Nan WANG

big data --- 2016 fall

AWS implementation i

**Part A: Contextual Advertising in HIVE**

**Problem Statement:**

This project is to use Amazon’s sample HIVE/STREAMING/CUSTOMER JAR programs to do Contextual advertising using HIVE by generating a statistically inspired model so that the best online advertisement could be chosen to show in a given context from impression and click logs.

**Analysis Roadmap:**

1. The ad serving machines produce two types of log files: impression logs and click logs. Every time an advertisement is displayed to a customer, an entry would be added to the impression log. Every time a customer clicks on an advertisement, an entry would be added to the click log;

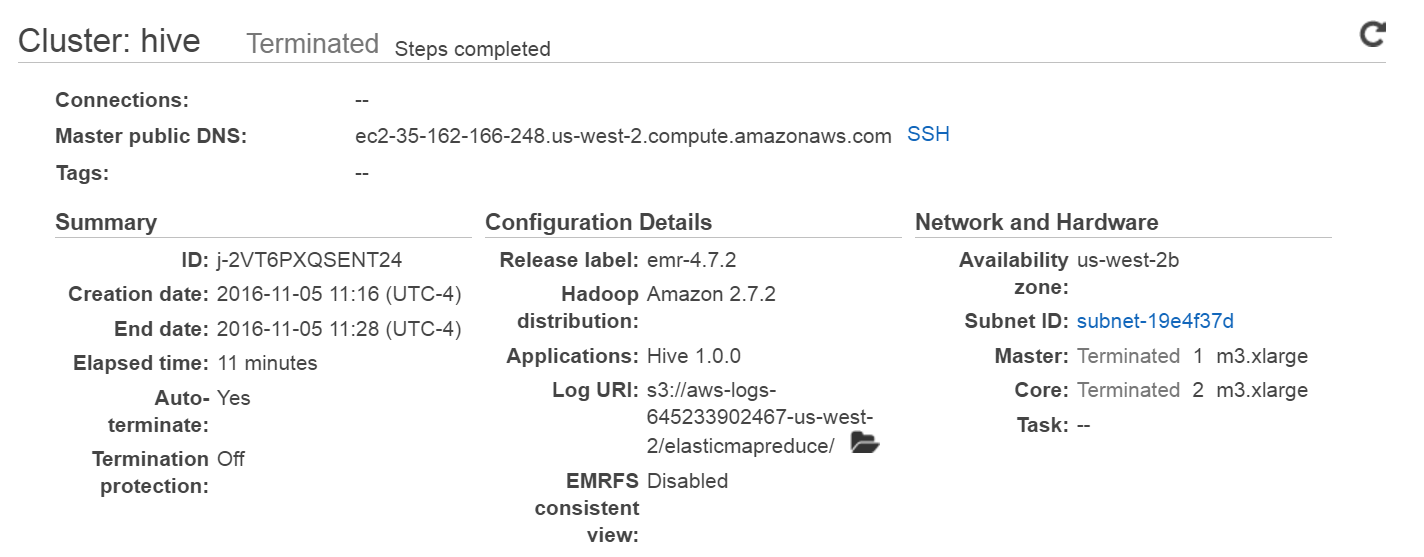
2. A single table that specifies if there was a click for a specific ad and information about that click has been created as our input;

