Source: The following hands-on tutorial is modified from AWS. Big Data on AWS 3.2 (EN): Student Guide. AWS/Gilmore. VitalBook file.

Lab 2 - Processing Server Logs with Hive on Amazon EMR

In this lab, you will use Apache Hive and Amazon EMR to process logs uploaded to Amazon S3 from servers that support online advertising. The logs are processed and the resulting data is stored in a collection of query-ready, relational database tables persisted in Amazon S3.

There are two types of log files: **impression** logs and **click** logs. Each time an advertisement is displayed to a customer, an entry is added to the impression log. Each time a customer clicks an advertisement, an entry is added to the click-stream log.

In this example, the log data is stored in an Amazon S3 bucket named /us-west-2-aws-training/, in a folder named /awsu-ilt/AWS-200-BIG/v3.0/data/lab-4-hive/data/tables/impressions/. The /impressions/ folder will contain partitioned data (partitioned data files facilitate query performance by allowing your query to download only relevant data from Amazon S3). The naming syntax for the partitioned table is: [Partition column]=[Partition value]. For example: dt=2009-04-13-05, where dt (date/time) is the partition column name and 2009-04-13-05 is the partition value.

Task 1.1: Creating an Output Bucket

1. In the AWS Management Console, on the Services menu, click S3.
2. Click Create bucket, e.g., hive-bucket<yourinitials> (note: Amazon S3 bucket names used by Amazon EMR must be lower case and cannot contain spaces, underscores, or periods.)
3. Click Create.
4. Click your bucket name.
5. Click Create folder. Type **output** and then press ENTER. (Ensure that it is typed in lower-case.)

Task 1.2: Launching an Amazon EMR Cluster

1. On the Services menu, click EMR. Click Create cluster.
2. At the top of the screen, next to Create Cluster - Quick Options, **click Go to advanced options**.
    a. On the Step 1: Software and Steps page, in the Software Configuration section:
        i. For Release, click emr-4.3.0 (EMR Release).
        ii. Ensure these applications are selected: Hadoop, Hive
        iii. Clear (untick) all other applications (eg Hue, Pig)
    b. Notice at the bottom of the screen that Auto-terminate cluster after the last step is completed is automatically deselected, which creates a long-running cluster.

3. Click Next.
4. On the Step 2: Hardware page, in the Hardware Configuration section:
    a. For Master, click the pencil edit icon and select m4.xlarge.
    b. For Core, click the pencil edit icon and select m4.xlarge, and in the Instance Count column, a 2 appears.
5. Click Next.
6. On the Step 3: General Cluster Settings page, in the General Options section:
    a. For Cluster name, type Hive EMR cluster
    b. For Logging S3 folder, type the following (replacing <yourinitials>* [including the angle brackets] with the lower case first letter of your first and last name):

        s3://hive-bucket<yourinitials>/logs/

    c. Clear (turn off) Debugging.
    d. Click Next.

On the Step 4: Security page, in the Security Options section, for EC2 key pair, click your generated key pair. If haven't generated your key, see the last page '**Create an Amazon EC2 Key Pair and PEM File'** for the related instructions.

7. Click Create cluster.
8. Under Security and Access, next to Security groups for Master, click the link for the master security group (ElasticMapReduce-master).
9. In the bottom pane, click the Inbound tab. Click Edit. Click Add Rule.
    a. For Type, click SSH.
    b. For Source, click Anywhere.
    c. Click Save.
10. On the Services menu, click EMR. Click Hive EMR cluster.
11. The Cluster list page will show the status of your cluster. The status will change from Starting to Running to Waiting (there are no bootstrap actions for this cluster). When the cluster is in the Waiting state, it is ready for use. This will take up to 10 minutes. Wait until the cluster is in the Waiting state. You may need to click the Refresh icon and update the status manually.

