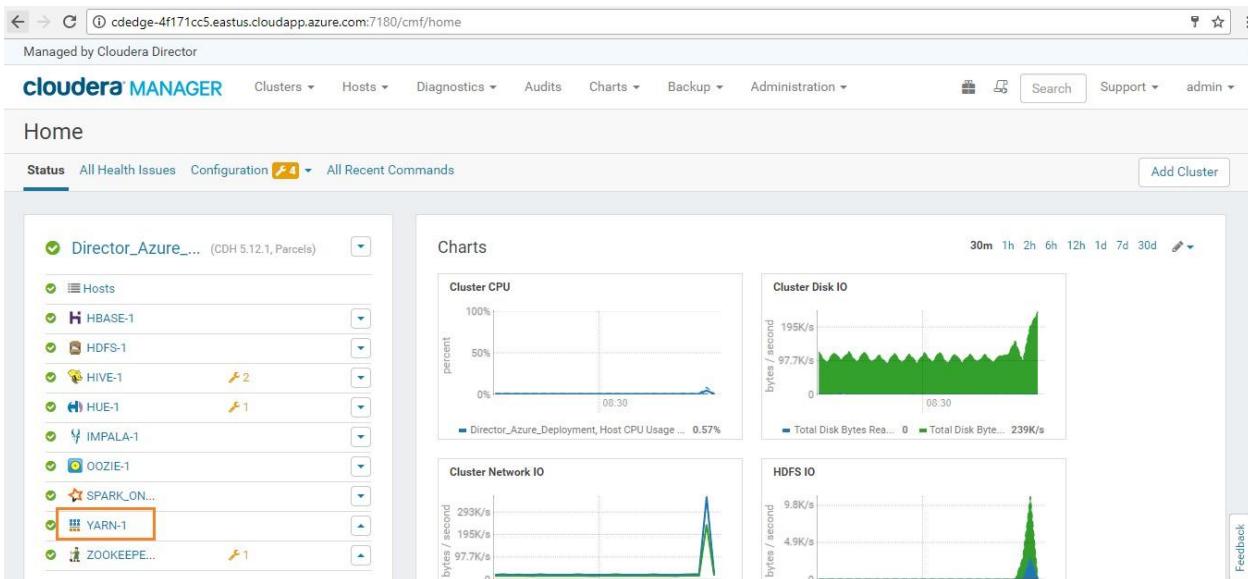
3.5. Viewing Jobs in UI

Next, navigate to the Yarn/Spark UI to see the WordCount Spark job.

1. Go to <http://<Manager Node FQDN>:7180/cmf/home>

Example: <http://cedge-4f171cc5.eastus.cloudapp.azure.com:7180>

2. Click on **YARN-1**.



3. Click on the **Applications** tab in the top navigation menu to view the available jobs.

Managed by Cloudera Director

cloudera MANAGER

YARN-1 (Director_Azure_Deployment) Actions ▾

Clusters ▾ Hosts ▾ Diagnostics ▾ Audits Charts ▾ Backup ▾ Administration ▾

Search Support ▾ admin ▾

Status Instances Configuration Commands **Applications** Resource Pools Charts Library Audits Web UI ▾ Quick Links ▾

30 minutes preceding Sep 20, 8:46 AM CDT

Search for YARN applications, e.g. 'pool = default' or press space to start typeahead.

30m 1h 2h 6h 12h 1d 7d 30d

Workload Summary (For Completed Applications)

Allocated Memory Seconds
60K 1

Allocated VCore Seconds
45 1

CPU Time

Duration

Results Charts

09/20/2017 8:43 AM - 09/20/2017 8:43 AM

Spark Count
ID: application_1505912365943_0001 Type: SPARK
Pool: root.users.cloudera Duration: 22.68s
User: cloudera Allocated Memory Seconds: 60K

Collect Diagnostic Data Export Select Attributes

Feedback

Each job has Summary and Detail information. A job Summary includes the following attributes: **start & end timestamps**, **query name** (if the job is part of a Hive query), **queue**, **job type**, **job ID**, and **user**.

4. You can also see the available applications by navigating to the Spark UI:

1. Go to <http://<Manager Node private FQDN>:7180/cmf/home>

Example: <http://cedge-4f171cc5.eastus.cloudapp.azure.com:7180>

2. Click on **SPARK_ON_YARN-1**. (May appear as '**SPARK_ON...**')

Managed by Cloudera Director

cloudera MANAGER

Home

Status All Health Issues Configuration **SPARK_ON...** All Recent Commands Add Cluster

Director_Azure.... (CDH 5.12.1, Parcels)

- Hosts
- HBASE-1
- HDFS-1
- HIVE-1
- HUE-1
- IMPALA-1
- OOZIE-1
- SPARK_ON...**
- YARN-1
- ZOOKEEPER-1

Charts

Cluster CPU

percent

0% 50% 100%

09:45

Director_Azure_Deployment, Host CPU Usage A... 0.6%

Cluster Disk IO

bytes / second

0 48.8K/s 97.7K/s

09:45

Total Disk Bytes Read... 0 Total Disk Byte... 138K/s

Cluster Network IO

bytes / second

0 19.5K/s 39.1K/s 58.6K/s

09:45

HDFS IO

bytes / second

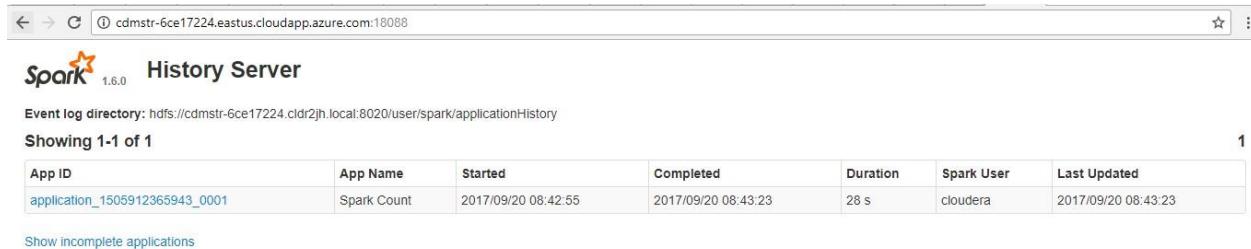
0 10b/s 20b/s 30b/s

09:45

Feedback

3. Navigate to the History Server WEB UI by going to <http://<Master FQDN>:18088>

Example: <http://cdmstr-4f171cc5.eastus.cloudapp.azure.com:18088/>



The screenshot shows the Spark History Server interface at the URL <http://cdmstr-6ce17224.eastus.cloudapp.azure.com:18088/>. The page title is "Spark History Server". It displays a table of application logs. The table has columns: App ID, App Name, Started, Completed, Duration, Spark User, and Last Updated. There is one entry: "application_1505912365943_0001" with "Spark Count" as the app name, started at 2017/09/20 08:42:55 and completed at 2017/09/20 08:43:23, duration 28 s, user cloudera, and last updated at 2017/09/20 08:43:23. Below the table is a link "Show incomplete applications".

