Clickpath Analytics

By

Satish Vittalam

Contents

Web Server Log – Clickpath Analytics Summary 3

Problem Statement: 3

Overview of technology: 3

High Level Steps: 3

Data Source: 3

Hardware/OS: 3

Software used: 3

Installation of Linux based Hadoop Cluster using Azure Portal 4

Installation of SQL Server: 12

Data used for Visualizing Website Clickstream Data: 13

File upload through the file view in Ambari Server UI: 15

Data processing in Hadoop HDFS system: 16

Visualization and Analytics: 22

Analytics using SparkSQL with Zepplin: 22

Visualization using Zepplin: 23

Invoking Zepplin through the Ambari UI: 23

Interpretor Binding: 23

Zepplin Visualization 1: 24

Zepplin Visualization 2: 24

Zepplin Visualization 3: 25

Zepplin Visualization 4: 25

Visualization with PowerBI: 26

Setting up the DSN from control panel to for Hadoop HIVE: 26

Steps to import data into PowerBI: 27

PowerBI Visualization 1: 29

PowerBI Visualization 2: 29

Web Server Log – Clickpath Analytics Summary

Problem Statement:

Ecommerce retailers and other companies who have an online presence are trying gather more details about the customers browsing or online shopping patterns, the products they buy, the products they may be interested in the future and also provide a better shopping experience. They perform Basket Analysis, Path optimization and even try to analyze the next product to buy.To achieve this, companies have to process massive amounts of data sets in terms of web server logs which is also referred to as Clickpath or Clickstream data. This information is captured by Webserver as customer navigates around the website.

Overview of technology:

Azure HDInsight offers a cost-effective way to process massive amounts of data. Hadoops framework and its ecosystem helps to analyze this information easier, get better insights about the customer and help improve the effectiveness of the shopping. We use the tools and technologies provided to process large datasets of log files, get the required information from the logs and combine them with user profile and products data (these could be available from the OLTP application) to perform the required analytics. Hadoop offers multiple analytics tools for these big datasets. We will load, refine and visualize the log data.

High Level Steps:

1) Created a HortonWorks Cluster available on Azure and configure it for SSH
2) Pre-process the Web logs sample data that is obtained
3) Create a SQL database in Azure
4) Move the sample data into Hadoop File system
5) Use Pig latin script to combine the web server logs into one.
6) Use Sqoop to move the data from traditional RDBMS (OLTP system) to Hive
7) Create custom Hive tables that will store the final data that is created for visualization.
8) Use Zepplin and Power BI for Analytics and visualization

Data Source:

https://s3.amazonaws.com/hw-sandbox/tutorial8/RefineDemoData.zip

Hardware/OS:

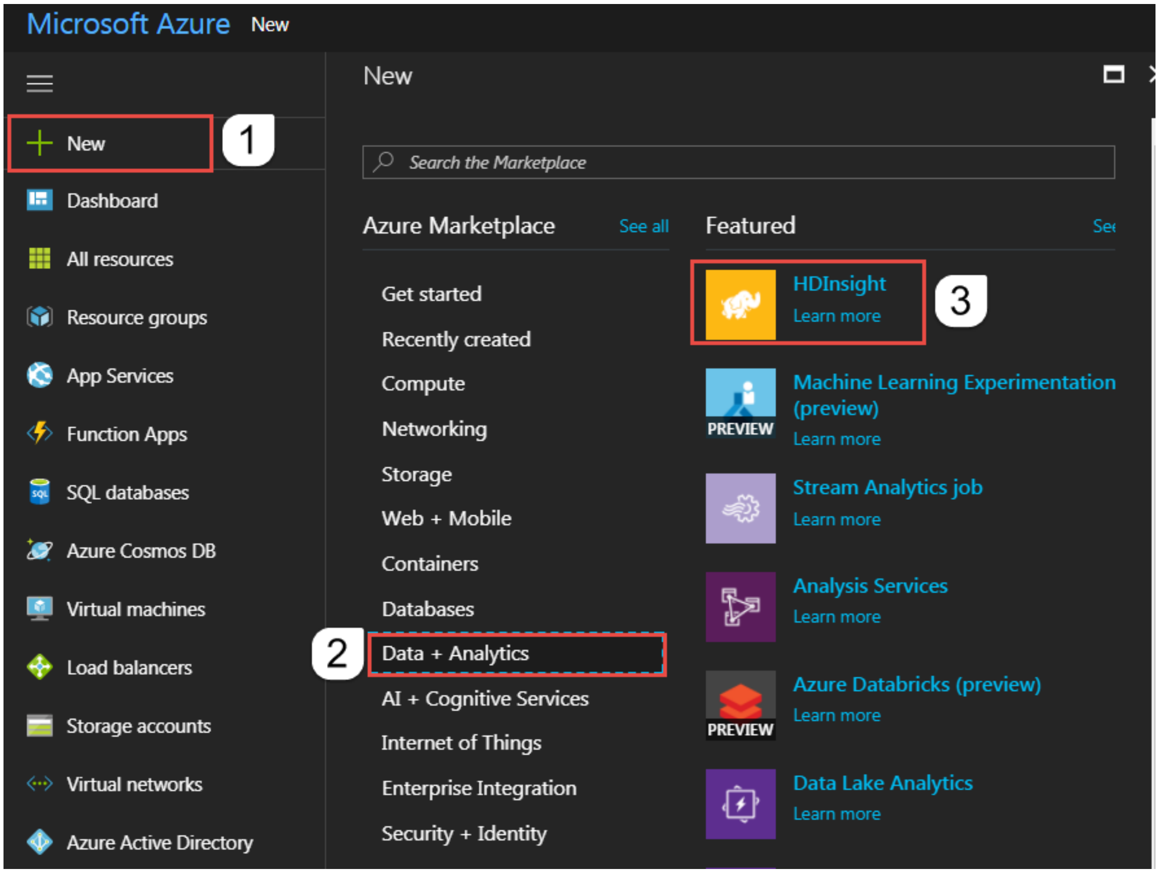
Intel Core i5-5300U CPU 2.30 GHZ, 16 GB RAM, 64 bit Windows 7 operating system

Software used:

SQL Server Management studio, PowerBI, Powerview

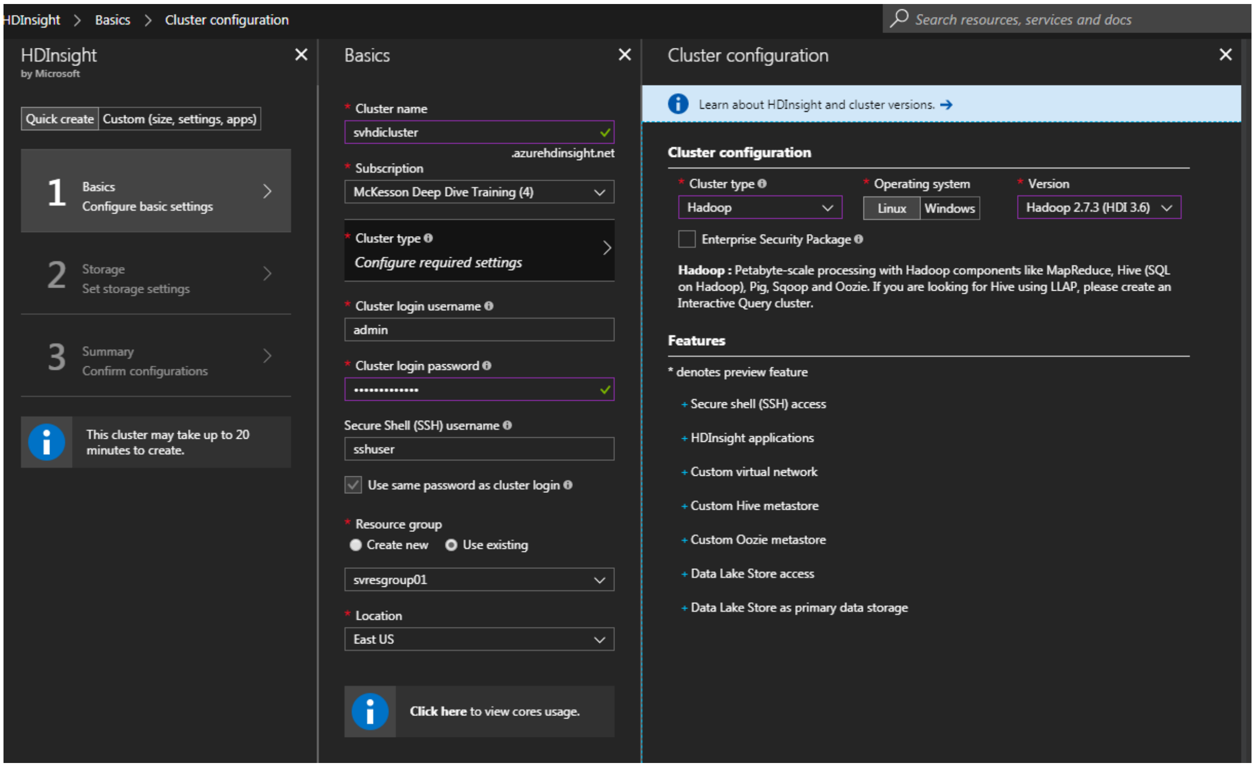
Installation of Linux based Hadoop Cluster using Azure Portal

In this step, we begin with using Azure Portal and creating a Hadoop Cluster using Linux.



The basics screen talks about providing the following:

- Clustername
- Cluster Type
- Cluster login name
- Password
- Username used for SSH login
- Resource group needed for creation
- Location.



This cluster may take up to 20 minutes to create.

* Cluster name: svhdicluster .azurehdinsight.net

* Subscription: McKesson Deep Dive Training (4)

* Cluster type: *Configure required settings*

* Cluster login username: admin

* Cluster login password: •••••••••••••

Secure Shell (SSH) username: sshuser

☑ Use same password as cluster login

* Resource group: ○ Create new ● Use existing

svresgroup01

* Location: East US

Click here to view cores usage.

Learn about HDInsight and cluster versions. →

Cluster configuration

* Cluster type: Hadoop

* Operating system: Linux | Windows

* Version: Hadoop 2.7.3 (HDI 3.6)

☐ Enterprise Security Package

Hadoop : Petabyte-scale processing with Hadoop components like MapReduce, Hive (SQL on Hadoop), Pig, Sqoop and Oozie. If you are looking for Hive using LLAP, please create an Interactive Query cluster.

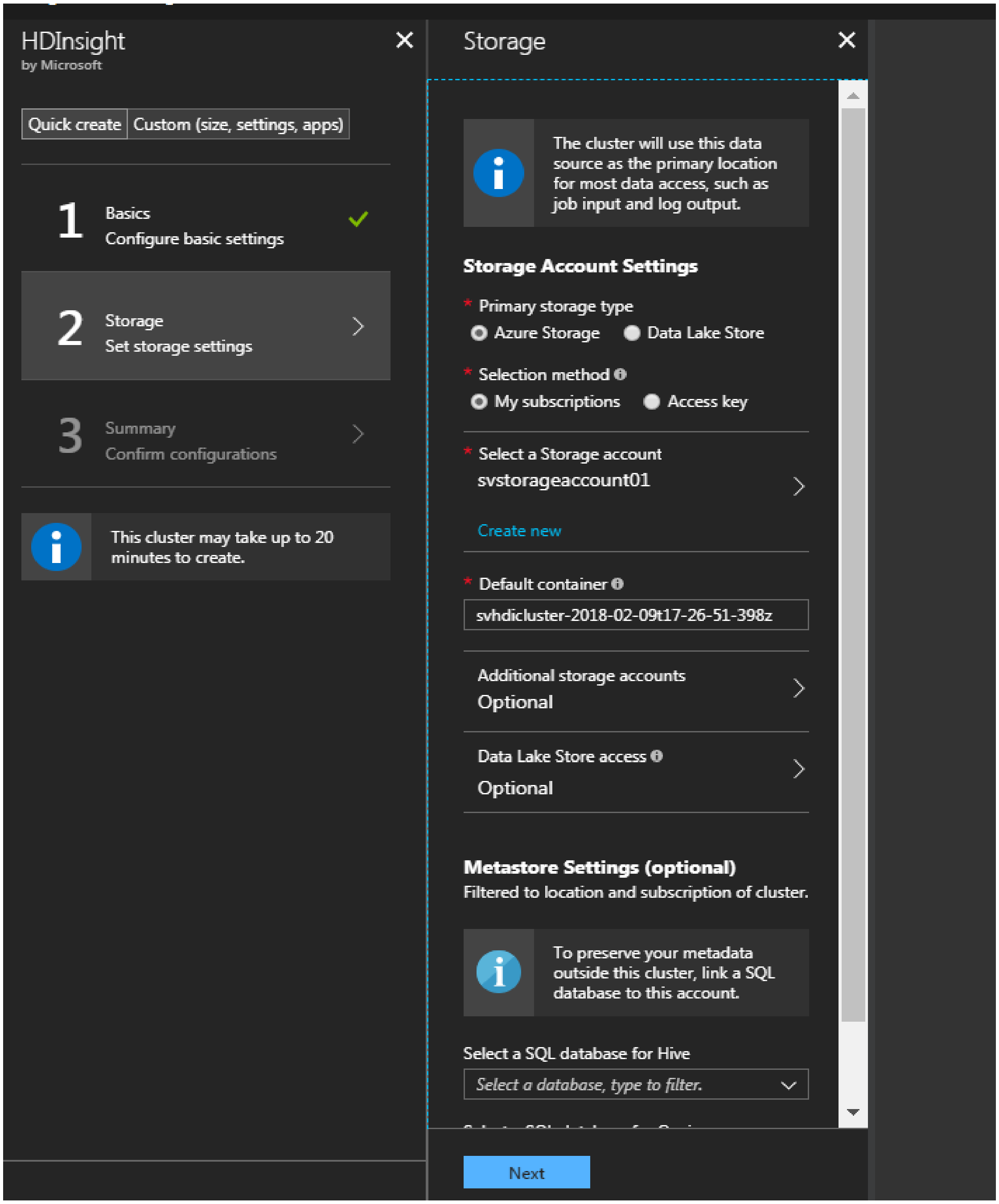
Features

* denotes preview feature

- + Secure shell (SSH) access
- + HDInsight applications
- + Custom virtual network
- + Custom Hive metastore
- + Custom Oozie metastore
- + Data Lake Store access
- + Data Lake Store as primary data storage

The storage section talks about the following:

- Selecting the storage type – Data lake or Azure storage
- Storage Account name



HDInsight
by Microsoft

1 Basics
Configure basic settings ✓

2 Storage
Set storage settings

3 Summary
Confirm configurations

This cluster may take up to 20 minutes to create.

Storage

The cluster will use this data source as the primary location for most data access, such as job input and log output.

Storage Account Settings

* Primary storage type
● Azure Storage ○ Data Lake Store

* Selection method
● My subscriptions ○ Access key

* Select a Storage account
svstorageaccount01

Create new

* Default container
svhdicluster-2018-02-09t17-26-51-398z

Additional storage accounts
Optional

Data Lake Store access
Optional

Metastore Settings (optional)
Filtered to location and subscription of cluster.