Task 2: Connecting to the Hadoop Master Node Using SSH

1. You will retrieve the public DNS name of the EMR cluster. On the Services menu, click EMR.
2. On the Cluster list page, click Hive EMR cluster.
3. On the Cluster list page, the Master public DNS appears in the cluster details. Copy the master public DNS name to the clipboard. You may also paste the value in a text document for later retrieval.
4. Connecting to the Hadoop Master Node Using Windows
    a. You will use a PuTTY client to connect to your cluster, with your private key used for authentication.

b. On the Basic options for your PuTTY session screen, for Host Name (or IP address), type hadoop@ and then paste the DNS hostname of your master node that you copied earlier.
c. In the Category section, click Connection. On the Options for controlling the connection screen:
    i. For Seconds between keepalives (0 to turn off), type 60
    ii. In the Low-level TCP connection options section, select Enable TCP keepalives (SO_KEEPALIVE option) check box.
d. In the Category section, expand SSH and click Auth. For Private key file for authentication, click Browse, navigate to the folder containing your .PPK file, select the file, and click Open. If prompted to cache the server's host key, click Yes.

Task 3.1: Running Hive Interactively.

You will create the Hive tables that form your simple data warehouse. Before creating the Hive tables, you start an interactive Hive session with the master node and reference the JSON serializer/de-serializer (SerDe) used to read the log content.

1. In your SSH client, paste the following commands. This creates a logging directory that will be used by Hive.
```
sudo chown hadoop -R /var/log/hive
mkdir /var/log/hive/user
mkdir /var/log/hive/user/hadoop
```
2. In your SSH client, paste the following command (replace hive-bucket<YourInitials> (including the angle brackets) with your Amazon S3 bucket name).
```
hive -d SAMPLE=s3://aws-tc-largeobjects/AWS-200-
BIG/v3.1/lab-4-hive/data -d DAY=2009- 04-13 -d HOUR=08 -d
NEXT_DAY=2009-04-13 -d NEXT_HOUR=09 -d OUTPUT=s3://hive-
bucket<YourInitials>/output/
```
3. Output: You should see a hive> prompt.
4. The version of Hive installed in Amazon EMR enables you to reference resources, such as JAR files, located in Amazon S3. Copy and paste the following command to configure Hive to use the JSON SerDe JAR file to read the log data.
```
ADD JAR ${SAMPLE}/libs/jsonserde.jar;
```

Task 3.2: Creating Tables Using Hive

1. Copy the following Hive statement and paste it into your SSH client in order to create the external impressions table from the logs stored in Amazon S3.
```
CREATE EXTERNAL TABLE impressions (requestBeginTime string,
adId string, impressionId string, referrer string, userAgent
string, userCookie string, ip string) PARTITIONED BY (dt
string) ROW FORMAT serde
'com.amazon.elasticmapreduce.JsonSerde' with serdeproperties
```

```
('paths'='requestBeginTime, adId, impressionId, referrer,
userAgent, userCookie, ip') LOCATION
'${SAMPLE}/tables/impressions';
```

2. Indicate to Hive the existence of a single partition (dt='2009-04-13-08-05') by copying the following statement and pasting it into your SSH client:
```
ALTER TABLE impressions ADD PARTITION (dt='2009-04-13-08-
05');
```
3. To inspect the log data and return all partitions, run the following command:
```
MSCK REPAIR TABLE impressions;
```
4. To create the external clicks table from the click-stream logs stored in Amazon S3, run following command:
```
CREATE EXTERNAL TABLE clicks (impressionId string)
PARTITIONED BY (dt string) ROW FORMAT SERDE
'com.amazon.elasticmapreduce.JsonSerde' WITH SERDEPROPERTIES
('paths'='impressionId') LOCATION '${SAMPLE}/tables/clicks';
```
5. To return all partitions from the click-stream log data, run the following command:
```
MSCK REPAIR TABLE clicks;
```
6. Type the following statement to verify the creation of your tables. The output should return both external tables stored in Amazon S3: clicks and impressions.
```
show tables;
```