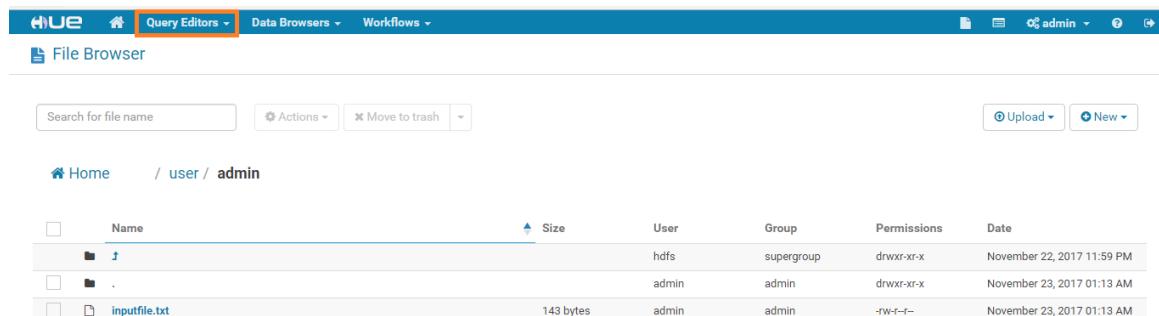
Note: Please visit section **5.2** in the **Reference** section for additional details and help for any error messages you may encounter.

3.6. Hive

Apache Hive is a data warehouse software project built on top of Apache Hadoop for providing data summarization, query, and analysis. Hive gives a SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

Now we will create a Hive table from the output of the Spark application stored on ADLS and run a Hive query from Hue.

1. Navigate to the **Query Editors** drop-down menu in the Hue WEB UI and click on **Hive**.



The screenshot shows the Hue File Browser interface. The top navigation bar includes "HUE", "Query Editors" (which is highlighted), "Data Browsers", "Workflows", and "admin". Below the navigation is a search bar "Search for file name" and action buttons "Actions", "Move to trash", "Upload", and "New". The main area shows a directory listing under "/ user / admin". The table has columns: Name, Size, User, Group, Permissions, and Date. The entries are: a folder named "t" (Size: 0 bytes, User: hdfs, Group: supergroup, Permissions: drwxr-xr-x, Date: November 22, 2017 11:59 PM); a folder named "." (Size: 0 bytes, User: admin, Group: admin, Permissions: drwxr-xr-x, Date: November 23, 2017 01:13 AM); and a file named "inputfile.txt" (Size: 143 bytes, User: admin, Group: admin, Permissions: -rw-r--r--, Date: November 23, 2017 01:13 AM).

2. In the default database, execute the below query:

```
create external table <tablename> (character varchar(1), frequency  
varchar(10)) row format delimited fields terminated by ',' lines terminated  
by '\n' stored as textfile location "<Output Data files on Datalake for the  
testdrive>";
```

Note: Add any name for <tablename> and replace the <Output Data files on Datalake for the testdrive> placeholder with the corresponding data from the NodeDetails file.

The screenshot shows the Hue interface for Apache Hive. The top navigation bar includes links for Query Editors, Data Browsers, Workflows, and a user dropdown for 'admin'. The main area is titled 'Hive' and shows a table creation dialog. The 'Tables' section on the left indicates '(0)' tables. The central pane displays the SQL query:`create external table testtable (character varchar(1), frequency
varchar(10)) row format delimited fields terminated by ',' lines terminated
by '\n' stored as textfile location "adl://cddatalake2jhb.azuredatalakestore.net/demotd2jhb/WordCount";`

The status bar at the bottom of the central pane says 'Success.' Below the query history, the query is listed with a timestamp 'a few seconds ago' and the same SQL code. To the right, the 'Tables' section shows 'No tables identified.'

3. View the table by giving the query:

```
Select * from <tablename>
```

```

1 create external table testtable (character varchar(1), frequency  varchar(10)) row format delimited fields
2
3 Select * from testtable

```

	testtable.character	testtable.frequency
1	z	1
2	p	2
3	x	1
4	t	6
5	b	1
6	h	3
7	n	6
R	f	?

3.7. Impala

Impala is an open source, massively parallel processing query engine on top of clustered systems like Apache Hadoop. It is an interactive SQL like query engine that runs on top of Hadoop Distributed File System (HDFS). It integrates with HIVE metastore to share the table information between both the components.

1. **Note:** Impala now integrates with ADLS from version CDH 5.12.
2. Navigate to the **Query Editor** drop-down menu and click on **Impala**.

Example: SELECT * FROM tablename, or press CTRL + space

You don't have any saved query.

3. Execute the below query in the default database to sync the data from Hive to Impala:

```
INVALIDATE METADATA;
```

The screenshot shows the Hue interface for an Impala connection. The top navigation bar includes links for Query Editors, Data Browsers, Workflows, and a user dropdown for 'admin'. The main area has tabs for 'Tables' and 'Assistant'. A search bar at the top says 'Search SQL tables...'. Below it, under 'Tables', there is a section for the 'default' database which shows '(0)' tables and a note 'The database has no tables'. The central query editor window contains the command 'INVALIDATE METADATA;'. To the right of the editor, a message box indicates 'Success.' Below the editor, the 'Query History' tab is selected, showing the query 'INVALIDATE METADATA' was run 'a few seconds ago'. The 'Saved Queries' tab is also visible.

4. View the table by giving the query:

```
Select * from <tablename>
```

The screenshot shows the Hue interface for an Impala connection. The top navigation bar includes links for Query Editors, Data Browsers, Workflows, and a user dropdown for 'admin'. The main area has tabs for 'Tables' and 'Assistant'. A search bar at the top says 'Search SQL tables...'. Below it, under 'Tables', there is a section for the 'default' database which shows '(1)' table and a note 'testtable'. The central query editor window contains the command 'Select * from testtable;'. To the right of the editor, a message box indicates '6.35s default text'. Below the editor, the 'Results (21)' tab is selected, showing a table with two columns: 'character' and 'frequency'. The data rows are: 1 l, 2 w, 3 s, 4 e, 5 a, 6 i, 7 y, 8 u. The 'Saved Queries' tab is also visible.

character	frequency
1 l	2
2 w	2
3 s	3
4 e	12
5 a	5
6 i	5
7 y	1
8 u	1

5. You have now successfully run the Impala query using Hue!

3.8. Uploading Roadshow data to ADLS:

Let's make it more interesting, we will now try to get some structured and unstructured data loaded into ADLS.