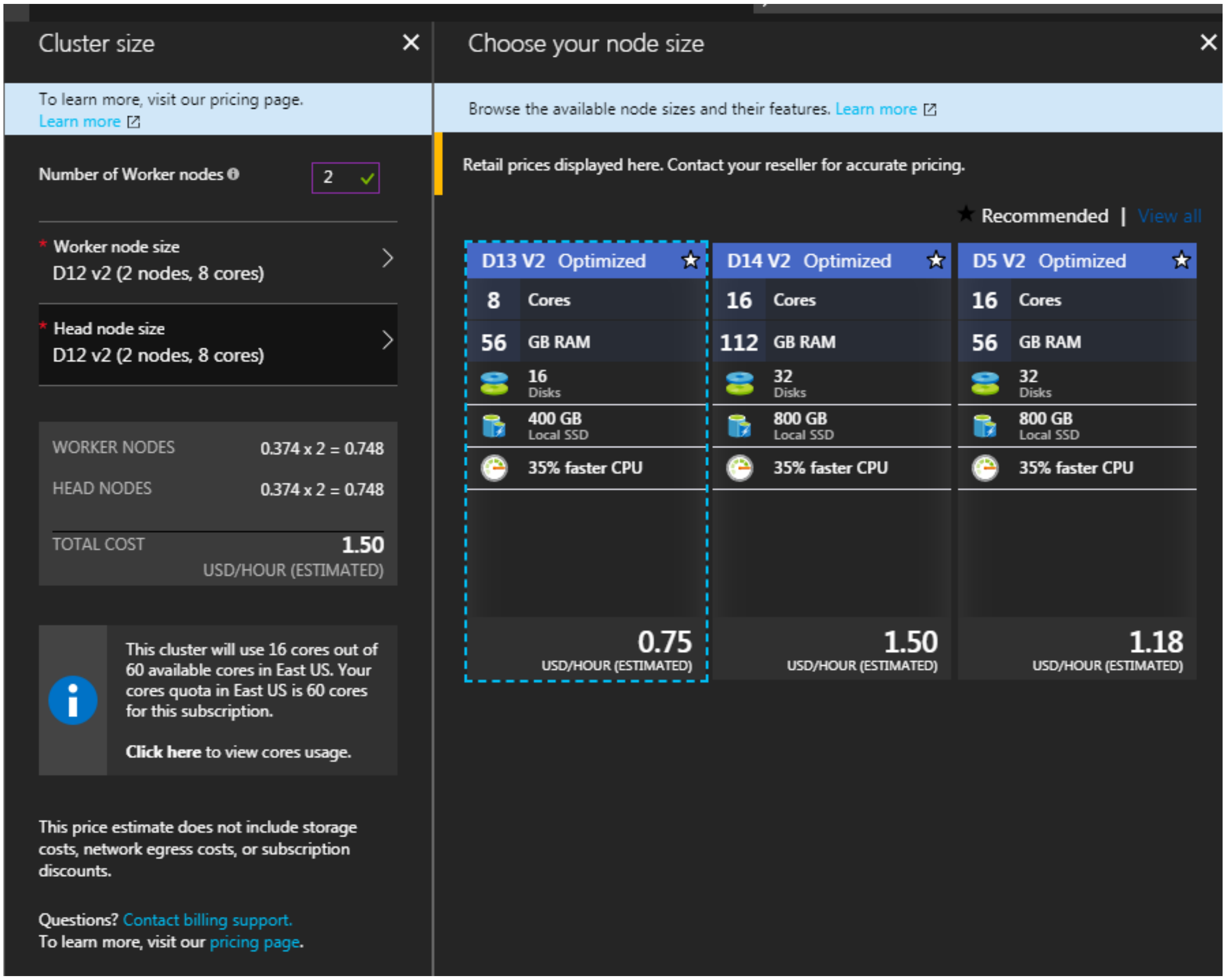
To preserve your metadata outside this cluster, link a SQL database to this account.

Select a SQL database for Hive
Select a database, type to filter.

In this screen we define the following:

- Number of Worker nodes
- Worker Node Size
- Head Node Size

This screen also shows the cost associated with each selection



Cluster size ✕

To learn more, visit our pricing page.
Learn more

Number of Worker nodes: 2 ✓

* Worker node size
D12 v2 (2 nodes, 8 cores)

* Head node size
D12 v2 (2 nodes, 8 cores)

WORKER NODES	0.374 x 2 = 0.748
HEAD NODES	0.374 x 2 = 0.748
TOTAL COST	1.50 USD/HOUR (ESTIMATED)

This cluster will use 16 cores out of 60 available cores in East US. Your cores quota in East US is 60 cores for this subscription.

Click here to view cores usage.

This price estimate does not include storage costs, network egress costs, or subscription discounts.

Questions? Contact billing support.
To learn more, visit our pricing page.

Choose your node size ✕

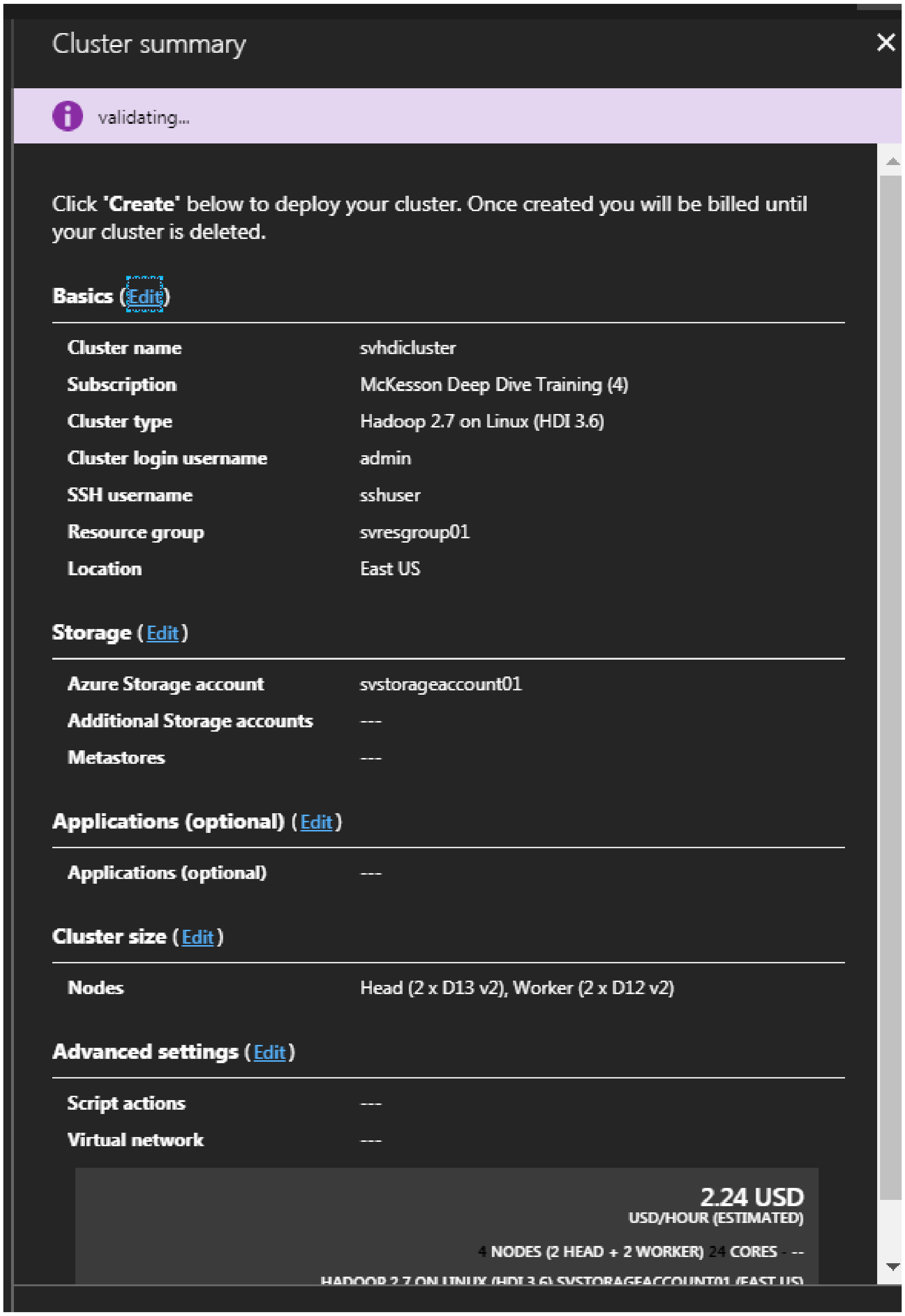
Browse the available node sizes and their features. Learn more

Retail prices displayed here. Contact your reseller for accurate pricing.

★ Recommended | View all

D13 V2 Optimized ☆	D14 V2 Optimized ☆	D5 V2 Optimized ☆
8 Cores	16 Cores	16 Cores
56 GB RAM	112 GB RAM	56 GB RAM
16 Disks	32 Disks	32 Disks
400 GB Local SSD	800 GB Local SSD	800 GB Local SSD
35% faster CPU	35% faster CPU	35% faster CPU
0.75 USD/HOUR (ESTIMATED)	1.50 USD/HOUR (ESTIMATED)	1.18 USD/HOUR (ESTIMATED)

The below screen shows the final settings selected for the cluster creation along with the cost associated for spinning up the cluster.



Click **'Create'** below to deploy your cluster. Once created you will be billed until your cluster is deleted.

Basics (Edit)

Cluster name	svhdicluster
Subscription	McKesson Deep Dive Training (4)
Cluster type	Hadoop 2.7 on Linux (HDI 3.6)
Cluster login username	admin
SSH username	sshuser
Resource group	svresgroup01
Location	East US

Storage (Edit)

Azure Storage account	svstorageaccount01
Additional Storage accounts	---
Metastores	---

Applications (optional) (Edit)

Applications (optional)	---

Cluster size (Edit)

Nodes	Head (2 x D13 v2), Worker (2 x D12 v2)

Advanced settings (Edit)

Script actions	---
Virtual network	---

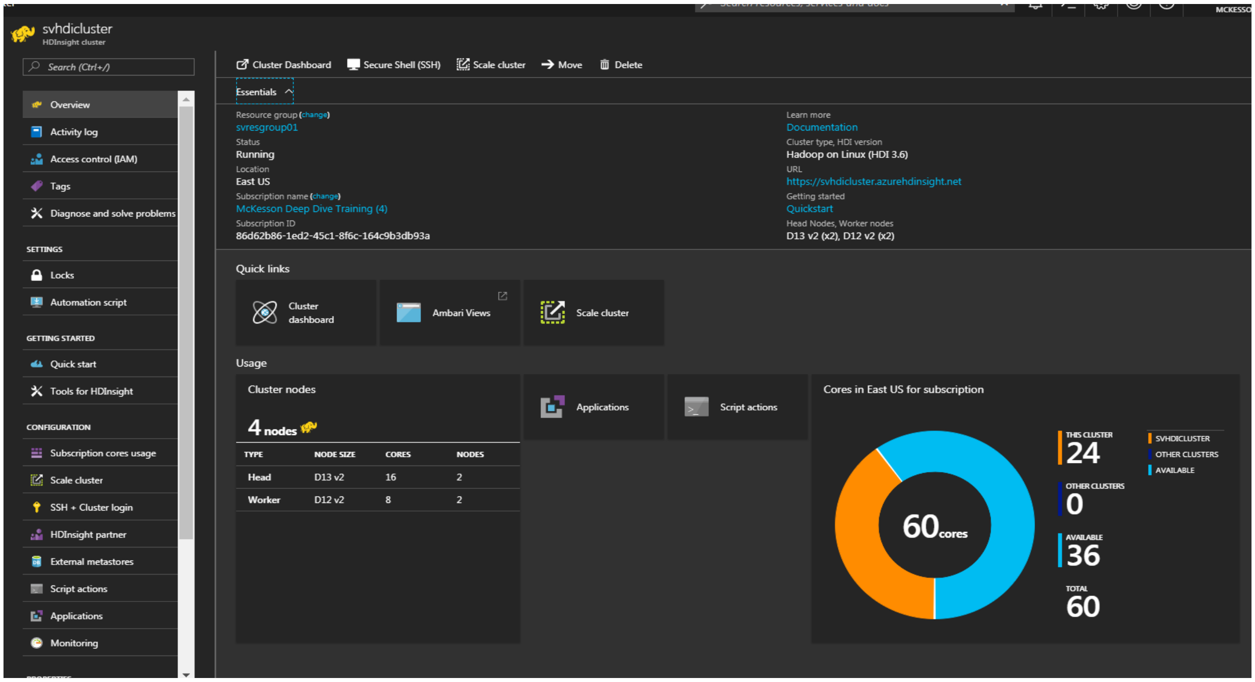
2.24 USD
USD/HOUR (ESTIMATED)

4 NODES (2 HEAD + 2 WORKER) 24 CORES - --

HADOOP 2.7 ON LINUX (HDI 3.6) SVSTORAGEACCOUNT01 (EAST US)

The following would be the cluster URL→ https://svhdicluster.azurehdinsight.net/

This screen shows the final settings of the cluster after deployment.



Resource group (change)
svresgroup01

Status
Running

Location
East US

Subscription name (change)
McKesson Deep Dive Training (4)

Subscription ID
86d62b86-1ed2-45c1-8f6c-164c9b3db93a

Learn more
Documentation

Cluster type, HDI version
Hadoop on Linux (HDI 3.6)

URL
https://svhdicluster.azurehdinsight.net

Getting started
Quickstart

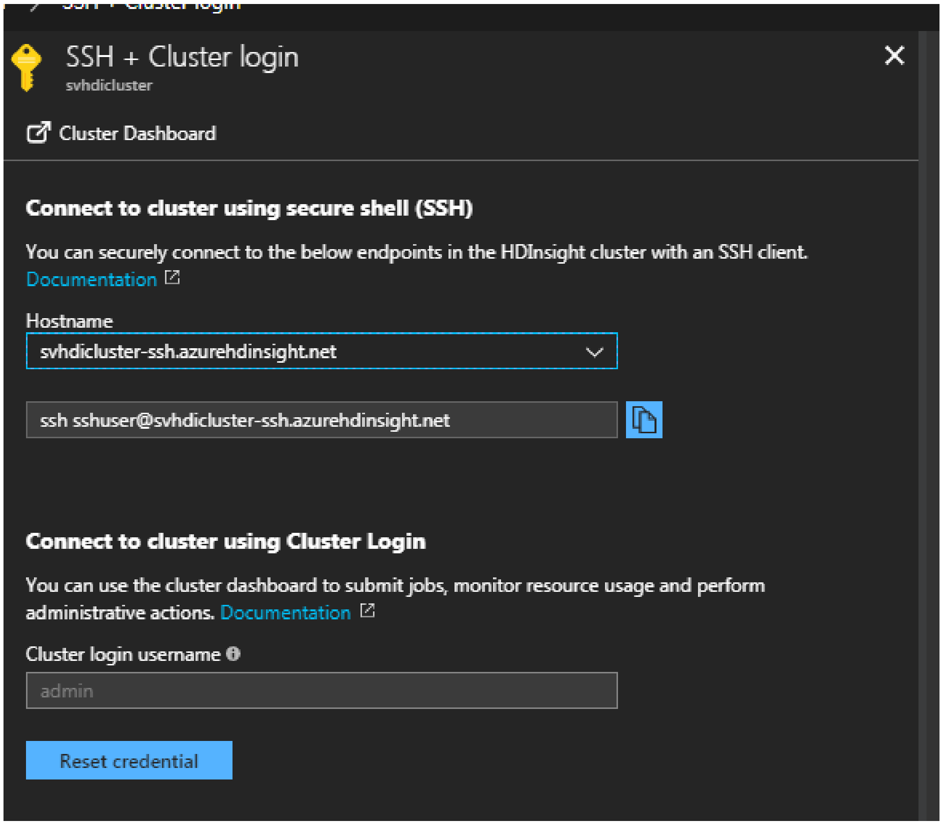
Head Nodes, Worker nodes
D13 v2 (x2), D12 v2 (x2)

Cluster nodes

4 nodes

TYPE	NODE SIZE	CORES	NODES
Head	D13 v2	16	2
Worker	D12 v2	8	2

The following screen shows the connection details for SSH login:



Connect to cluster using secure shell (SSH)

You can securely connect to the below endpoints in the HDInsight cluster with an SSH client. Documentation

Hostname
svhdicluster-ssh.azurehdinsight.net

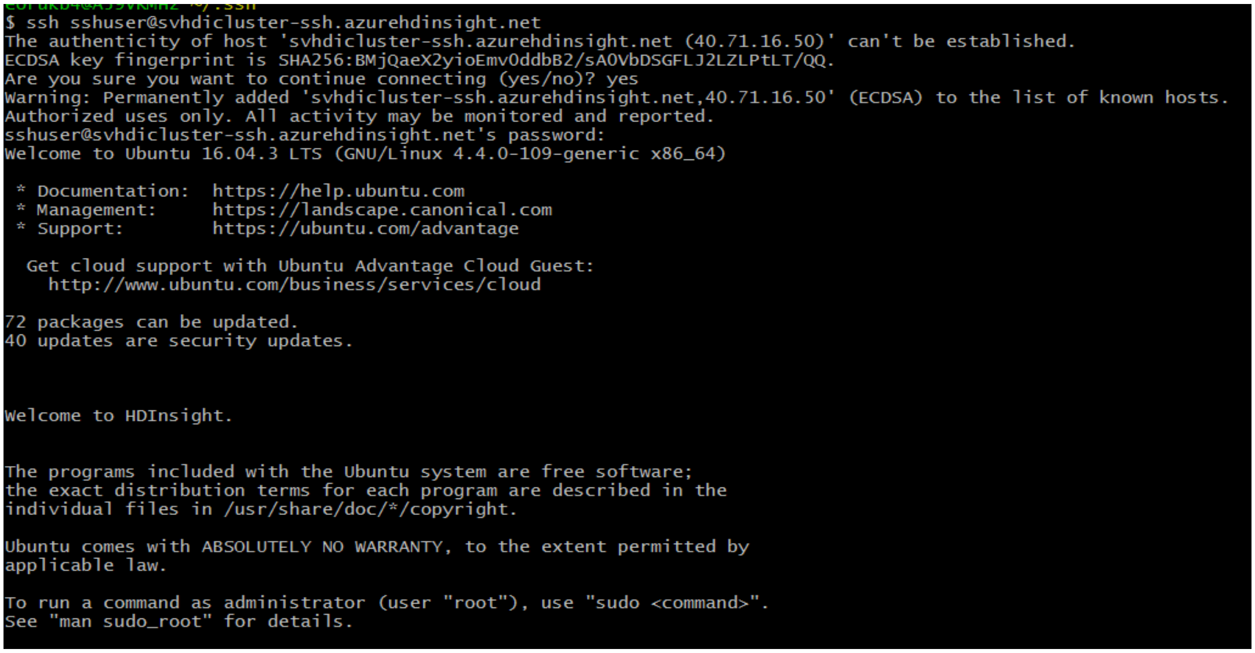
```
ssh sshuser@svhdicluster-ssh.azurehdinsight.net
```