3. HIVE commands are verified to create a script, which later are stored and used to create an execution job flow;

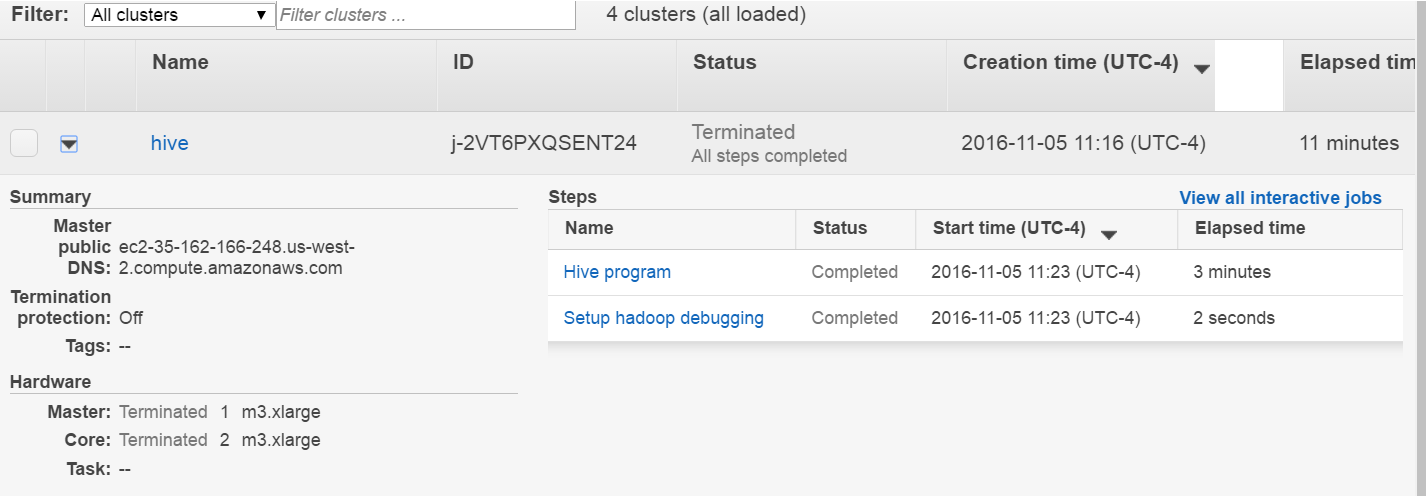
**Managerial Conclusion:**

 The result of this job flow is a table in Amazon S3 that could be used by an analyst to develop and test a model for contextual advertising. The Hive statements collected by the analyst could be used within a job flow to generate a model file. The analyst could upload the file to Amazon S3 and thus make it available to ad-server machines to serve ads contextually.

**Relevant Screenshots:**



**Figure 1: Configuration of HIVE program**



**Figure 2: Successfully Execute Scripts**

**Part B: Word Count Using a Python-Based Map/Reduce Code**

**Problem Statement:**

This project is to use Hadoop Streaming to count the number of times the words occur within a text collection.

**Analysis Roadmap: *(Python code can be found here:*** https://github.com/MiyainNYC/Big-Data-Course-Projects/blob/master/amazon-wordoccurrence.py***)***

1. To count the occurrence of words we need a map function that iterates through its input emitting word, count pairs, which would be implemented using Python;

2. The word count example is using the built-in reducer called aggregate. This reducer adds up the counts of words being emitted by the word-Splitter map function. It knows to use data type Long from the prefix on the words.

3. In Python, tokens (separate words) have been lowered before counted;

**Managerial Recommendation**

Although the job flow creates and executes correctly and creates output the make sense, it is not clear how the input looks.

**Code:**



**Result:**



**Part C: Analyzing CloudFront Log Files using a custom JAR (Java)**

**Problem Statement:**

This project is to generate usage reports containing total traffic volume, object popularity report, a breakdown of traffic by client IPs and edge location by using Amazon Elastic MapReduce and LogAnalyzer application.