The structured data like retail information, **e.g.** products, order items, sales etc. The unstructured data like a simple weblog.

1. Download the zipped data by running the below command.

```
 wget https://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/upload-roadshow-data.sh
```

```
[cloudera@cdmstr-ade8c929 ~]$ wget https://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/upload-roadshow-data.sh
--2017-11-23 03:46:08-- http://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/upload-roadshow-data.sh
Resolving raw.githubusercontent.com... 151.101.32.133
Connecting to raw.githubusercontent.com|151.101.32.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3487 (3.4K) [text/plain]
Saving to: "upload-roadshow-data.sh"

100%[=====] 3,487          2017-11-23 03:46:08 (73.4 MB/s) - "upload-roadshow-data.sh" saved [3487/3487]

[cloudera@cdmstr-ade8c929 ~]$
```

2. To give permissions to **upload-roadshow-data.sh** file, run the following commands:

```
dos2unix /home/cloudera/upload-roadshow-data.sh
chmod 755 /home/cloudera/upload-roadshow-data.sh
```

```
[cloudera@cdmstr-ade8c929 ~]$ dos2unix /home/cloudera/upload-roadshow-data.sh
dos2unix: converting file /home/cloudera/upload-roadshow-data.sh to UNIX format ..
[cloudera@cdmstr-ade8c929 ~]$ chmod 755 /home/cloudera/upload-roadshow-data.sh
[cloudera@cdmstr-ade8c929 ~]$
```

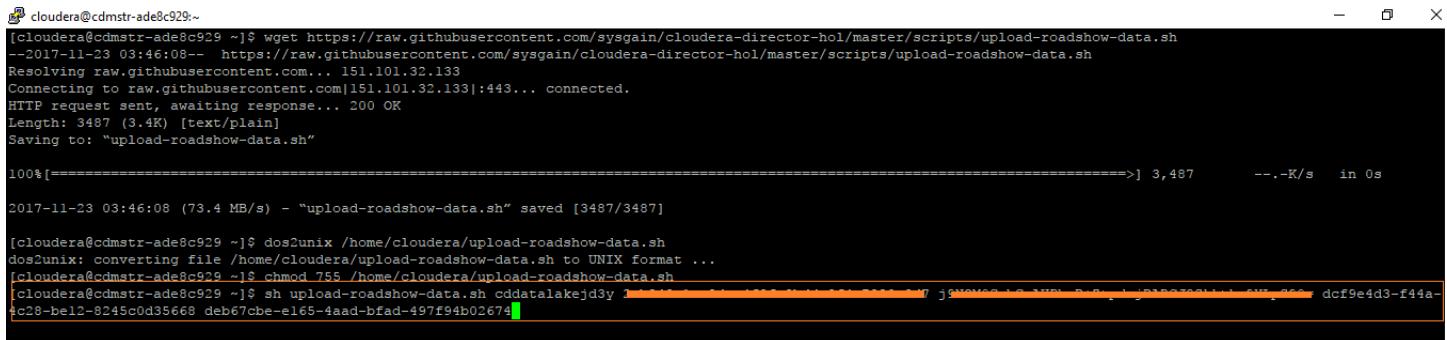
3. Run the following command to execute the **upload-roadshow-data.sh** script:

```
sh upload-roadshow-data.sh <DataLakename> <ClientID> <ClientSecret>
<TenantID> <SubscriptionID>
```

Note: Replace the above values from **NodeDetails**.

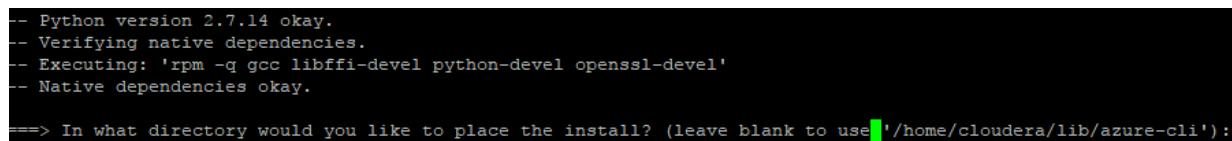
Example:

```
sh upload-roadshow-data.sh cddatalakejd3y 2ab243e1-ABCD-4f9f-ABCD-9f4c7820a3d7  
j9NOM0GubGABDEC GhaB+Ztpd12bdgtJ8Skktko9YLpS90= dcf9e4d3-f44a-ABCD-be12-8CFRc0d35668  
deb67cbe-e165-ABCD-bfad-245CVF4b02674
```



```
cloudera@cdmstr-ade8c929:~$ wget https://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/upload-roadshow-data.sh  
--2017-11-23 03:46:08-- https://raw.githubusercontent.com/sysgain/cloudera-director-hol/master/scripts/upload-roadshow-data.sh  
Resolving raw.githubusercontent.com... 151.101.32.133  
Connecting to raw.githubusercontent.com|151.101.32.133|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 3487 (3.4K) [text/plain]  
Saving to: "upload-roadshow-data.sh"  
  
100%[=====] 3,487 --.-K/s in 0s  
  
2017-11-23 03:46:08 (73.4 MB/s) - "upload-roadshow-data.sh" saved [3487/3487]  
  
[cloudera@cdmstr-ade8c929 ~]$ dos2unix /home/cloudera/upload-roadshow-data.sh  
dos2unix: converting file /home/cloudera/upload-roadshow-data.sh to UNIX format ...  
[cloudera@cdmstr-ade8c929 ~]$ chmod 755 /home/cloudera/upload-roadshow-data.sh  
[cloudera@cdmstr-ade8c929 ~]$ sh upload-roadshow-data.sh cddatalakejd3y 2ab243e1-ABCD-4f9f-ABCD-9f4c7820a3d7 j9NOM0GubGABDEC GhaB+Ztpd12bdgtJ8Skktko9YLpS90= dcf9e4d3-f44a-ABCD-be12-8CFRc0d35668 deb67cbe-e165-ABCD-bfad-245CVF4b02674
```

4. While running the script will prompt for action as in below screen shot.



```
-- Python version 2.7.14 okay.  
-- Verifying native dependencies.  
-- Executing: 'rpm -q gcc libffi-devel python-devel openssl-devel'  
-- Native dependencies okay.  
  
==> In what directory would you like to place the install? (leave blank to use '/home/cloudera/lib/azure-cli'): 
```