Task 4: Joining Tables Using Hive

1. Run the following command to create the external joined table (impressions table with the clicks table).
```
CREATE EXTERNAL TABLE joined_impressions (requestBeginTime
string, adId string, impressionId string, referrer string,
userAgent string, userCookie string, ip string, clicked
Boolean) PARTITIONED BY (day string, hour string) STORED AS
SEQUENCEFILE LOCATION '${OUTPUT}/tables/joined_impressions';
```
2. Run the following statement to create a temporary table (tmp_impressions) in the job flow's local HDFS partition to store intermediate data for the specific time duration.
```
CREATE TABLE tmp_impressions (requestBeginTime string, adId
string, impressionId string, referrer string, userAgent
string, userCookie string, ip string) STORED AS SEQUENCEFILE;
```
3. Run this command to insert impression log data for the time duration referenced. Because the impressions table is partitioned, only the partitions relevant to the specified time duration are read.
```
INSERT OVERWRITE TABLE tmp_impressions SELECT
from_unixtime(cast((cast(i.requestBeginTime as bigint) /
1000) as int)) requestBeginTime, i.adId, i.impressionId,
i.referrer, i.userAgent, i.userCookie, i.ip FROM impressions
i WHERE i.dt >= '${DAY}-${HOUR}-00' and i.dt < '${NEXT_DAY}-
${NEXT_HOUR}-00';
```

Notice that the Hive statement is converted into MapReduce jobs, all of which are mapper tasks. MapReduce jobs transform source data into a desired output using mappers and reducers. Input data is provided to the mapper as key/value pairs.

4. Run the following statement to create the temporary clicks table (named tmp_clicks) in HDFS.
   ```
   CREATE TABLE tmp_clicks ( impressionId string ) STORED AS SEQUENCEFILE;
   ```
5. To insert data into the tmp_clicks table, run:
   ```
   INSERT OVERWRITE TABLE tmp_clicks SELECT impressionId FROM clicks c WHERE c.dt >= '${DAY}-${HOUR}-00' AND c.dt < '${NEXT_DAY}-${NEXT_HOUR}-20';
   ```
6. For the clicks table, the time period is extended by 20 minutes. This allows for clicks that occurred up to 20 minutes after the impression and preserves any impressions that did not result in a click.
7. To create a left outer join of tmp_clicks and tmp_impressions that writes the resulting data set to the joined_impressions table in your S3 output bucket, copy:
   ```
   INSERT OVERWRITE TABLE joined_impressions PARTITION (day='${DAY}', hour='${HOUR}') SELECT i.requestBeginTime, i.adId, i.impressionId, i.referrer, i.userAgent, i.userCookie, i.ip, (c.impressionId is not null) clicked FROM tmp_impressions i LEFT OUTER JOIN tmp_clicks c ON i.impressionId=c.impressionId;
   ```
8. Type the following statement to update the metadata store with information about all partitions in the joined_impressions table.
   ```
   MSCK REPAIR TABLE joined_impressions;
   ```
9. Type the following statement to query the joined_impressions table. (Note the 10-limit parameter restricts the result set to 10 records.)
   ```
   select * from joined_impressions limit 10;
   ```
10. Switch back to the AWS Management Console in your browser. On the Services menu, click S3.Click hive-bucket<YourInitials>.
11. Click output, click tables, click joined_impressions, click day=2009-04-13, and then click hour=08. Notice the resulting table data stored as an S3 object.

Create an Amazon EC2 Key Pair and PEM File

Amazon EMR uses an Amazon Elastic Compute Cloud (Amazon EC2) key pair to ensure that you alone have access to the instances that you launch. The PEM file associated with this key pair is required to ssh directly to the master node of the cluster.

To create an Amazon EC2 key pair:

- Go to the Amazon EC2 console
- In the Navigation pane, click Key Pairs
- On the Key Pairs page, click Create Key Pair
- In the Create Key Pair dialog box, enter a name for your key pair, such as, mykeypair
- Click Create
- Save the resulting PEM file in a safe location

Modify Your PEM File

Amazon Elastic MapReduce (Amazon EMR) enables you to work interactively with your cluster, allowing you to test cluster steps or troubleshoot your cluster environment. You use your PEM file to authenticate to the master node. The PEM file requires a modification based on the tool you use that supports your operating system.

To modify your credentials file:

- Download PuTTYgen.exe to your computer from:
- http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- Launch PuTTYgen
- Click Load
- Select the PEM file you created earlier
- Click Open
- Click OK on the PuTTYgen Notice telling you the key was successfully imported
- Enter a pass phrase in the Key passphrase field
- Click Save private key to save the key in the PPK format
- Enter a name for your PuTTY private key, such as, mykeypair.ppk
- Click Save
- Exit the PuTTYgen application

Your credentials file is now modified to allow you to log in directly to the master node of your running cluster.