**Analysis Roadmap:**

1. [Amazon CloudFront](http://aws.amazon.com/cloudfront) is a web service that delivers your content using a global network of edge locations. Amazon CloudFront can be configured to collect access logs by updating the distribution configuration;

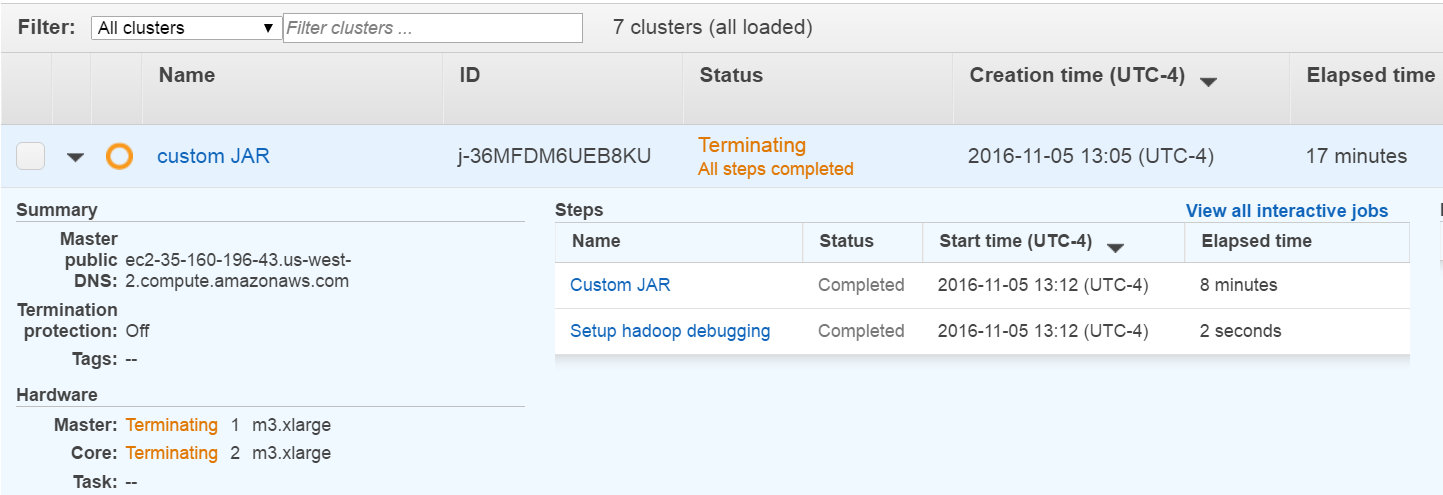
2. The LogAnalyzer for Amazon CloudFront analyzes the service's raw log files to produce a series of reports that answer business questions commonly asked by content owner.

3. The Object Popularity Report shows how many times each of objects are requested. The Client IP report shows the traffic from each different Client IP that made a request for the content. The Edge Location Report shows the total number of traffic delivered through each edge location.  Each report measures traffic in three ways: the total number of requests, the total number of bytes transferred, and the number of request broken down by HTTP response code.

**Managerial Recommendation**

This application produces four sets of reports based on Amazon CloudFront access logs. The Overall Volume Report displays total amount of traffic delivered by CloudFront over the course of whatever period.

**Figure 1: Executing Scripts**



**Figure 2: Results**



**Part D: Apache Log Reports (Pig Script)**

**Problem Statement:**

This project is to determine which of online ad campaigns of a popular e-commerce drive the most traffic to a specific online store by analyzing Apache web logs to see how people find the site.

**Analysis Roadmap:**

1. The web server logs, too large to be imported into a MySQL database and being in a non-relational format, need Hadoop framework.

2. Import data from Amazon S3 and create an Amazon EMR cluster, then connect to the master node of the cluster, where Hive would be executed to query the Apache logs using a simplified SQL syntax;

3. <http://search.pch.com>, <http://www.google.co.zw> and <http://search.virginmedia.com> being the top one external reference, the store could target its customers from those recourses;