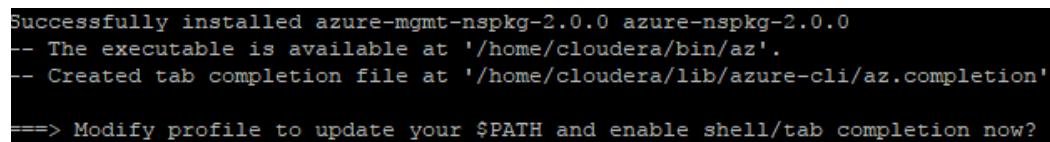
"press **Enter** to proceed"

5. It will prompt with same message again
"press **Enter** to proceed"



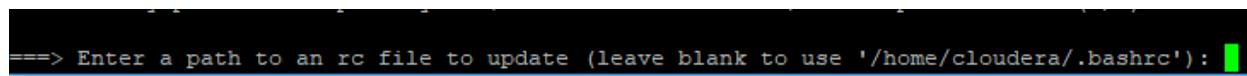
```
-- Creating directory '/home/cloudera/lib/azure-cli'.  
-- We will install at '/home/cloudera/lib/azure-cli'.  
  
==> In what directory would you like to place the 'az' executable? (leave blank to use '/home/cloudera/bin'): 
```

6. Press **Y** (yes) to proceed when prompted for below message.



```
Successfully installed azure-mgmt-nspkg-2.0.0 azure-nspkg-2.0.0  
-- The executable is available at '/home/cloudera/bin/az'.  
-- Created tab completion file at '/home/cloudera/lib/azure-cli/az.completion'  
  
==> Modify profile to update your $PATH and enable shell/tab completion now? (Y/n): 
```

7. Press **Enter** to proceed further when prompted for the below message.



```
==> Enter a path to an rc file to update (leave blank to use '/home/cloudera/.bashrc'): 
```

8. Now the roadshow data is uploaded to ADLS.

```
--2017-11-23 05:33:01-- https://aztdrepo.blob.core.windows.net/clouderadirator/roadshow.zip
Resolving aztdrepo.blob.core.windows.net... 52.238.56.168
Connecting to aztdrepo.blob.core.windows.net|52.238.56.168|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4711274 (4.5M) [application/x-zip-compressed]
Saving to: "/home/cloudera/roadshow.zip"

100%[=====]
2017-11-23 05:33:04 (1.94 MB/s) - "/home/cloudera/roadshow.zip" saved [4711274/4711274]

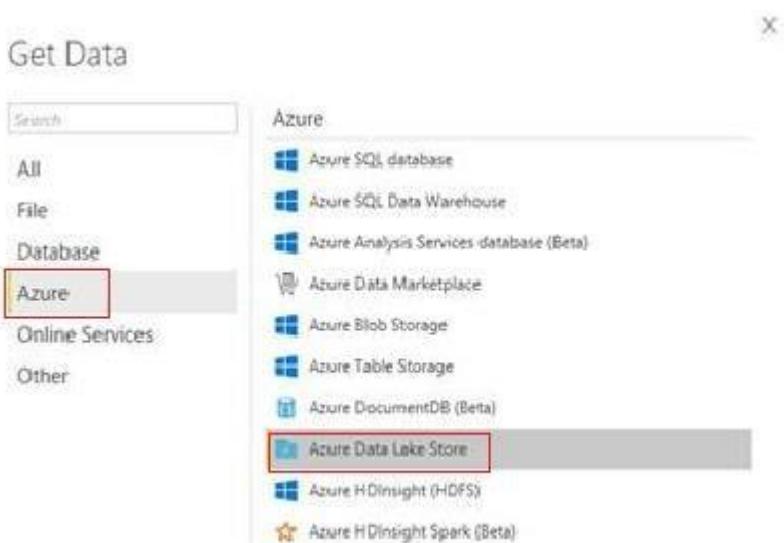
Finished[########################################] 100.0000%
[cloudera@cdmstr-ade8c929 ~]$
```

9. Now you can run queries in hue.

4. Power BI integration with Data Lake Store and Impala (Optional)

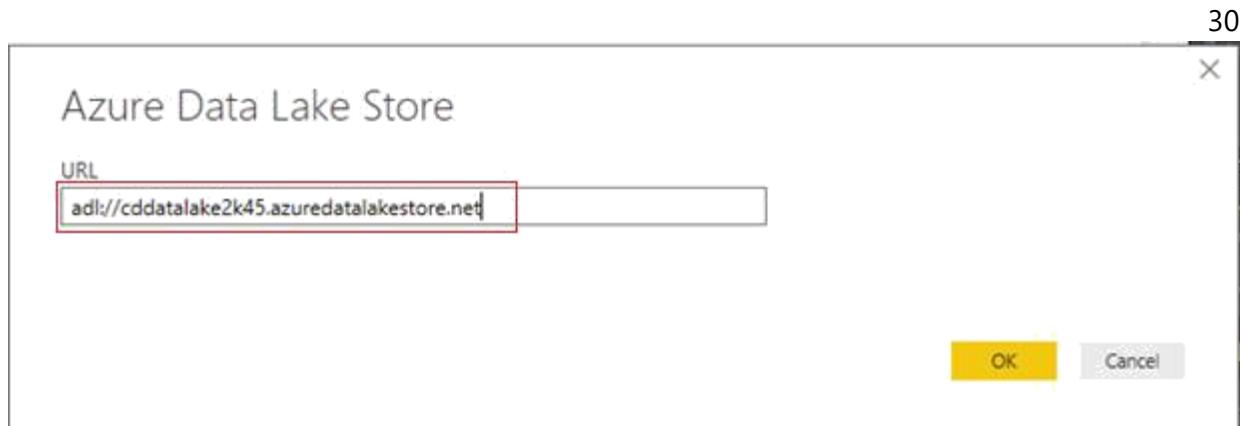
4.1. Integrating with Data Lake Store

1. Launch **Power BI Desktop** on your computer.
2. From the Home ribbon, click Get Data, and then click More. In the Get Data dialog box, click Azure, click Azure Data Lake Store, and then click Connect.