Connect to cluster using Cluster Login

You can use the cluster dashboard to submit jobs, monitor resource usage and perform administrative actions. Documentation

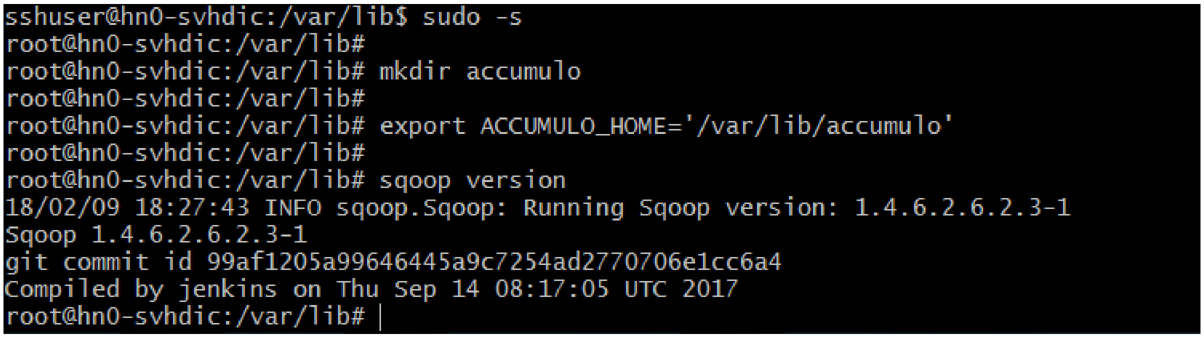
Cluster login username
admin

The below screen shows successful connection to the cluster via SSH:



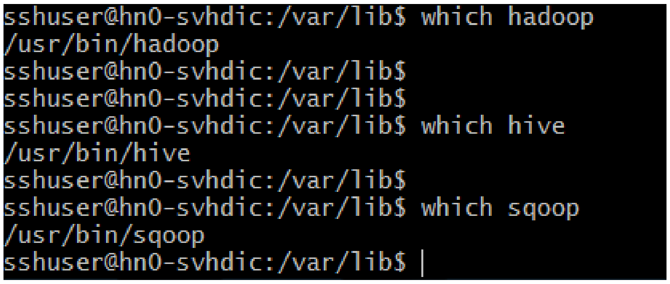
```
$ ssh sshuser@svhdicluster-ssh.azurehdinsight.net
The authenticity of host 'svhdicluster-ssh.azurehdinsight.net (40.71.16.50)' can't be established.
ECDSA key fingerprint is SHA256:BMjQaeX2yioEmv0ddbB2/sA0VbDSGFLJ2LZLPtLT/QQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'svhdicluster-ssh.azurehdinsight.net,40.71.16.50' (ECDSA) to the list of known hosts.
Authorized uses only. All activity may be monitored and reported.
sshuser@svhdicluster-ssh.azurehdinsight.net's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

72 packages can be updated.
40 updates are security updates.



Welcome to HDInsight.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

The following steps are needed for configuring Sqoop.



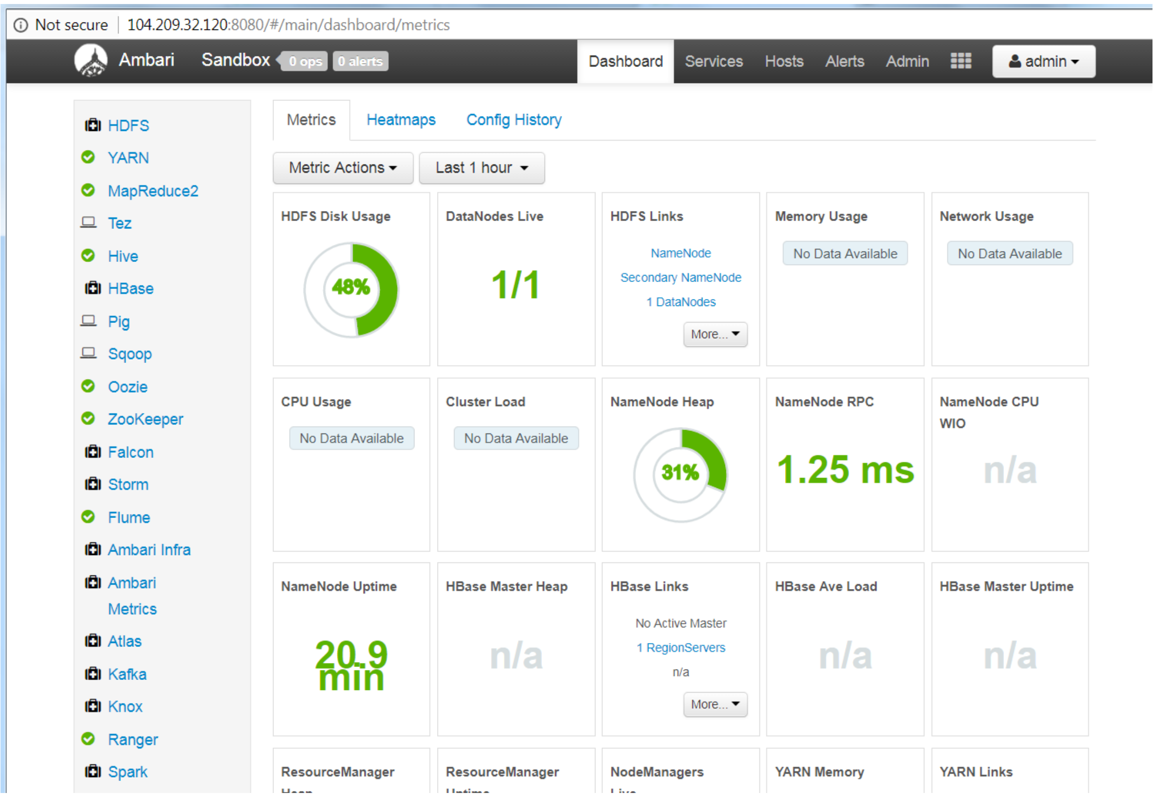
```
sshuser@hn0-svhdic:/var/lib$ sudo -s
root@hn0-svhdic:/var/lib#
root@hn0-svhdic:/var/lib# mkdir accumulo
root@hn0-svhdic:/var/lib#
root@hn0-svhdic:/var/lib# export ACCUMULO_HOME='/var/lib/accumulo'
root@hn0-svhdic:/var/lib#
root@hn0-svhdic:/var/lib# sqoop version
18/02/09 18:27:43 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6.2.6.2.3-1
Sqoop 1.4.6.2.6.2.3-1
git commit id 99af1205a99646445a9c7254ad2770706e1cc6a4
Compiled by jenkins on Thu Sep 14 08:17:05 UTC 2017
root@hn0-svhdic:/var/lib# |
```

The below link shows that Hadoop, Hive and Sqoop are successfully installed:



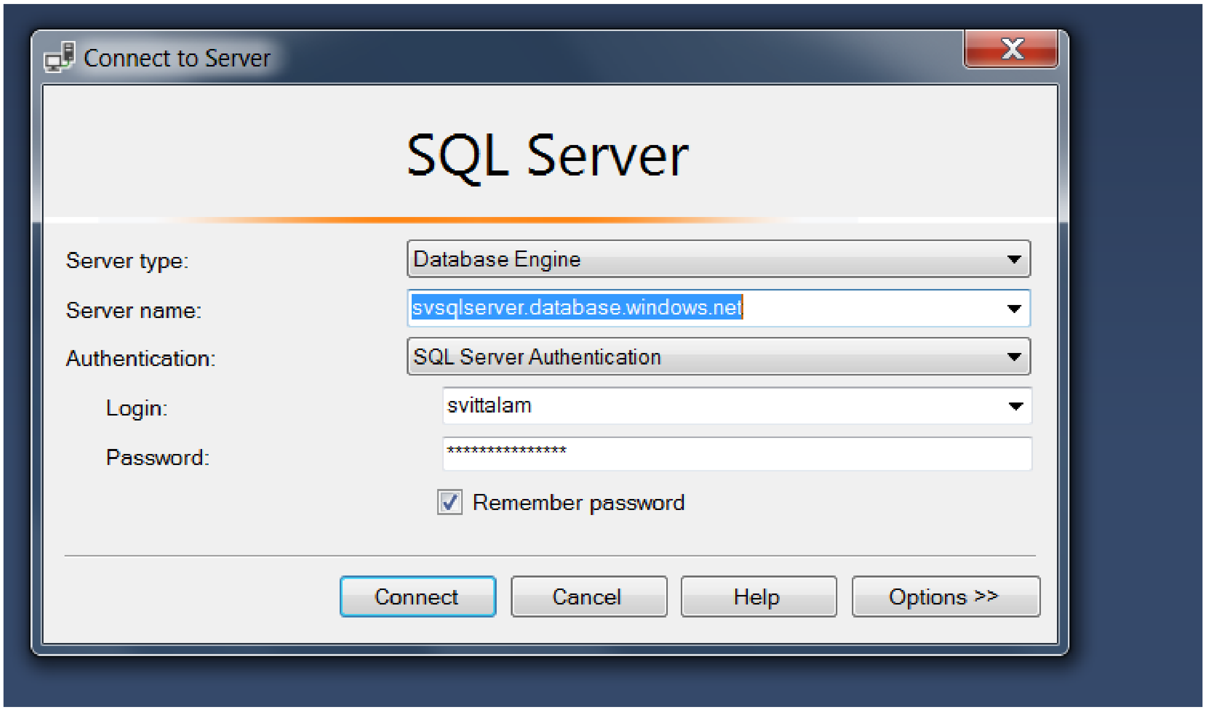
```
sshuser@hn0-svhdic:/var/lib$ which hadoop
/usr/bin/hadoop
sshuser@hn0-svhdic:/var/lib$
sshuser@hn0-svhdic:/var/lib$
sshuser@hn0-svhdic:/var/lib$ which hive
/usr/bin/hive
sshuser@hn0-svhdic:/var/lib$
sshuser@hn0-svhdic:/var/lib$ which sqoop
/usr/bin/sqoop
sshuser@hn0-svhdic:/var/lib$ |
```

Successful connection into Ambari Server UI:



Installation of SQL Server:

As a part of this project, SQL server is used as a source for the analytics. This contains part of the information needed for the final analytics



Server type: Database Engine

Server name: svsqlserver.database.windows.net

Authentication: SQL Server Authentication

Login: svittalam

Password: ***************

☑ Remember password

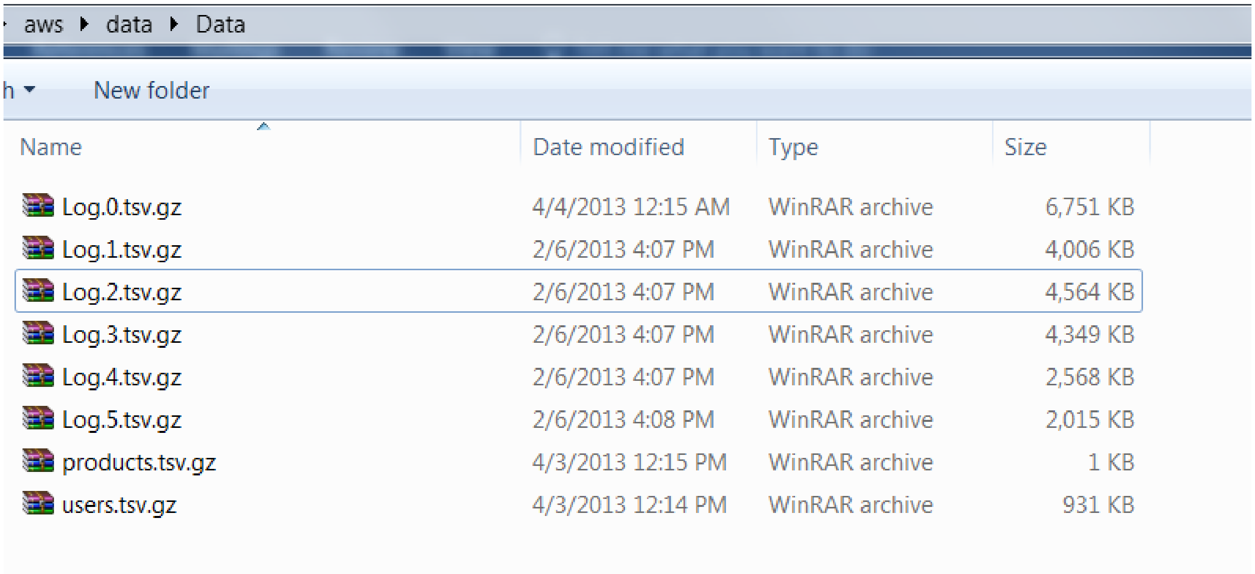
Data used for Visualizing Website Clickstream Data:

These are the website log files that have various attributes like time stamp, the visitor's IP address, destination URLs of the pages visited, and a user ID that uniquely identifies the website visitor.

https://s3.amazonaws.com/hw-sandbox/tutorial8/RefineDemoData.zip;

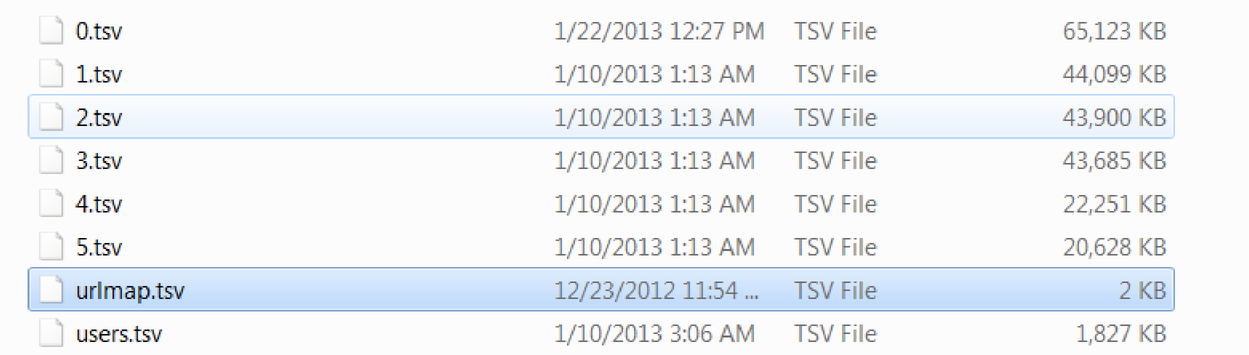
Following are the main files used for this process:

- Server-logs
- Products - urlmap
- Users



Name	Date modified	Type	Size
Log.0.tsv.gz	4/4/2013 12:15 AM	WinRAR archive	6,751 KB
Log.1.tsv.gz	2/6/2013 4:07 PM	WinRAR archive	4,006 KB
Log.2.tsv.gz	2/6/2013 4:07 PM	WinRAR archive	4,564 KB
Log.3.tsv.gz	2/6/2013 4:07 PM	WinRAR archive	4,349 KB
Log.4.tsv.gz	2/6/2013 4:07 PM	WinRAR archive	2,568 KB
Log.5.tsv.gz	2/6/2013 4:08 PM	WinRAR archive	2,015 KB
products.tsv.gz	4/3/2013 12:15 PM	WinRAR archive	1 KB
users.tsv.gz	4/3/2013 12:14 PM	WinRAR archive	931 KB

After Extraction:



0.tsv	1/22/2013 12:27 PM	TSV File	65,123 KB
1.tsv	1/10/2013 1:13 AM	TSV File	44,099 KB
2.tsv	1/10/2013 1:13 AM	TSV File	43,900 KB
3.tsv	1/10/2013 1:13 AM	TSV File	43,685 KB
4.tsv	1/10/2013 1:13 AM	TSV File	22,251 KB
5.tsv	1/10/2013 1:13 AM	TSV File	20,628 KB
urlmap.tsv	12/23/2012 11:54 ...	TSV File	2 KB
users.tsv	1/10/2013 3:06 AM	TSV File	1,827 KB

Sample data for Server-logs:

```
1331799426          2012-03-15 01:17:06  2860005755985467733            4611687631188657821            FAS-2.8-AS3          N
          0         99.122.210.248       1         0                    10        http://www.acme.com/SH55126545/VD55170364
          {7AAB8415-E803-3C5D-7100-E362D7F67CA7}
                                                   U         en-us,en;q=0.5                 516       575       1366      Y
          N         Y         2          0         304       sbcglobal.net        15/2/2012 4:16:0 4 240        45        41
          10002,00011,10020,00007        Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.648
          0         2         3          0         homestead            usa       528       fl        0         0         0
          0                                                                               0

          WPLG                                               0
                                         120

                    WPLG




          0
```

Sample Data for Products:

url	category
http://www.acme.com/	books
http://www.acme.com/SH55126545/VD55149415	movies
http://www.acme.com/SH55126545/VD55163347	games
http://www.acme.com/SH55126545/VD55165149	electronics

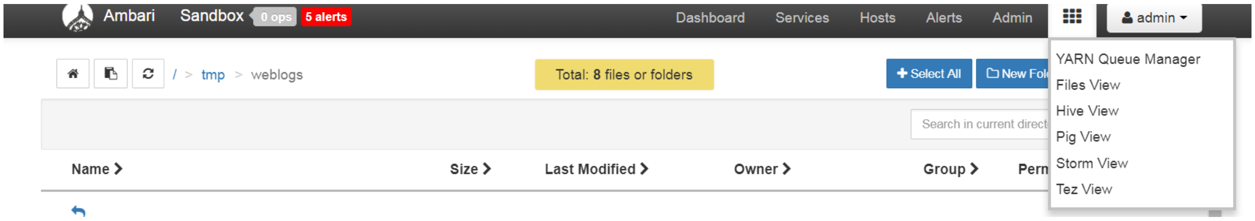
Sample data for Users:

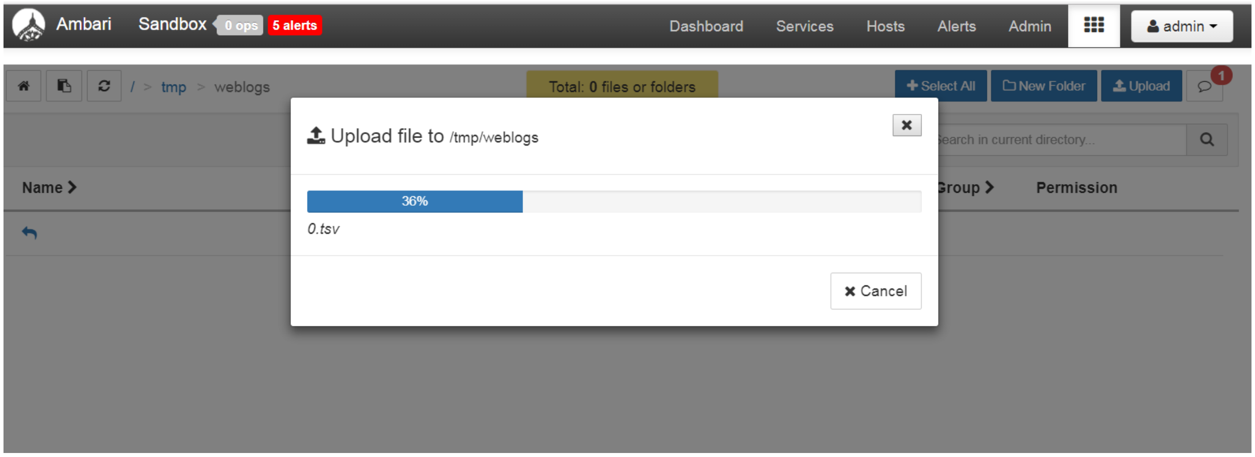
SWID	BIRTH_DT	GENDER_CD
0001BDD9-EABF-4D0D-81BD-D9EABFCD0D7D	8-Apr-84	F
00071AA7-86D2-4EB9-871A-A786D27EB9BA	7-Feb-88	F
00071B7D-31AF-4D85-871B-7D31AFFD852E	22-Oct-64	F
0007967E-F188-4598-9C7C-E64390482CFB	1-Jun-66	M

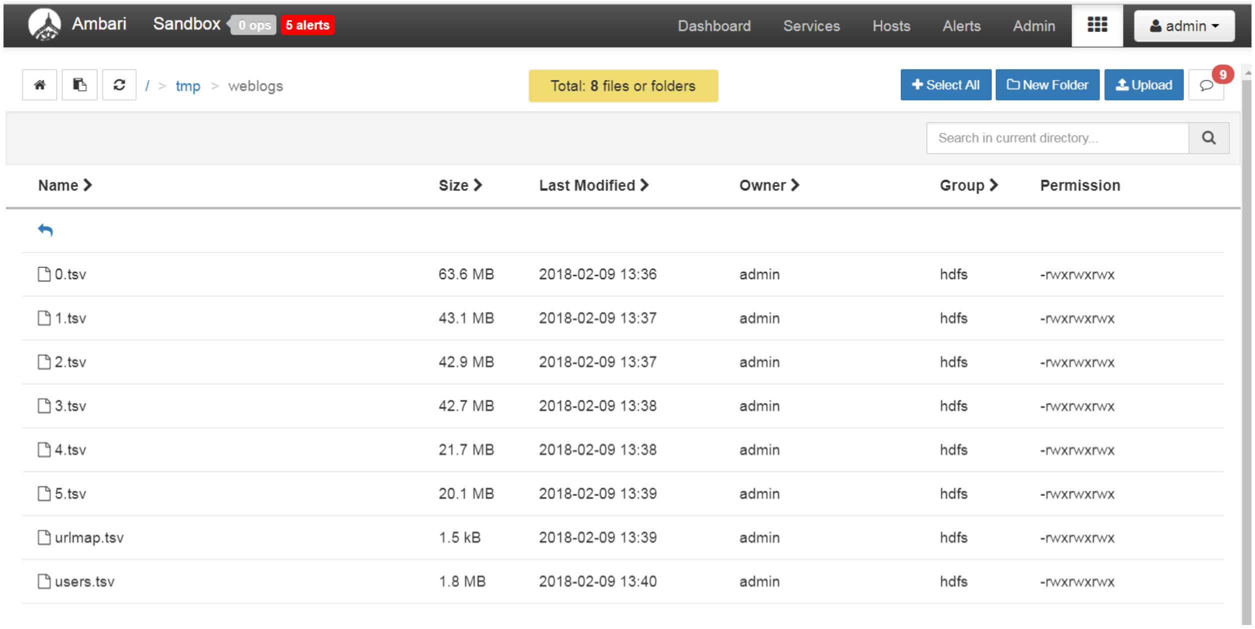
Description on how this data will be used:

- The server logs have attributes like IP address, location, timestamp, URL accessed, user details etc
- The product data contains information on what products does the website offers identified by the URL.
- The user data tell us more information about user demographics
- We will be combining this data and making a dataset that contains the following attributes:
    - Time of access
    - IP address
    - URL being accessed
    - User city, state and country
    - Product being accssed
    - Users gender and Age

File upload through the file view in Ambari Server UI:







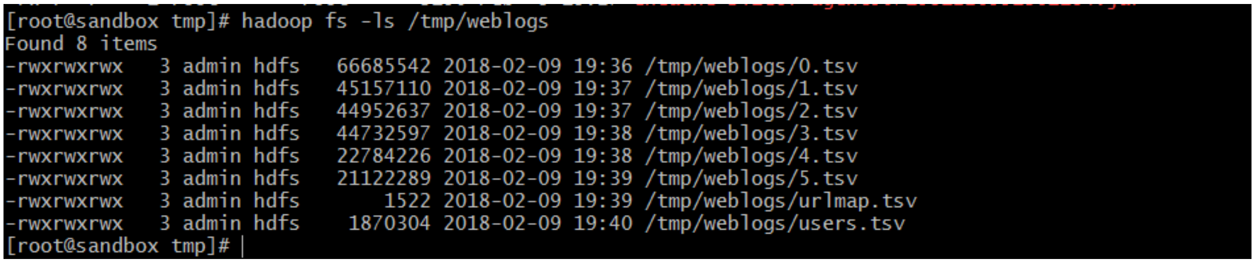
Name	Size	Last Modified	Owner	Group	Permission
0.tsv	63.6 MB	2018-02-09 13:36	admin	hdfs	-rwxrwxrwx
1.tsv	43.1 MB	2018-02-09 13:37	admin	hdfs	-rwxrwxrwx
2.tsv	42.9 MB	2018-02-09 13:37	admin	hdfs	-rwxrwxrwx
3.tsv	42.7 MB	2018-02-09 13:38	admin	hdfs	-rwxrwxrwx
4.tsv	21.7 MB	2018-02-09 13:38	admin	hdfs	-rwxrwxrwx
5.tsv	20.1 MB	2018-02-09 13:39	admin	hdfs	-rwxrwxrwx
urlmap.tsv	1.5 kB	2018-02-09 13:39	admin	hdfs	-rwxrwxrwx
users.tsv	1.8 MB	2018-02-09 13:40	admin	hdfs	-rwxrwxrwx

Data processing in Hadoop HDFS system:

In this step, we perform the following:

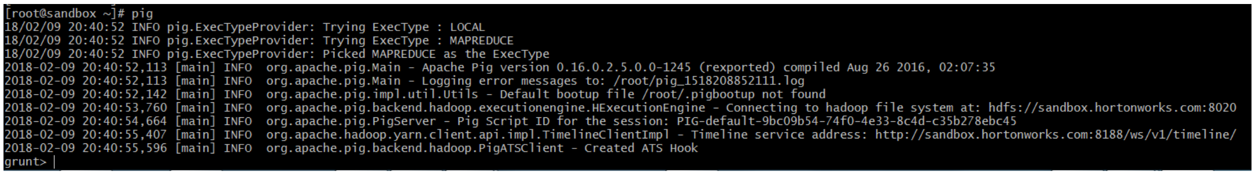
- Validate the data uploaded into HDFS via Ambari File View
- Use Pig latin script to merge the all the Log data file
- Use Sqoop to get the data from SQL Server RDBMS server
- Create Hive tables needed
- Load the data into Hive tables
- Finally create a view/table for the analytics/visualization.

Data check in the HDFS file system:



```
[root@sandbox tmp]# hadoop fs -ls /tmp/weblogs
Found 8 items
-rwxrwxrwx   3 admin hdfs   66685542 2018-02-09 19:36 /tmp/weblogs/0.tsv
-rwxrwxrwx   3 admin hdfs   45157110 2018-02-09 19:37 /tmp/weblogs/1.tsv
-rwxrwxrwx   3 admin hdfs   44952637 2018-02-09 19:37 /tmp/weblogs/2.tsv
-rwxrwxrwx   3 admin hdfs   44732597 2018-02-09 19:38 /tmp/weblogs/3.tsv
-rwxrwxrwx   3 admin hdfs   22784226 2018-02-09 19:38 /tmp/weblogs/4.tsv
-rwxrwxrwx   3 admin hdfs   21122289 2018-02-09 19:39 /tmp/weblogs/5.tsv
-rwxrwxrwx   3 admin hdfs       1522 2018-02-09 19:39 /tmp/weblogs/urlmap.tsv
-rwxrwxrwx   3 admin hdfs    1870304 2018-02-09 19:40 /tmp/weblogs/users.tsv
[root@sandbox tmp]# |
```

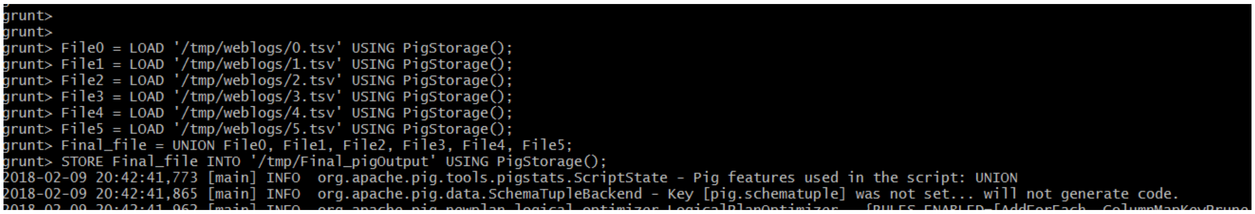
Invoking the PIG Latin utility:



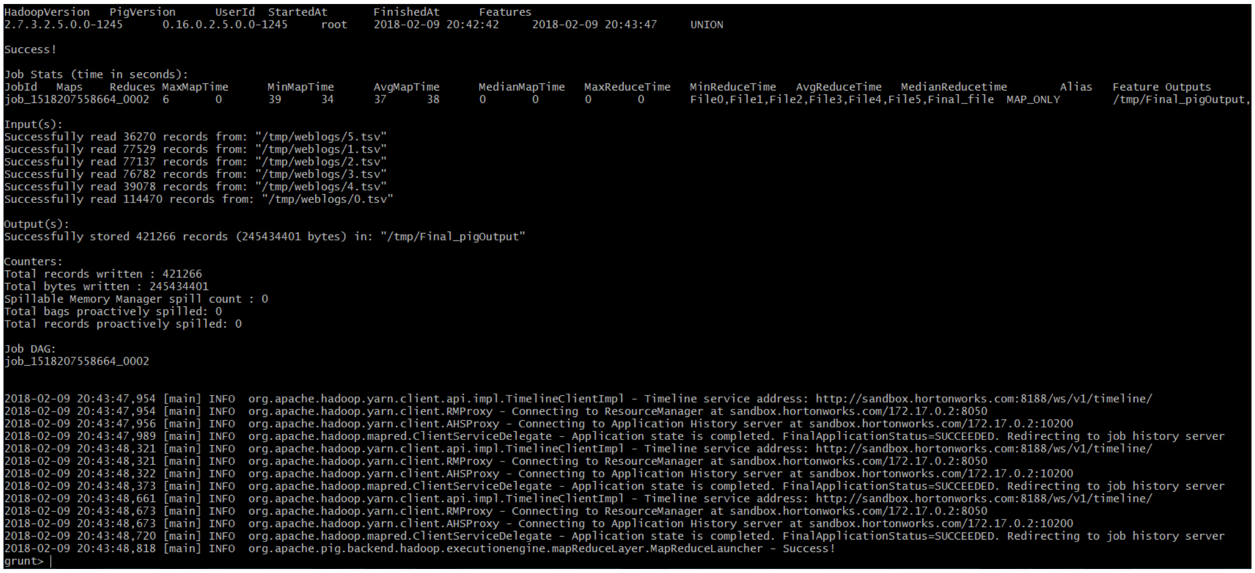
```
[root@sandbox ~]# pig
18/02/09 20:40:52 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
18/02/09 20:40:52 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
18/02/09 20:40:52 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2018-02-09 20:40:52,113 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0.2.5.0.0-1245 (rexported) compiled Aug 26 2016, 02:07:35
2018-02-09 20:40:52,113 [main] INFO  org.apache.pig.Main - Logging error messages to: /root/pig_1518208852111.log
2018-02-09 20:40:52,142 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /root/.pigbootup not found
2018-02-09 20:40:53,760 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: hdfs://sandbox.hortonworks.com:8020
2018-02-09 20:40:54,664 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-9bc09b54-74f0-4e33-8c4d-c35b278ebc45
2018-02-09 20:40:55,407 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/timeline/
2018-02-09 20:40:55,596 [main] INFO  org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
grunt> |
```

Using the GRUNT Editor:

The log data files are being merged in this step

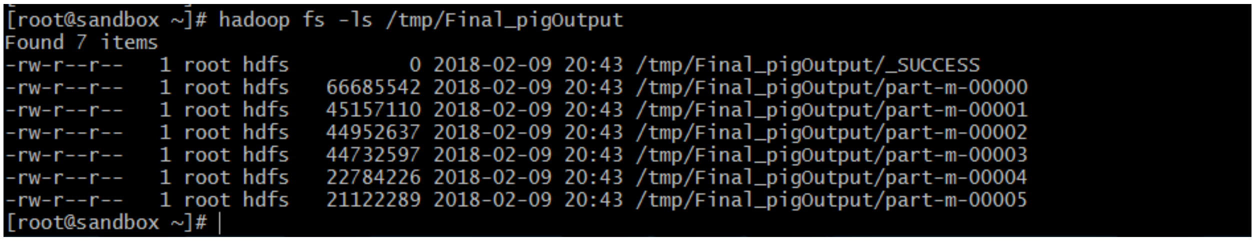


```
grunt>
grunt>
grunt> File0 = LOAD '/tmp/weblogs/0.tsv' USING PigStorage();
grunt> File1 = LOAD '/tmp/weblogs/1.tsv' USING PigStorage();
grunt> File2 = LOAD '/tmp/weblogs/2.tsv' USING PigStorage();
grunt> File3 = LOAD '/tmp/weblogs/3.tsv' USING PigStorage();
grunt> File4 = LOAD '/tmp/weblogs/4.tsv' USING PigStorage();
grunt> File5 = LOAD '/tmp/weblogs/5.tsv' USING PigStorage();
grunt> Final_file = UNION File0, File1, File2, File3, File4, File5;
grunt> STORE Final_file INTO '/tmp/Final_pigOutput' USING PigStorage();
2018-02-09 20:42:41,773 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: UNION
2018-02-09 20:42:41,865 [main] INFO  org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... will not generate code.
2018-02-09 20:42:41,962 [main] INFO  org.apache.pig.newplan.logical.optimizer.LogicalPlanOptimizer - {RULES_ENABLED=[AddForEach, ColumnMapKeyPrune
```



```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.7.3.2.5.0.0-1245      0.16.0.2.5.0.0-1245     root    2018-02-09 20:42:42     2018-02-09 20:43:47     UNION

Success!