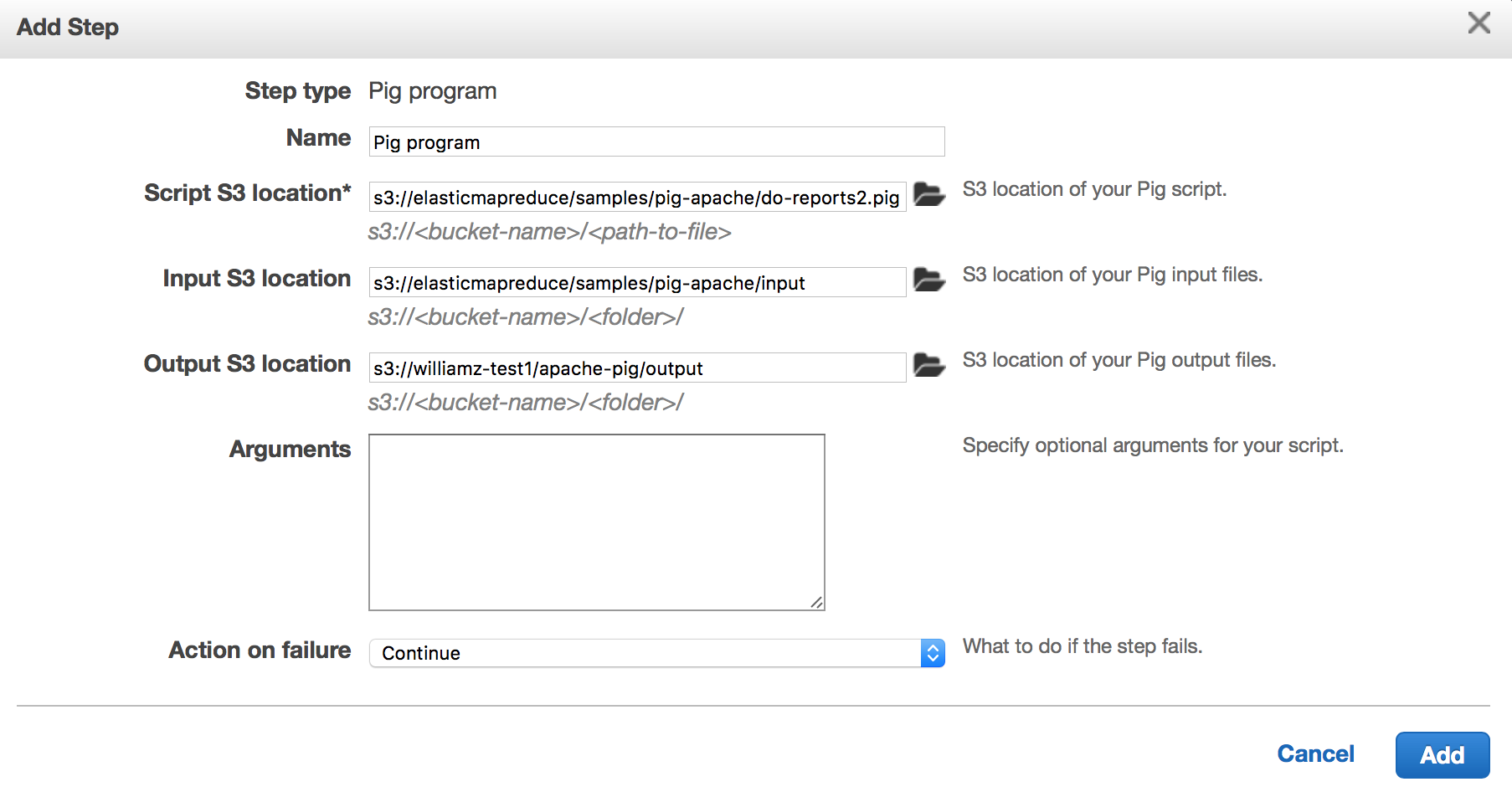
4. 66.249.67.\* being the top frequent IP address visiting the store.

5. “Views” turns out to top search term, followed by “login” and “search”.

**Managerial Recommendation**

The store should utilize information of top 50 IP address and top 50 search terms to know its customer better and therefore brand themselves more effectively.

**Figure 1: Configuration**



**Figure 2: Result**