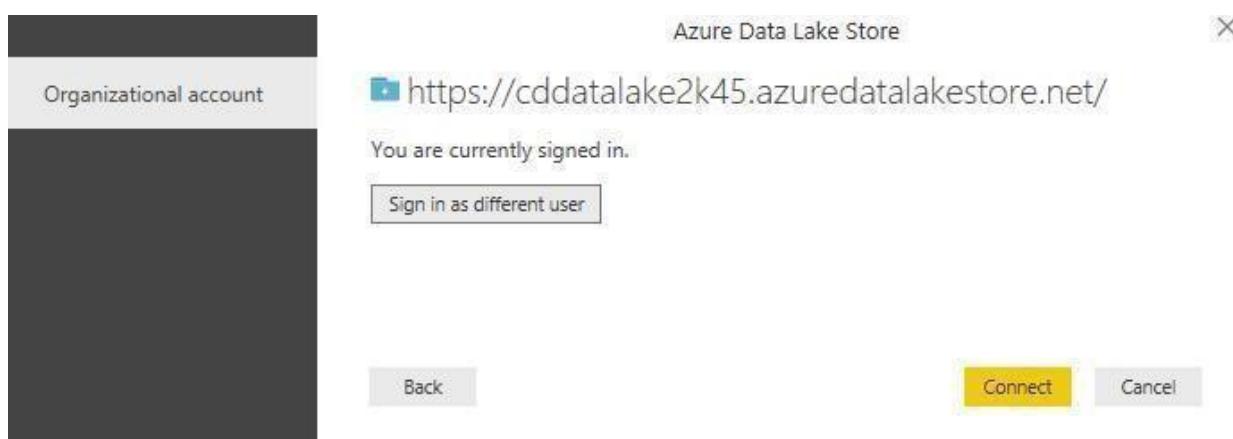
3. In the Microsoft Azure Data Lake Store dialog box, provide the **URL to your Data Lake Store account**, and then click **OK**.

Note: Get the **URL - Datalake Endpoint** from the NodeDetails file. (Refer to section **4.1**)



4. In the next dialog box, click **Sign in** to sign into Data Lake Store account. You will be redirected to your organization's sign in page. **Follow the prompts** to sign into the account.

After you have successfully signed in, click **Connect**.



5. The next dialog box shows the file that you uploaded to your Data Lake Store account. **Verify** the info and then click **Load**.

The screenshot shows two windows side-by-side. The top window is a browser window titled 'Untitled - Power BI Desktop' displaying a table with one row of data from 'adl://cddatalake2k45.azuredatalakestore.net/'. The bottom window is the 'Power BI Desktop' application itself, showing the same table in the 'Fields' tab of the 'Modeling' ribbon.

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Table	demotd2k45		7/5/2017 11:27:43 AM +00:00	7/5/2017 11:28:15 AM +00:00	null	Record	https://cddatalake2k45.a...

Power BI Desktop - Untitled - Modeling

Clipboard External data Resources Insert Relationships Calculations Share

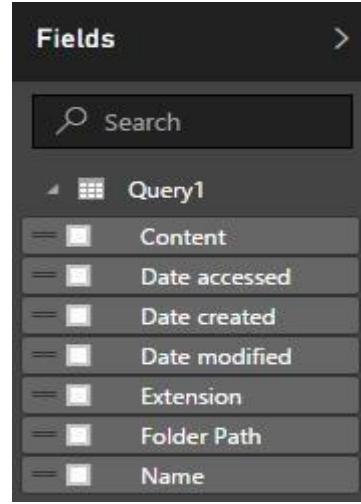
Fields >

Search

Query1

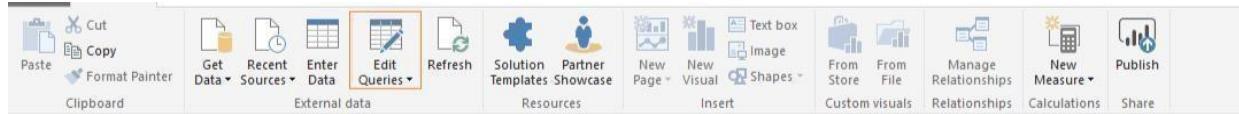
- Content
- Date accessed
- Date created
- Date modified
- Extension
- Folder Path
- Name

- After the data has been successfully loaded into Power BI, you will see the available fields in the **Fields** tab.



7. However, to visualize and analyze the data, you might prefer the data be available as per your requirements. To do so, follow the steps below:

8. Select **Edit Query** from the top menu bar:



Under the content column, right click on **Table** and select **Add as New Query**, you will see a new query added in the queries column:

The screenshot shows the Power BI Query Editor interface. The ribbon at the top has tabs for File, Home, Transform, Add Column, and View. The Home tab is selected. The toolbar below the ribbon contains icons for Close & Apply, New Source, Recent Sources, Data source settings, Manage Parameters, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Split Column, Group By, Replace Values, Data Type Any, Use First Row As Headers, Merge Queries, Append Queries, Combine Files, and Continue. A 'Queries [1]' pane on the left lists 'Query1'. The main area displays a table with one row: 'Value' and 'demodd545'. The 'Attributes' pane on the right shows 'Name: Query1' and 'Applied Steps: Source'. The status bar at the bottom indicates '1 row found'.

This screenshot is similar to the first one, showing the Power BI Query Editor with 'Query1' selected. However, a context menu is open over the table, with the 'Add as New Query' option highlighted by a red box. The rest of the menu options are 'Copy' and 'Drill Down'.

9. Once again, **right click** and select **Add as New Query** to convert the table content to binary form.

The screenshot shows the Azure Data Lake Storage Explorer Query Editor interface. On the left, there's a sidebar titled 'Queries [2]' containing 'Query1' and 'demotd2k45'. The main area displays a table with the following data:

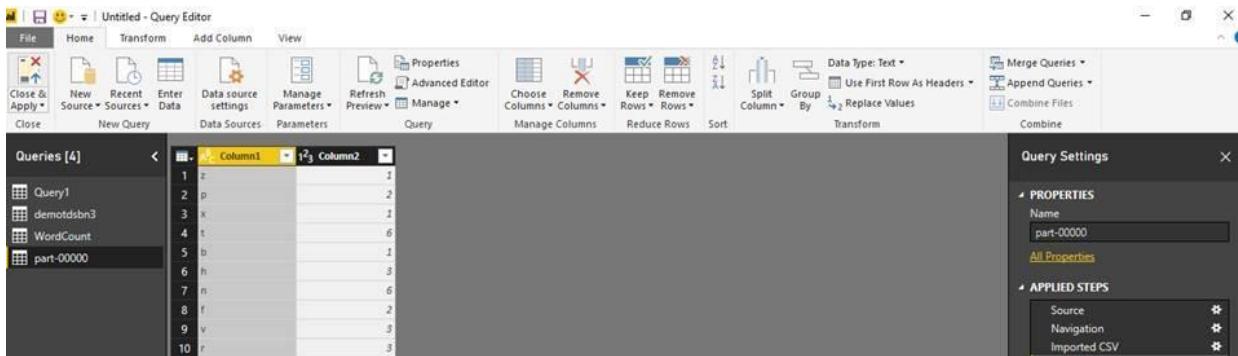
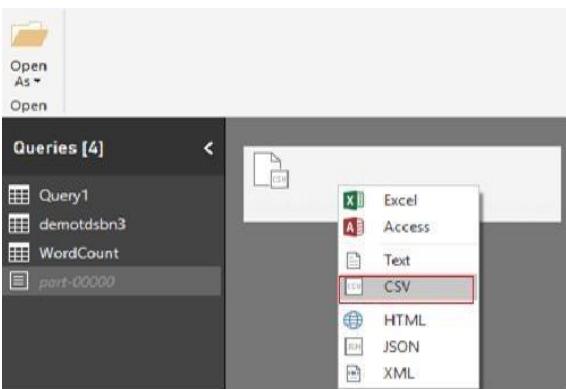
Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	_SUCCESS		7/5/2017 11:28:19 AM +00:00	7/5/2017 11:28:19 AM +00:00	null	Record	https://cdatalake2k45
Binary	part-00000		7/5/2017 11:28:18 AM +00:00	7/5/2017 11:28:18 AM +00:00	null	Record	https://cdatalake2k45
Binary	part-00001		7/5/2017 11:28:17 AM +00:00	7/5/2017 11:28:17 AM +00:00	null	Record	https://cdatalake2k45