Job Stats (time in seconds):
JobId   Maps    Reduces MaxMapTime      MinMapTime      AvgMapTime      MedianMapTime   MaxReduceTime   MinReduceTime   AvgReduceTime   MedianReducetime        Alias   Feature Outputs
job_1518207558664_0002  6       0       39      34      37      38      0       0       0       0       File0,File1,File2,File3,File4,File5,Final_file  MAP_ONLY        /tmp/Final_pigOutput,

Input(s):
Successfully read 36270 records from: "/tmp/weblogs/5.tsv"
Successfully read 77529 records from: "/tmp/weblogs/1.tsv"
Successfully read 77137 records from: "/tmp/weblogs/2.tsv"
Successfully read 76782 records from: "/tmp/weblogs/3.tsv"
Successfully read 39078 records from: "/tmp/weblogs/4.tsv"
Successfully read 114470 records from: "/tmp/weblogs/0.tsv"

Output(s):
Successfully stored 421266 records (245434401 bytes) in: "/tmp/Final_pigOutput"

Counters:
Total records written : 421266
Total bytes written : 245434401
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1518207558664_0002


2018-02-09 20:43:47,954 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/timeline/
2018-02-09 20:43:47,954 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8050
2018-02-09 20:43:47,956 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
2018-02-09 20:43:47,989 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-02-09 20:43:48,321 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/timeline/
2018-02-09 20:43:48,321 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8050
2018-02-09 20:43:48,322 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
2018-02-09 20:43:48,373 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-02-09 20:43:48,661 [main] INFO  org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://sandbox.hortonworks.com:8188/ws/v1/timeline/
2018-02-09 20:43:48,673 [main] INFO  org.apache.hadoop.yarn.client.RMProxy - Connecting to ResourceManager at sandbox.hortonworks.com/172.17.0.2:8050
2018-02-09 20:43:48,673 [main] INFO  org.apache.hadoop.yarn.client.AHSProxy - Connecting to Application History server at sandbox.hortonworks.com/172.17.0.2:10200
2018-02-09 20:43:48,720 [main] INFO  org.apache.hadoop.mapred.ClientServiceDelegate - Application state is completed. FinalApplicationStatus=SUCCEEDED. Redirecting to job history server
2018-02-09 20:43:48,818 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt> |
```

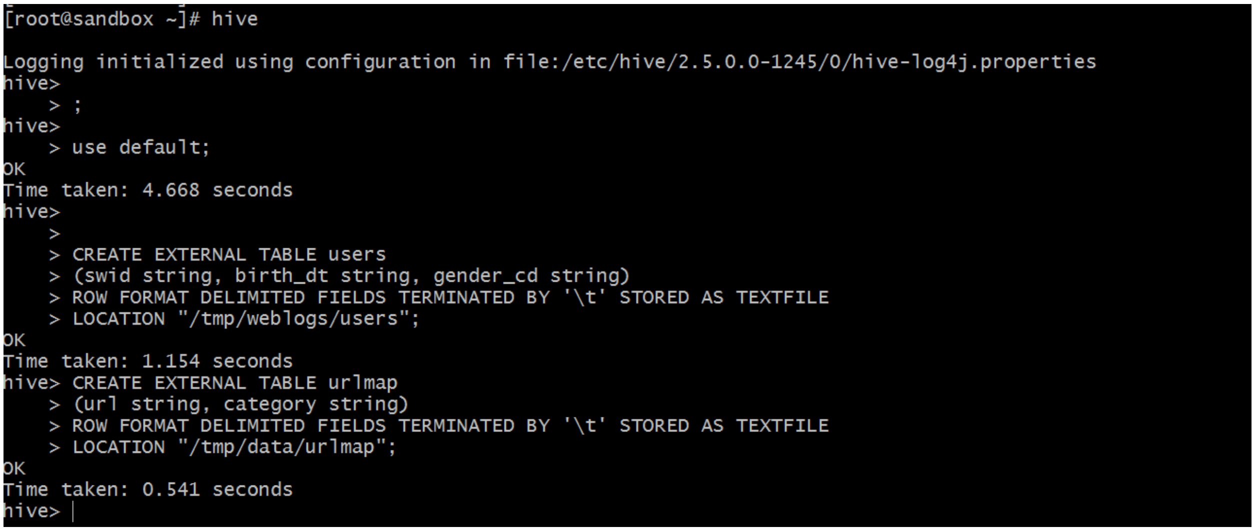
This shows the final output created by the PIG script. The merged files are collected in a separate folder:



```
[root@sandbox ~]# hadoop fs -ls /tmp/Final_pigOutput
Found 7 items
-rw-r--r--   1 root hdfs          0 2018-02-09 20:43 /tmp/Final_pigOutput/_SUCCESS
-rw-r--r--   1 root hdfs   66685542 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00000
-rw-r--r--   1 root hdfs   45157110 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00001
-rw-r--r--   1 root hdfs   44952637 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00002
-rw-r--r--   1 root hdfs   44732597 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00003
-rw-r--r--   1 root hdfs   22784226 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00004
-rw-r--r--   1 root hdfs   21122289 2018-02-09 20:43 /tmp/Final_pigOutput/part-m-00005
[root@sandbox ~]# |
```

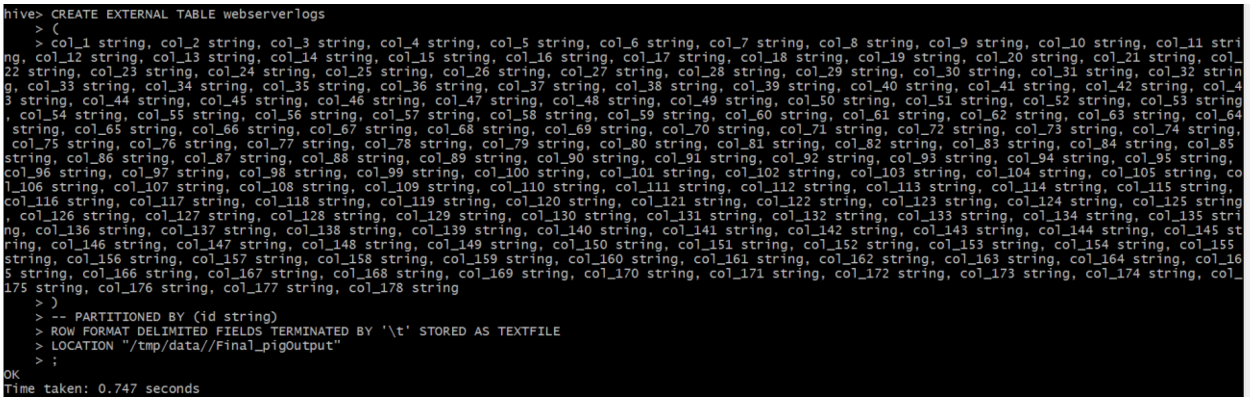
Invoking the HIVE Editor:

Step for creating tables:



```
[root@sandbox ~]# hive

Logging initialized using configuration in file:/etc/hive/2.5.0.0-1245/0/hive-log4j.properties
hive>
    > ;
hive>
    > use default;
OK
Time taken: 4.668 seconds
hive>
    >
    > CREATE EXTERNAL TABLE users
    > (swid string, birth_dt string, gender_cd string)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE
    > LOCATION "/tmp/weblogs/users";
OK
Time taken: 1.154 seconds
hive> CREATE EXTERNAL TABLE urlmap
    > (url string, category string)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE
    > LOCATION "/tmp/data/urlmap";
OK
Time taken: 0.541 seconds
hive> |
```

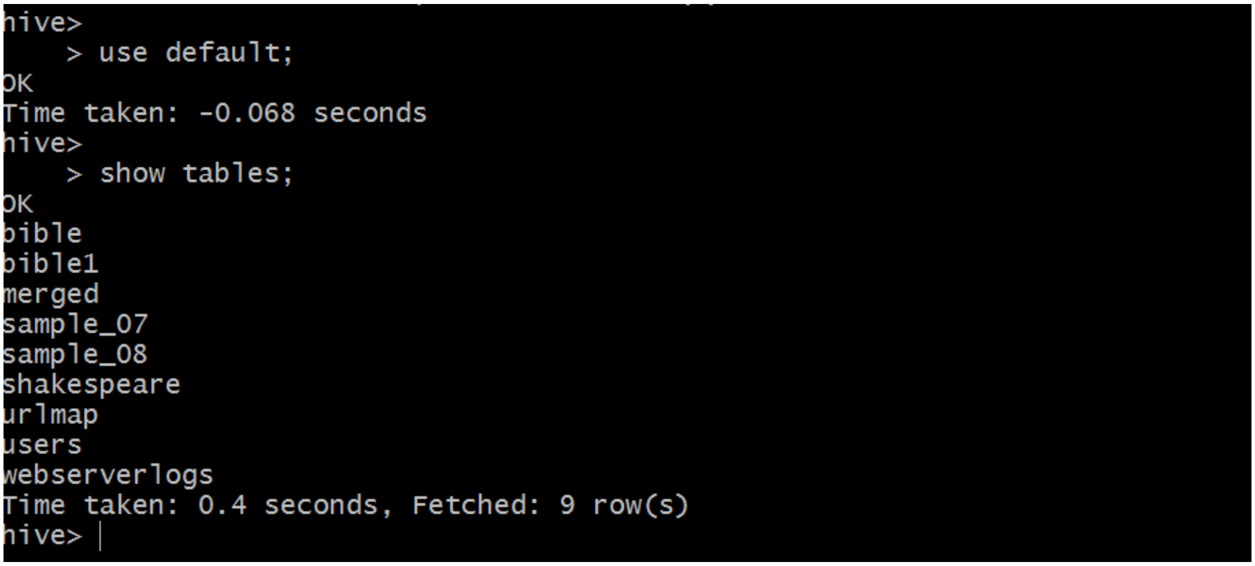
Step for creating tables:



```
hive> CREATE EXTERNAL TABLE webserverlogs
    > (
    > col_1 string, col_2 string, col_3 string, col_4 string, col_5 string, col_6 string, col_7 string, col_8 string, col_9 string, col_10 string, col_11 stri
ng, col_12 string, col_13 string, col_14 string, col_15 string, col_16 string, col_17 string, col_18 string, col_19 string, col_20 string, col_21 string, col_
22 string, col_23 string, col_24 string, col_25 string, col_26 string, col_27 string, col_28 string, col_29 string, col_30 string, col_31 string, col_32 strin
g, col_33 string, col_34 string, col_35 string, col_36 string, col_37 string, col_38 string, col_39 string, col_40 string, col_41 string, col_42 string, col_4
3 string, col_44 string, col_45 string, col_46 string, col_47 string, col_48 string, col_49 string, col_50 string, col_51 string, col_52 string, col_53 string
, col_54 string, col_55 string, col_56 string, col_57 string, col_58 string, col_59 string, col_60 string, col_61 string, col_62 string, col_63 string, col_64
 string, col_65 string, col_66 string, col_67 string, col_68 string, col_69 string, col_70 string, col_71 string, col_72 string, col_73 string, col_74 string,
 col_75 string, col_76 string, col_77 string, col_78 string, col_79 string, col_80 string, col_81 string, col_82 string, col_83 string, col_84 string, col_85
string, col_86 string, col_87 string, col_88 string, col_89 string, col_90 string, col_91 string, col_92 string, col_93 string, col_94 string, col_95 string,
col_96 string, col_97 string, col_98 string, col_99 string, col_100 string, col_101 string, col_102 string, col_103 string, col_104 string, col_105 string, co
l_106 string, col_107 string, col_108 string, col_109 string, col_110 string, col_111 string, col_112 string, col_113 string, col_114 string, col_115 string,
col_116 string, col_117 string, col_118 string, col_119 string, col_120 string, col_121 string, col_122 string, col_123 string, col_124 string, col_125 string
, col_126 string, col_127 string, col_128 string, col_129 string, col_130 string, col_131 string, col_132 string, col_133 string, col_134 string, col_135 stri
ng, col_136 string, col_137 string, col_138 string, col_139 string, col_140 string, col_141 string, col_142 string, col_143 string, col_144 string, col_145 st
ring, col_146 string, col_147 string, col_148 string, col_149 string, col_150 string, col_151 string, col_152 string, col_153 string, col_154 string, col_155
string, col_156 string, col_157 string, col_158 string, col_159 string, col_160 string, col_161 string, col_162 string, col_163 string, col_164 string, col_16
5 string, col_166 string, col_167 string, col_168 string, col_169 string, col_170 string, col_171 string, col_172 string, col_173 string, col_174 string, col_
175 string, col_176 string, col_177 string, col_178 string
    > )
    > -- PARTITIONED BY (id string)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE
    > LOCATION "/tmp/data//Final_pigOutput"
    > ;
OK
Time taken: 0.747 seconds
```

This shows the 3 tables that are created:

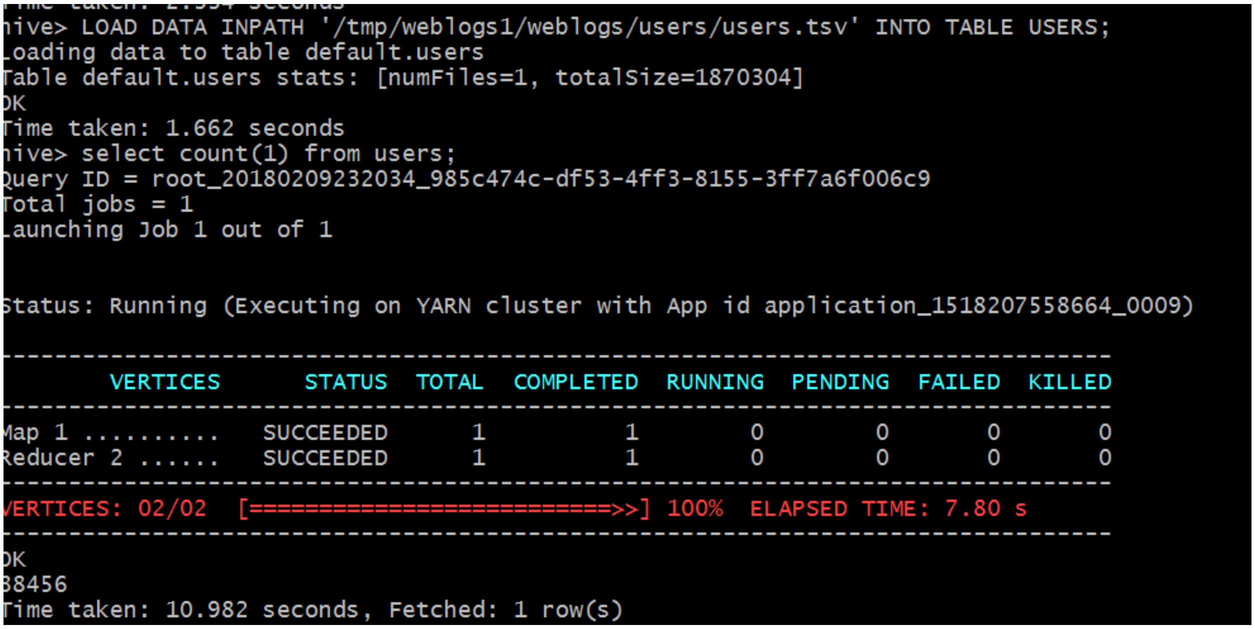
- Webserverlogs
- Urlmap
- Users



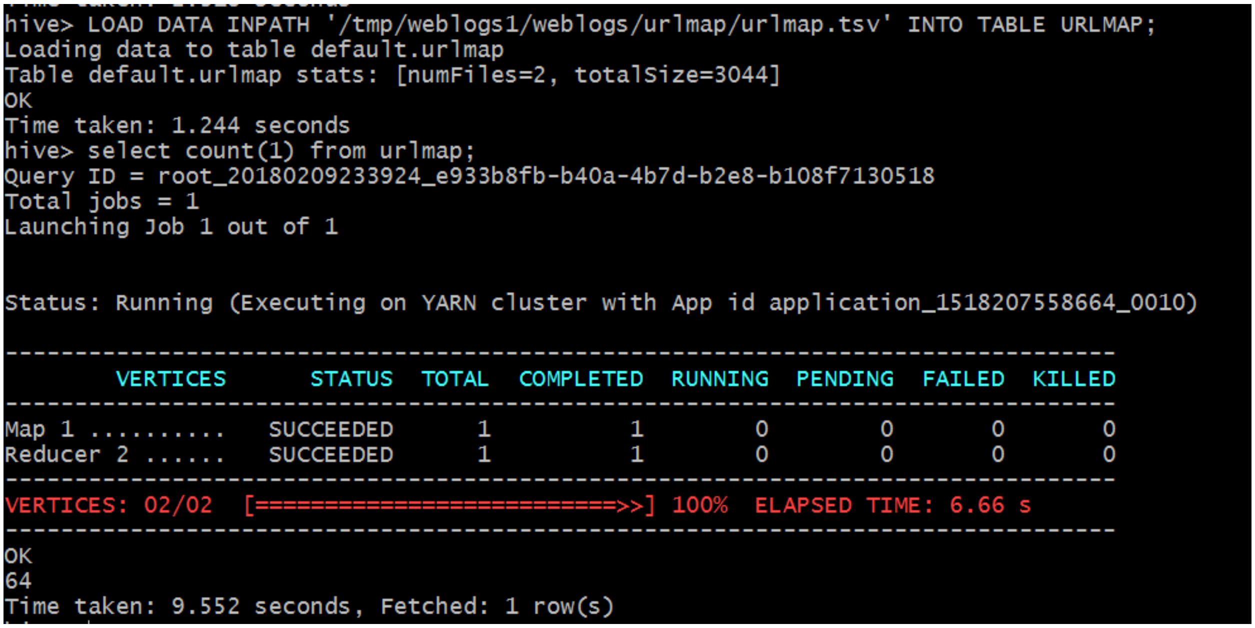
```
hive>
    > use default;
OK
Time taken: -0.068 seconds
hive>
    > show tables;
OK
bible
bible1
merged
sample_07
sample_08
shakespeare
urlmap
users
webserverlogs
Time taken: 0.4 seconds, Fetched: 9 row(s)
hive> |
```

The next few steps show the following:

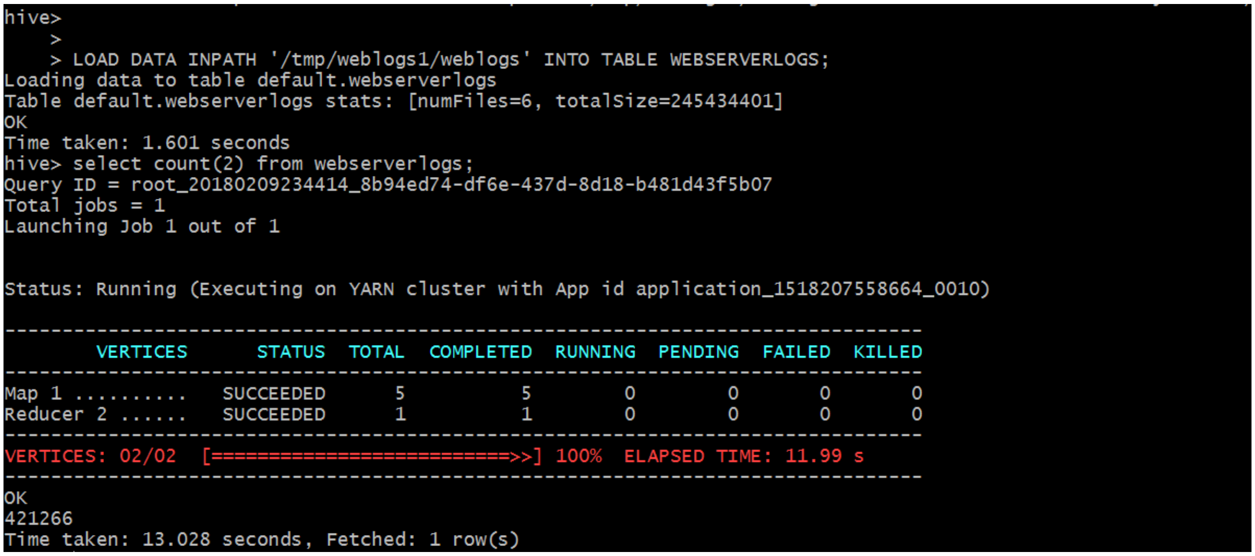
- Loading data into tables
- Creating the final table for analytics/visualization.



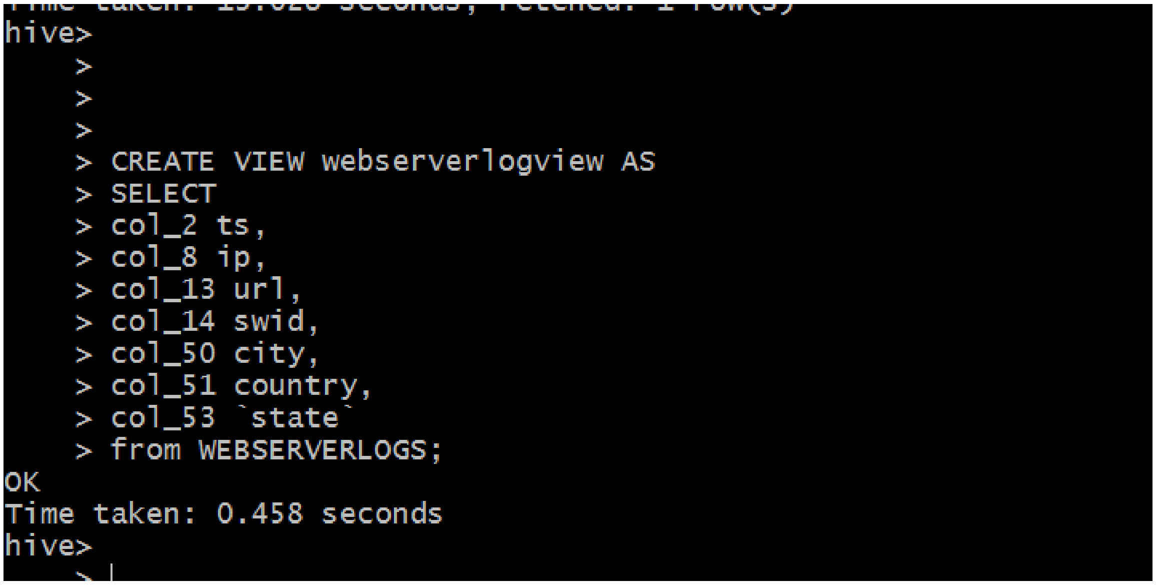
```
hive> LOAD DATA INPATH '/tmp/weblogs1/weblogs/users/users.tsv' INTO TABLE USERS;
Loading data to table default.users
Table default.users stats: [numFiles=1, totalSize=1870304]
OK
Time taken: 1.662 seconds
hive> select count(1) from users;
Query ID = root_20180209232034_985c474c-df53-4ff3-8155-3ff7a6f006c9
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1518207558664_0009)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 7.80 s
--------------------------------------------------------------------------------
OK
38456
Time taken: 10.982 seconds, Fetched: 1 row(s)
```



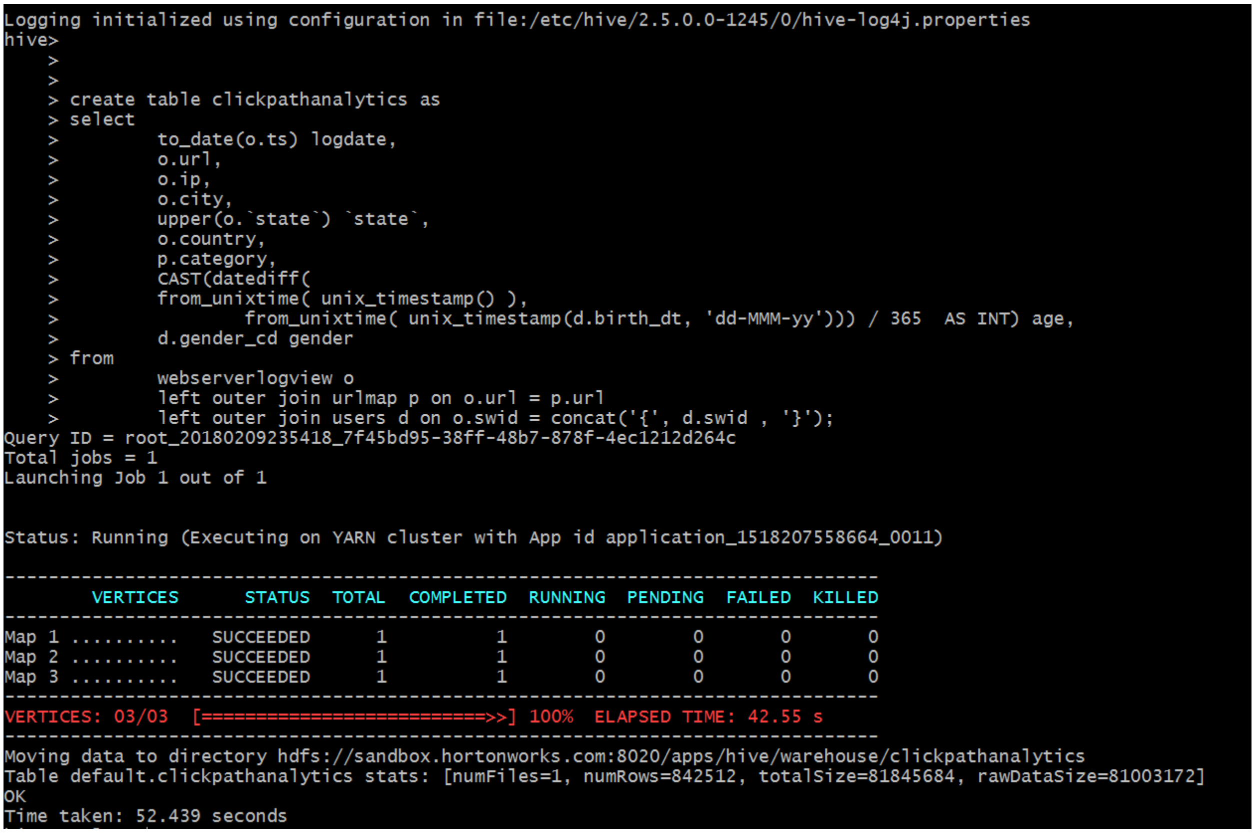
```
hive> LOAD DATA INPATH '/tmp/weblogs1/weblogs/urlmap/urlmap.tsv' INTO TABLE URLMAP;
Loading data to table default.urlmap
Table default.urlmap stats: [numFiles=2, totalSize=3044]
OK
Time taken: 1.244 seconds
hive> select count(1) from urlmap;
Query ID = root_20180209233924_e933b8fb-b40a-4b7d-b2e8-b108f7130518
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1518207558664_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.66 s
--------------------------------------------------------------------------------
OK
64
Time taken: 9.552 seconds, Fetched: 1 row(s)
```



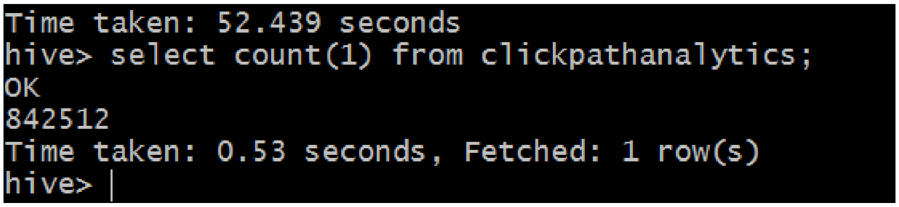
```
hive>
    >
    > LOAD DATA INPATH '/tmp/weblogs1/weblogs' INTO TABLE WEBSERVERLOGS;
Loading data to table default.webserverlogs
Table default.webserverlogs stats: [numFiles=6, totalSize=245434401]
OK
Time taken: 1.601 seconds
hive> select count(2) from webserverlogs;
Query ID = root_20180209234414_8b94ed74-df6e-437d-8d18-b481d43f5b07
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1518207558664_0010)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 .........    SUCCEEDED      5          5        0        0       0       0
Reducer 2 ......   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 11.99 s
--------------------------------------------------------------------------------
OK
421266
Time taken: 13.028 seconds, Fetched: 1 row(s)
```



```
Time taken: 13.028 seconds, Fetched: 1 row(s)
hive>
    >
    >
    >
    > CREATE VIEW webserverlogview AS
    > SELECT
    > col_2 ts,
    > col_8 ip,
    > col_13 url,
    > col_14 swid,
    > col_50 city,
    > col_51 country,
    > col_53 `state`
    > from WEBSERVERLOGS;
OK
Time taken: 0.458 seconds
hive>
    >
```



```
Logging initialized using configuration in file:/etc/hive/2.5.0.0-1245/0/hive-log4j.properties
hive>
    >
    >
    > create table clickpathanalytics as
    > select
    >         to_date(o.ts) logdate,
    >         o.url,
    >         o.ip,
    >         o.city,
    >         upper(o.`state`) `state`,
    >         o.country,
    >         p.category,
    >         CAST(datediff(
    >         from_unixtime( unix_timestamp() ),
    >                 from_unixtime( unix_timestamp(d.birth_dt, 'dd-MMM-yy'))) / 365  AS INT) age,
    >         d.gender_cd gender
    > from
    >         webserverlogview o
    >         left outer join urlmap p on o.url = p.url
    >         left outer join users d on o.swid = concat('{', d.swid , '}');
Query ID = root_20180209235418_7f45bd95-38ff-48b7-878f-4ec1212d264c
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1518207558664_0011)

--------------------------------------------------------------------------------
        VERTICES      STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
--------------------------------------------------------------------------------
Map 1 ..........   SUCCEEDED      1          1        0        0       0       0
Map 2 ..........   SUCCEEDED      1          1        0        0       0       0
Map 3 ..........   SUCCEEDED      1          1        0        0       0       0
--------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 42.55 s
--------------------------------------------------------------------------------
Moving data to directory hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/clickpathanalytics
Table default.clickpathanalytics stats: [numFiles=1, numRows=842512, totalSize=81845684, rawDataSize=81003172]
OK
Time taken: 52.439 seconds
```

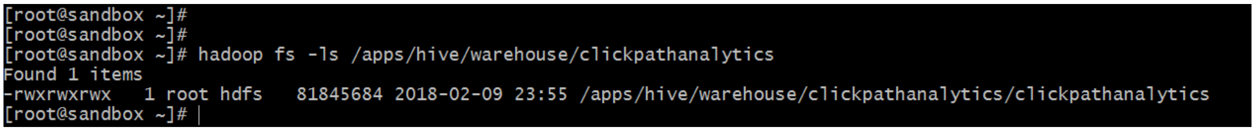


```
Time taken: 52.439 seconds
hive> select count(1) from clickpathanalytics;
OK
842512
Time taken: 0.53 seconds, Fetched: 1 row(s)
hive> |
```

Visualization and Analytics:

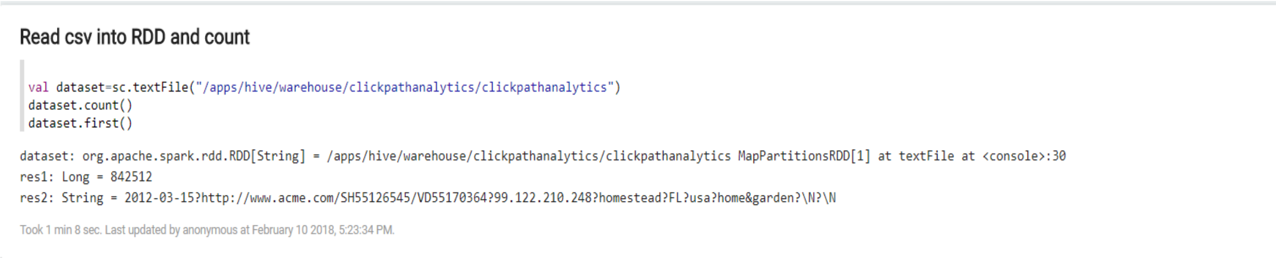
Analytics using SparkSQL with Zepplin:

The final file for analytics exist at the following location:



```
[root@sandbox ~]#
[root@sandbox ~]#
[root@sandbox ~]# hadoop fs -ls /apps/hive/warehouse/clickpathanalytics
Found 1 items
-rwxrwxrwx   1 root hdfs   81845684 2018-02-09 23:55 /apps/hive/warehouse/clickpathanalytics/clickpathanalytics
[root@sandbox ~]# |
```

Reading the dataset into a Spark RDD:



Read csv into RDD and count

```
val dataset=sc.textFile("/apps/hive/warehouse/clickpathanalytics/clickpathanalytics")
dataset.count()
dataset.first()
```

```
dataset: org.apache.spark.rdd.RDD[String] = /apps/hive/warehouse/clickpathanalytics/clickpathanalytics MapPartitionsRDD[1] at textFile at <console>:30
res1: Long = 842512
res2: String = 2012-03-15?http://www.acme.com/SH55126545/VD55170364?99.122.210.248?homestead?FL?usa?home&garden?\N?\N
```

Took 1 min 8 sec. Last updated by anonymous at February 10 2018, 5:23:34 PM.

Registering the RDD as a table:

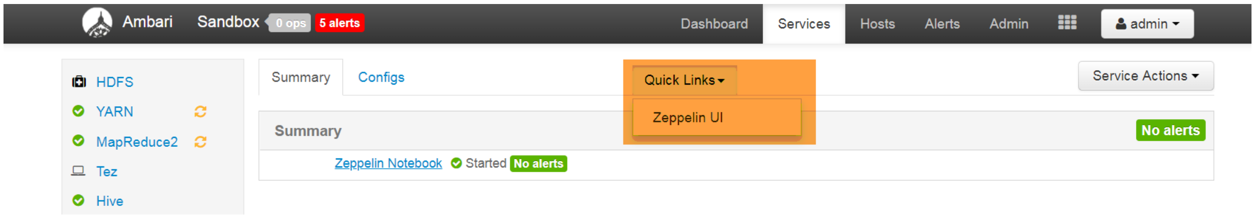
```
--------------------------------

case class clickpath (logdate:  String, url: String, ip: String, city: String, state: String, country: String, category: String, age: Integer, gender:  String,)

val clickpath = dataset.map(k=>k.split(",")).map(

   k => clickpath(k(0),k(1),k(2),k(3), k(4), k(5), k(6),k(7).toInt, k(9) )    )

clickpath.toDF().registerTempTable("clickpath")
```

Visualization using Zepplin:

Apache Zeppelin is a web-based notebook that enables interactive data analytics. With Zeppelin, you can make beautiful data-driven, interactive and collaborative documents with a rich set of pre-built language backends (or interpreters) such as Scala (with Apache Spark), Python (with Apache Spark), SparkSQL, Hive, Markdown, Angular, and Shell.

Source: https://hortonworks.com/tutorial/getting-started-with-apache-zeppelin/

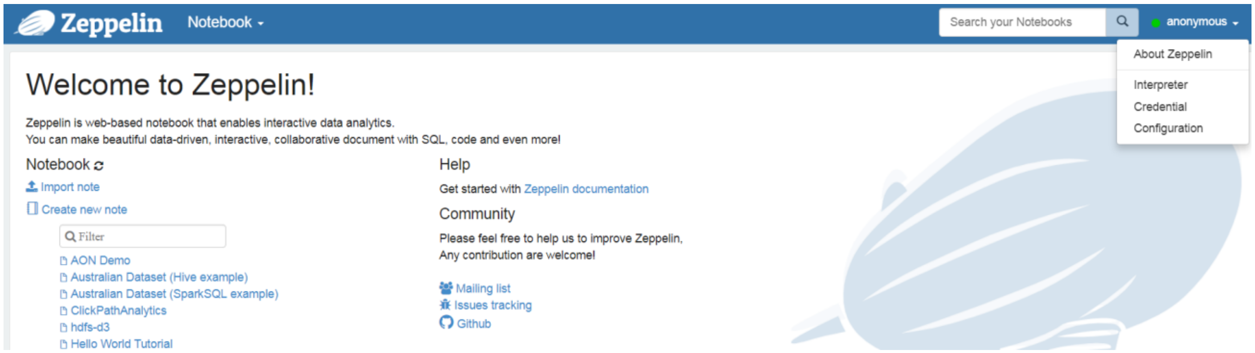
Invoking Zepplin through the Ambari UI:



Interpretor Binding:

We need to bind the appropriate interpretor. For our example, we will be using the following interpretor:

- Spark
- JDBC



Welcome to Zeppelin!

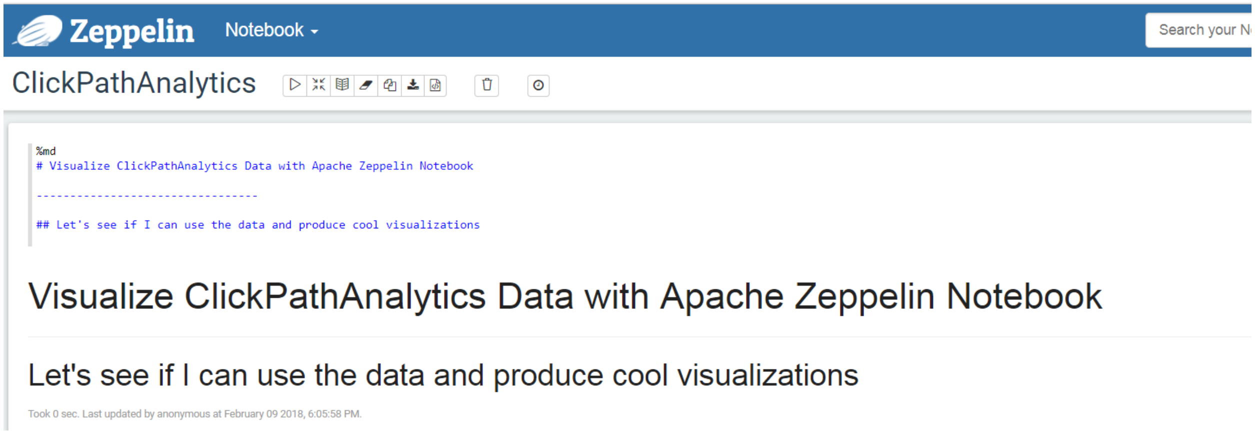
Zeppelin is web-based notebook that enables interactive data analytics.
You can make beautiful data-driven, interactive, collaborative document with SQL, code and even more!

Get started with Zeppelin documentation

Community

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

Zepplin Visualization 1:



ClickPathAnalytics

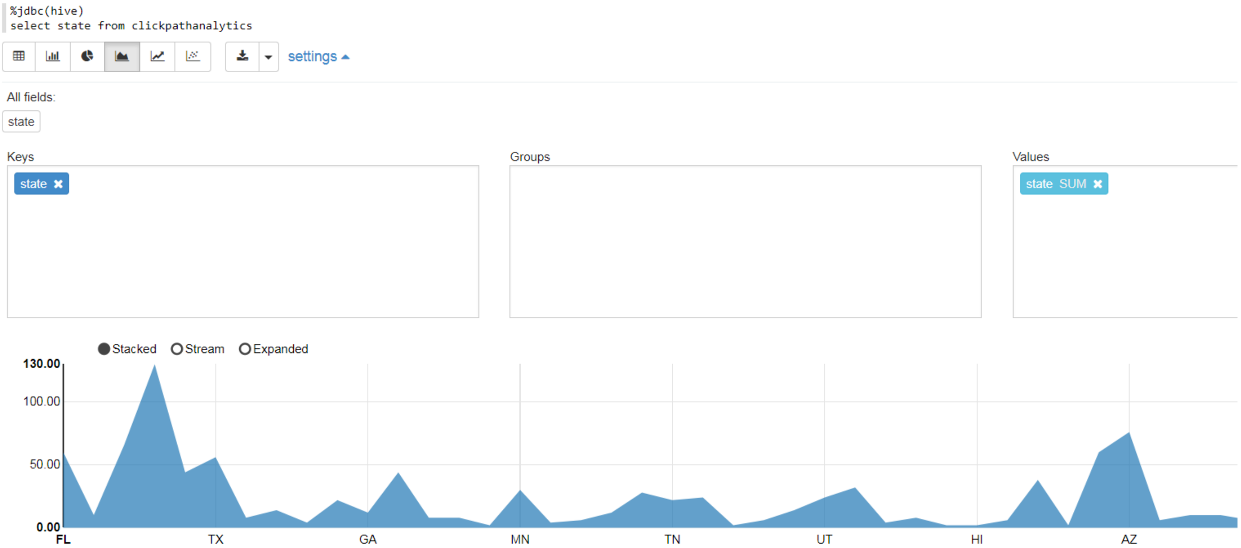
```
%md
# Visualize ClickPathAnalytics Data with Apache Zeppelin Notebook

--------------------------------

## Let's see if I can use the data and produce cool visualizations
```

Visualize ClickPathAnalytics Data with Apache Zeppelin Notebook

Let's see if I can use the data and produce cool visualizations

Took 0 sec. Last updated by anonymous at February 09 2018, 6:05:58 PM.

Zepplin Visualization 2:



```
%jdbc(hive)
select state from clickpathanalytics
```

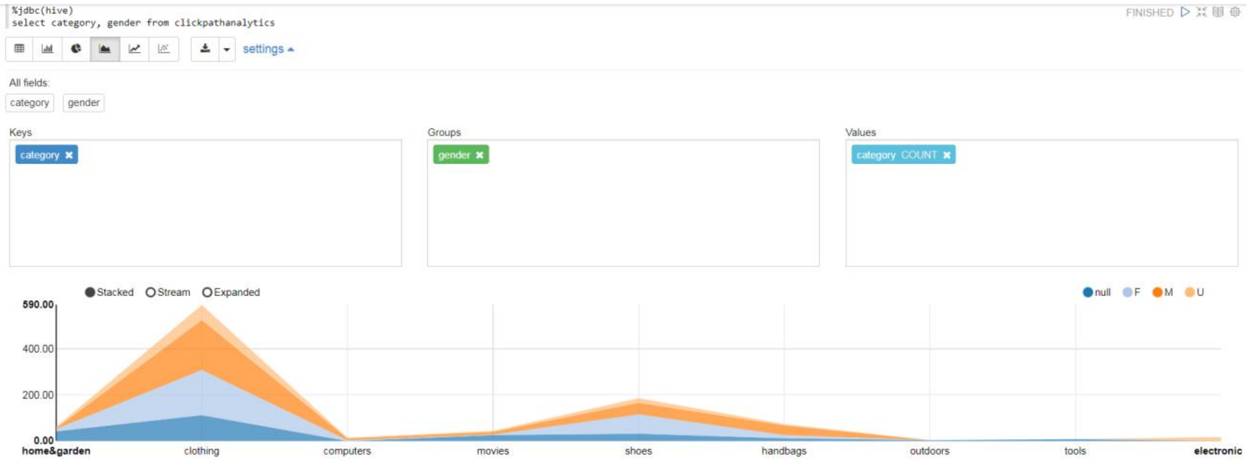
All fields: state

Keys: state

Groups:

Values: state SUM

Zepplin Visualization 3:



```
%jdbc(hive)
select category, gender from clickpathanalytics
```

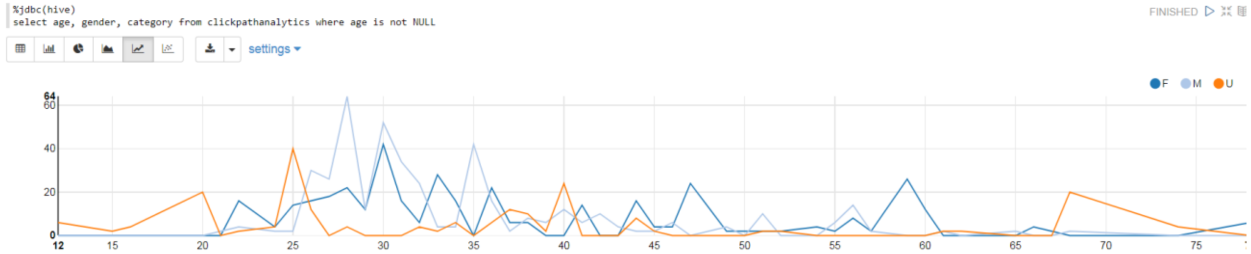
All fields: category gender

Keys: category

Groups: gender

Values: category COUNT

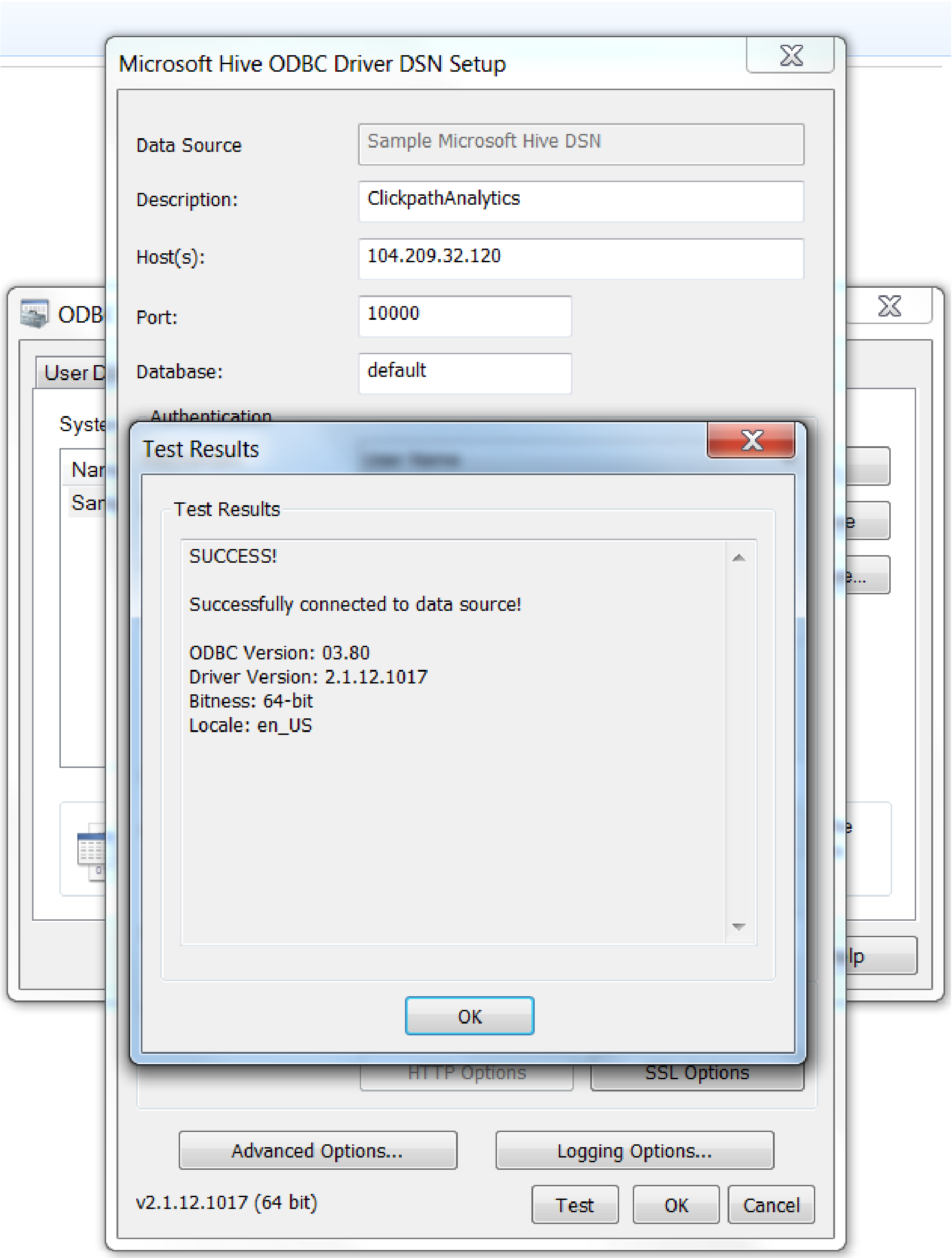
Zepplin Visualization 4:



```
%jdbc(hive)
select age, gender, category from clickpathanalytics where age is not NULL
```

Visualization with PowerBI:

Setting up the DSN from control panel to for Hadoop HIVE:



Data Source: Sample Microsoft Hive DSN

Description: ClickpathAnalytics

Host(s): 104.209.32.120

Port: 10000

Database: default

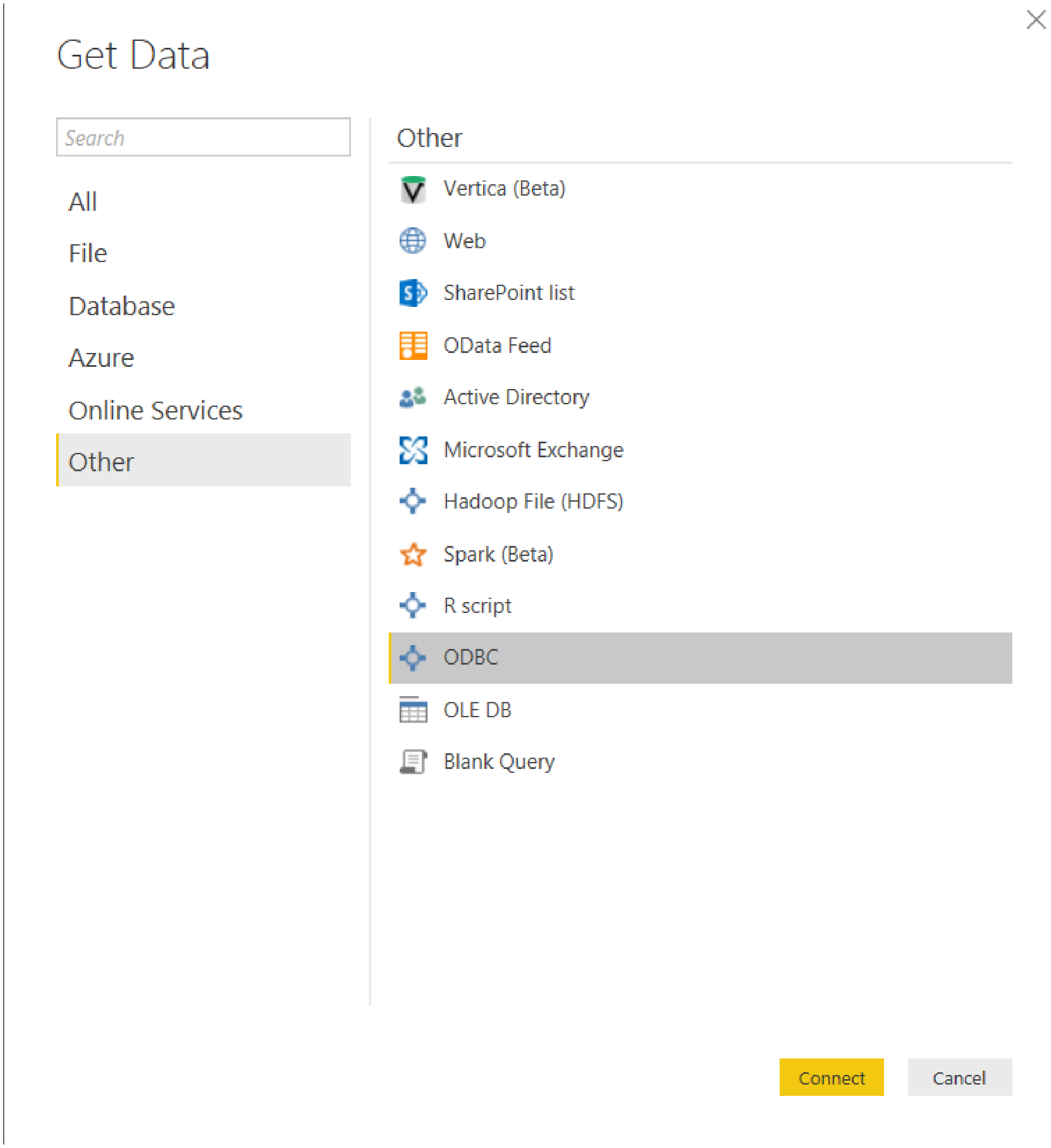
v2.1.12.1017 (64 bit)

Test Results

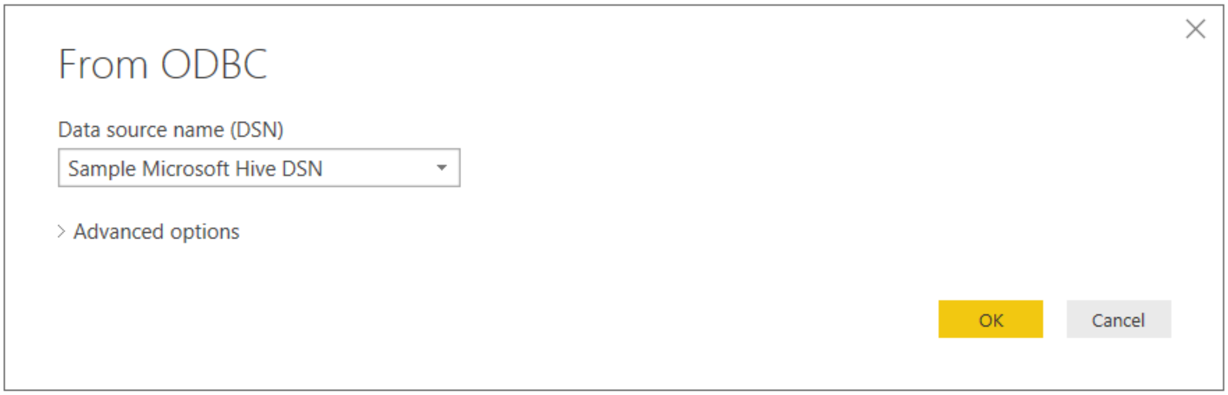
```
SUCCESS!

Successfully connected to data source!

ODBC Version: 03.80
Driver Version: 2.1.12.1017
Bitness: 64-bit
Locale: en_US
```

Steps to import data into PowerBI:

Use the 'Get Data' option and choose ODBC option and click Connect



That will list all the ODBC DSNs created in the system and we choose the one created for HIVE:

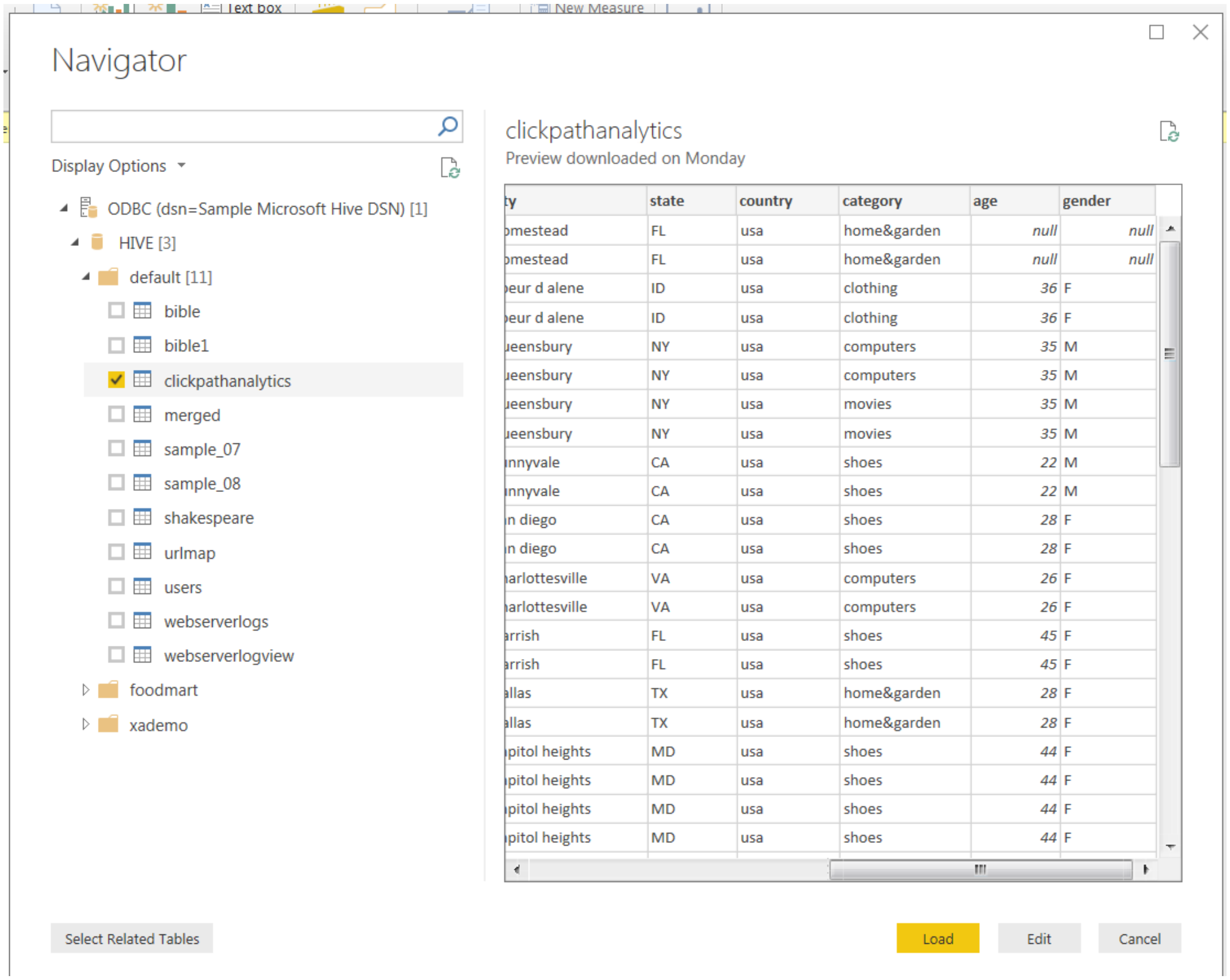


From ODBC

Data source name (DSN)

Sample Microsoft Hive DSN

Choose the data source object from the default database → Clickpathanalytics. That should load the input data needed for visualization.



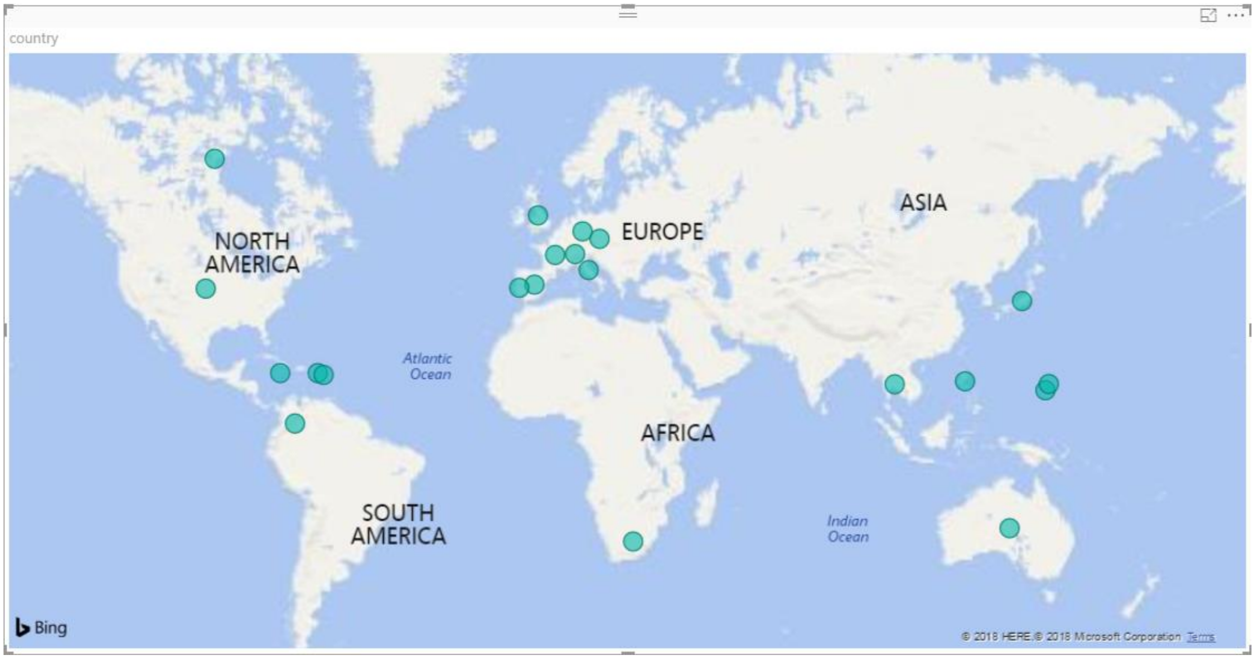
Navigator

clickpathanalytics

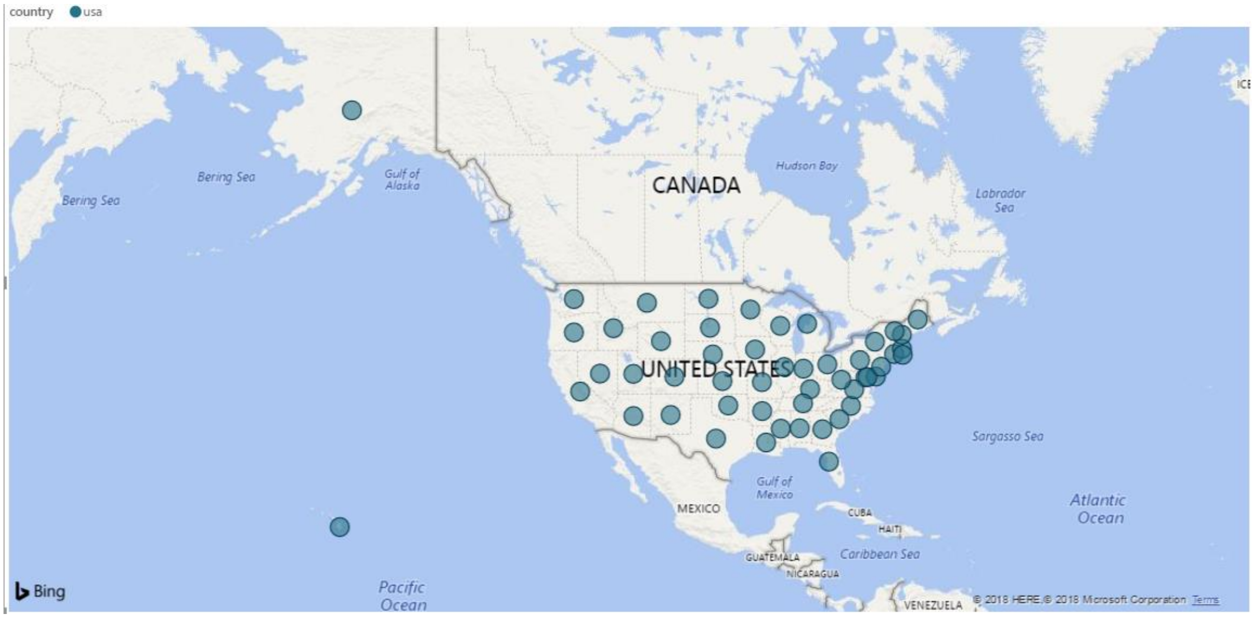
Preview downloaded on Monday

ty	state	country	category	age	gender
omestead	FL	usa	home&garden	*null*	*null*
omestead	FL	usa	home&garden	*null*	*null*
eur d alene	ID	usa	clothing	*36*	F
eur d alene	ID	usa	clothing	*36*	F
ueensbury	NY	usa	computers	*35*	M
ueensbury	NY	usa	computers	*35*	M
ueensbury	NY	usa	movies	*35*	M
ueensbury	NY	usa	movies	*35*	M
nnyvale	CA	usa	shoes	*22*	M
nnyvale	CA	usa	shoes	*22*	M
n diego	CA	usa	shoes	*28*	F
n diego	CA	usa	shoes	*28*	F
arlottesville	VA	usa	computers	*26*	F
arlottesville	VA	usa	computers	*26*	F
arrish	FL	usa	shoes	*45*	F
arrish	FL	usa	shoes	*45*	F
llas	TX	usa	home&garden	*28*	F
llas	TX	usa	home&garden	*28*	F
pitol heights	MD	usa	shoes	*44*	F
pitol heights	MD	usa	shoes	*44*	F
pitol heights	MD	usa	shoes	*44*	F
pitol heights	MD	usa	shoes	*44*	F

PowerBI Visualization 1:



PowerBI Visualization 2:


country ● usa
Bering Sea
Bering Sea
Gulf of Alaska
CANADA
Hudson Bay
Labrador Sea
ICE
UNITED STATES
Sargasso Sea
Gulf of Mexico
MEXICO
CUBA
HAITI
Atlantic Ocean
Caribbean Sea
GUATEMALA
NICARAGUA
VENEZUELA
Pacific Ocean
Bing
© 2018 HERE, © 2018 Microsoft Corporation Terms