The 'Query Settings' pane on the right shows the properties for 'demotd2k45', including 'Name' and 'All Properties'. The 'Applied Steps' pane shows a single step named 'Navigation'.

10. Right click and create a new query to get the data from the table as shown below:

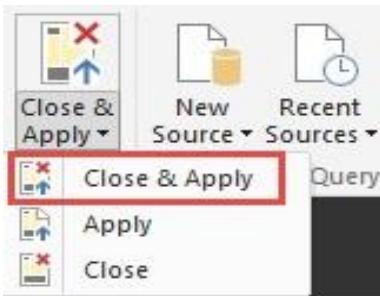
This screenshot is identical to the one above, showing the Azure Data Lake Storage Explorer Query Editor with the 'WordCount' table and its data. The 'Query Settings' and 'Applied Steps' panes are also visible on the right.

11. You will see a file icon that represents the file that you uploaded. Right-click the file, and click CSV.

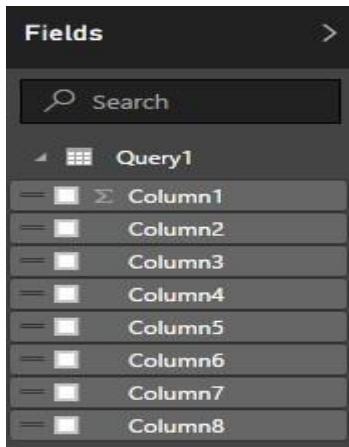


12. Your data is now available in a format that you can use to create visualizations.

13. From the **Home** ribbon, click **Close and Apply**, and then click **Close and Apply**.

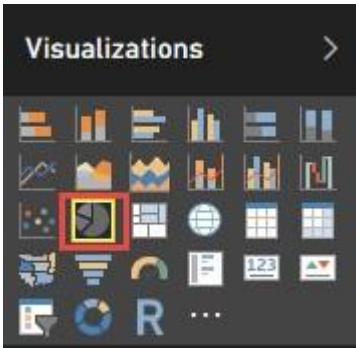


14. Once the query is updated, the **Fields** tab will show the new fields available for visualization.

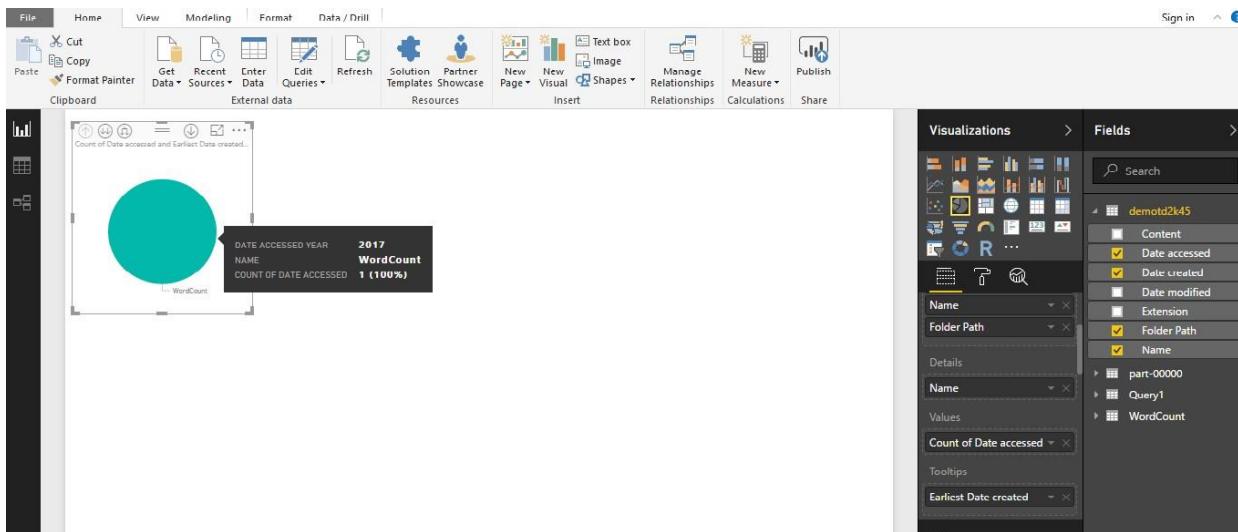


15. You can create a pie chart to represent your data. To do so, make the following selections:

- a) From the **Visualizations** tab, click the symbol for a **pie chart** (see below).



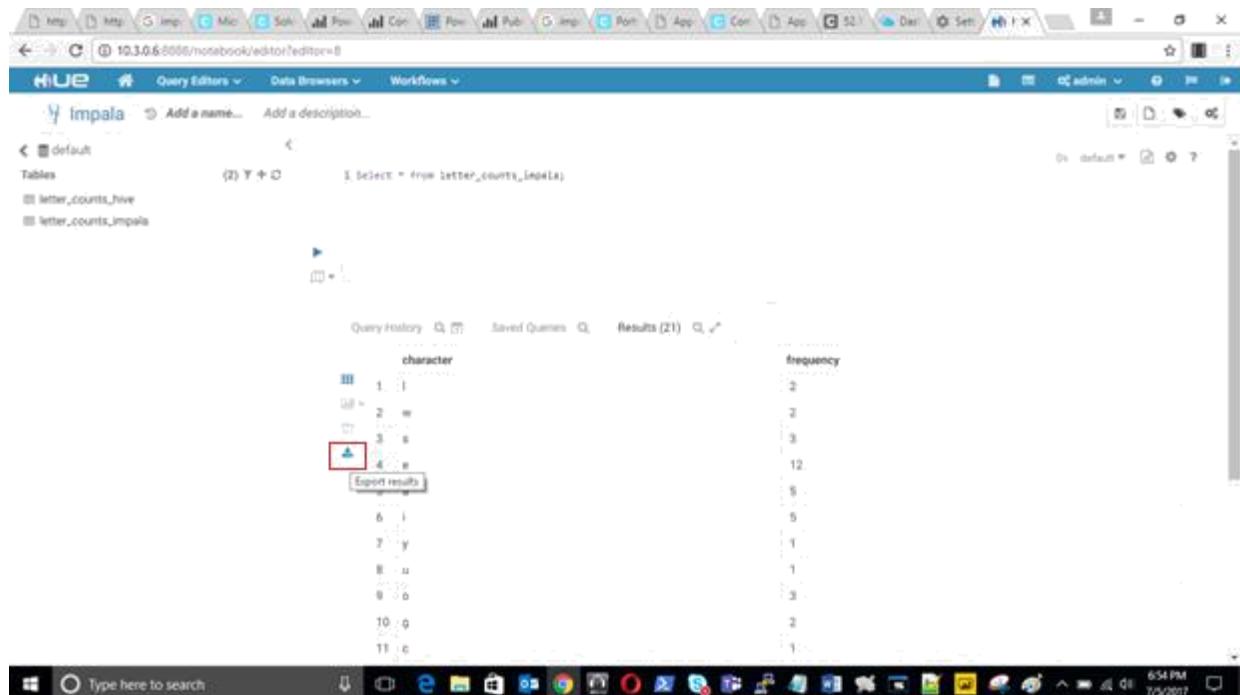
- b) Drag the columns that you want to use and represent in your pie-chart from the **Fields** tab to **Visualizations** tab, as shown below:



16. From the **file** menu, click **Save** to save the visualization as a Power BI Desktop file.

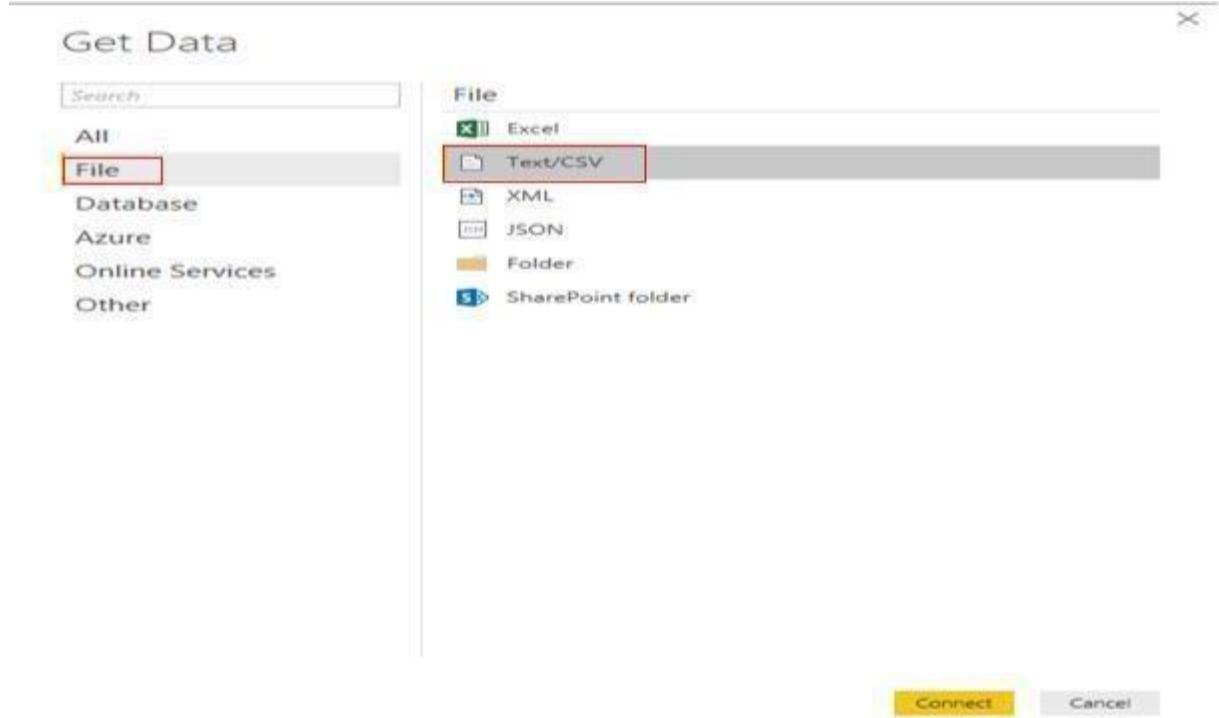
4.2. Integrating with Impala

1. Go to **point 7** of section **4.7**, where you ran a query from the table created using the output from ADLS copied to local HDFS.



2. Click the **Export Results** button in the Hue Impala UI, as seen in the above screenshot, to download the output as a **CSV** file.

3. From the **Home** ribbon in Power BI, click **Get Data**, and then click **More**. In the **Get Data** dialog box, click **File**, click **Text/CSV**, and then click **Connect**.



4. Select the **CSV** file exported from Impala in **Step 2** and click on **Open**.

character	frequency
I	2
w	2
s	3
e	12
a	5
i	5
y	2
u	2
o	5
g	2
c	1
z	1
p	2
x	1
t	6
b	1
h	3
n	6
r	2
v	3

The data in the preview has been truncated due to size limits.

Load Edit Cancel

5. Click on **Load**.
6. Select the **Data** button to visualize the content.

The screenshot shows the Power BI desktop interface with a table visualization titled "Untitled - PI". The table has two columns: "character" and "frequency". The data consists of 21 rows, each representing a character and its frequency. The characters listed are I, W, S, E, A, I, Y, U, O, G, C, Z, P, X, T, B, H, N, F, V, and R. The frequencies are 2, 2, 3, 12, 5, 5, 1, 1, 3, 2, 1, 2, 1, 1, 6, 1, 3, 6, 2, 3, and 3 respectively. The table is displayed in a dark-themed environment.

character	frequency
I	2
W	2
S	3
E	12
A	5
I	5
Y	1
U	1
O	3
G	2
C	1
Z	1
P	2
X	1
T	6
B	1
H	3
N	6
F	2
V	3
R	3

TABLE: query-impala-7 (21 rows)

You have successfully visualized the content exported from impala using power BI.

5. Reference

5.1. Restart Cloudera Management Service

You may need to restart Cloudera Management Service for the below errors:

Error:

- Request to the Service Monitor failed. This may cause slow page responses. [View the status of the Service Monitor.](#)
- Request to the Host Monitor failed. This may cause slow page responses. [View the status of the Host Monitor.](#)

The screenshot shows the Cloudera Manager Home page at the URL 10.3.0.5:7180/cmf/home. The page is managed by Cloudera Director. At the top, there are navigation links for Clusters, Hosts, Diagnostics, Audits, Charts, Backup, and Administration. On the right, there are icons for Help, Logout, Search, and Support. Below the header, a banner displays two error messages: "Request to the Service Monitor failed. This may cause slow page responses. View the status of the Service Monitor." and "Request to the Host Monitor failed. This may cause slow page responses. View the status of the Host Monitor." The main content area has tabs for Status, All Health Issues, Configuration, and All Recent Commands. On the left, there is a sidebar titled "Director_Azure.... (CDH 5.12.0, Parcels)" which lists various hosts: HBASE-1, HDFS-1, HIVE-1, HUE-1, IMPALA-1, OOZIE-1, SPARK_ON..., YARN-1, and ZOOKEEPER-1. Each host entry includes a status icon and a dropdown menu. To the right, there is a "Charts" section with three panels: "Cluster CPU" (status: QUERY ERROR), "Cluster Disk IO" (status: QUERY ERROR), and "Cluster Network IO". Above the charts, a time range selector shows options for 30m, 1h, 2h, 6h, 12h, 1d, 7d, and 30d. A message at the bottom right encourages activating Windows: "Activate Windows Go to Settings to activate Win".

1. Go to <http://<Manager Node FQDN>:7180/cmf/home>.
2. Go to **Cloudera Management Service** and select **MGMT**.

The screenshot shows the Cloudera Manager interface. On the left, there's a navigation tree with a node expanded to show 'Hosts'. Under 'Hosts', several services are listed with their status indicators (green, yellow, red) and dropdown menus:

- HBASE-1
- HDFS-1
- HIVE-1
- HUE-1
- IMPALA-1
- OOZIE-1
- SPARK_ON... (partially visible)
- YARN-1
- ZOOKEEPER-1

Below this is a section titled 'Cloudera Management Service' containing a single item:

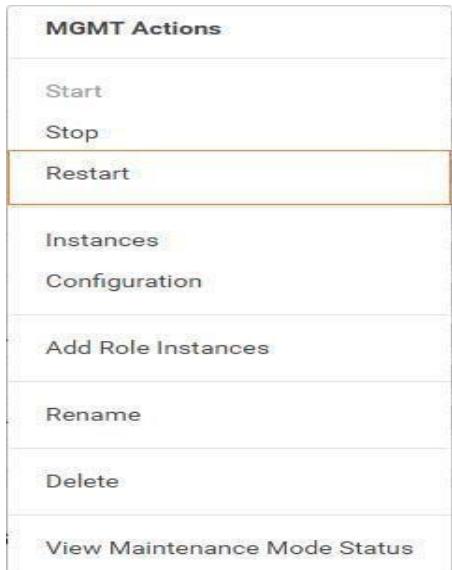
- MGMT

On the right side, there's a 'Charts' panel with a single entry:

- Internal error while querying the Host Monitor

Under the 'Charts' panel, there are two sections: 'Cluster CPU' and 'Cluster Network IO', both of which display the message 'QUERY ERROR'.

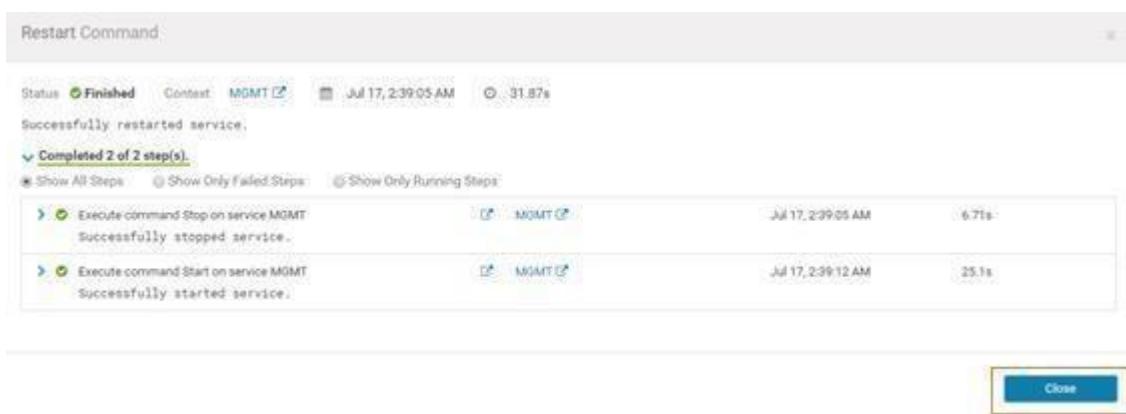
3. Click on the drop down menu and select **Restart**.



4. Confirm by clicking the **Restart** button.



5. Click on **Close** to complete the process.



Note: If you performed this restart in response to errors, please now re-run section **4.3** after performing the above steps.

5.2. Error Messages While Running the Spark Job

1. You may see a few errors popping up while executing the Spark job that can safely be ignored, such as the ones below.

Note: The permissions get properly set in the .sh file.

```
sh ClouderaSparkSetup.sh demotdweti 10.3.0.6 mkdir: Permission denied: user=cloudera,  
access=WRITE, inode="/":hdfs:supergroup:drwxr-xr-x --  
2017-07-11 16:55:54-- https://aztdrepo.blob.core.windows.net/clouderadirector/wordcount.jar  
Resolving aztdrepo.blob.core.windows.net... 52.238.56.168  
Connecting to aztdrepo.blob.core.windows.net|52.238.56.168|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 6371588 (6.1M) [application/octet-stream] Saving  
to: "/home/cloudera/wordcount.jar"
```

2. Searching for Cloudera Navigator – this error can safely be ignored.

```
INFO scheduler.DAGScheduler: Job 1 finished: saveAsTextFile at SparkWordCount.scala:32, took  
1.811055 s
```

```
INFO spark.SparkContext: Invoking stop() from shutdown hook  
ERROR scheduler.LiveListenerBus: Listener ClouderaNavigatorListener threw an exception  
java.io.FileNotFoundException: Lineage is enabled but lineage directory  
/var/log/spark/lineage doesn't exist  
at
```

```
com.cloudera.spark.lineage.ClouderaNavigatorListener.checkLineageEnabled(ClouderaNavigatorListener.scala:122) at com.cloudera.spark.lineage.
```

Note: You may refer to the **Spark** section of the **Cloudera release notes** for further details (link below).

https://www.cloudera.com/documentation/enterprise/releasenotes/topics/cn_rn_known_issues.html