* **Can we assign default workgroup to a user so he/she need not to change it in settings. As we tried last time, were not able to achieve. We were able to give access to workgroup but faced issues while setting default workgroup for user.**

MIND: Please check the attached document and policy.

* **What is maximum concurrency that can be requested for Athena, we currently have 50 for our production setup.**

MIND: Its case by case basis as per AWS.

* **For EMR, as suggested that concurrency can be increased by power of 2, does that require any additional hardware. Also, what is relation between concurrency and instance sizing.**

MIND: Yes, memory needs to be increased for increasing the concurrency. Memory consumption depends on the query, so you need to benchmark with your queries.

* **For usage analysis, viewing historic queries in console is not the best way I would say. Please suggest if there is any other way**.

MIND: The query history section in the console has the following columns-query String ,eventTime, encryption, state, runtime, data scanned.Out of these queryString, eventTime, encryption can be get using Athena StartQueryExecution API. Currently, there is no API to get the rest of the fields. We have asked AWS to raise a feature request for the same.

 You wanted to retrieve the query history of Athena programmatically. I have checked the commands for this and I am glad to inform you that you can get the query details of a particular query via both AWS CLI (Command Line Interface) and AWS API.  
  
Below are the steps to retrieve and save query history programmatically:-  
  
1) Use Athena ListQueryExecutions API [1] action or the list-query-executions CLI command [2] to retrieve the query IDs.  
      --> It provides a list of available query execution IDs in the specified workgroup  
  
2) Use the Athena GetQueryExecution API [3] action or the get-query-execution CLI [4] command to retrieve information about each query based on its ID.  
      -->This will give you to get the information of a particular/single queryid.  
  
      Note : You can use the Boto3 with python or any language to achieve this.  
  
That being said, you can use both above methods to fetch the information of all the queries executed in the specified workgroup programmatically.

References:-  
[1] <https://docs.aws.amazon.com/athena/latest/APIReference/API_ListQueryExecutions.html>  
[2] <https://docs.aws.amazon.com/cli/latest/reference/athena/list-query-executions.html>  
[3] <https://docs.aws.amazon.com/athena/latest/APIReference/API_GetQueryExecution.html>  
[4] <https://docs.aws.amazon.com/cli/latest/reference/athena/get-query-execution.html>

* **Also, have some queries regarding data size. Since files that we provided were already in parquet format, there is considerable difference in size after partitioning on PoC data. Just wondering if there are any compression settings that are resulting in size difference.**

MIND: Your assumption is correct, we just read your “aeps\_trans\_res” files with 33GB size and written to another folder with snappy compression (no partitioning), with same no. of files in the output folder and the size increase considerably by 10GB. Please share your compression type used while creating the Parquet format.

* **Cost of the data transformation job:**
  + Data: sm-csv-data/aeps\_trans\_res (33 GB)
  + Cluster:  1Master : m4.xlage , 10 Core Node ( c4.2xlarge ), 10 Task Node (c4.2xlarge) , All Spot instance used.
  + **Actual Cost: 1.10 USD**
* **On running query, like select \* from tablename, then does the data move from S3 to EMR?**

MIND: Yes, the selected objects are moved from s3 to EMR using EMRFS.

* **On running a query having join operation of two tables, how many catalog requests will be made to Glue Catalog?**

MIND: The glue catalog stores only the schema of the table and the requests are made per table so for join operation of two tables, two catalog requests will be counted.

* **Is the no. of get requests that go to s3 dependent on no. of files or the size of the files?**

MIND: Yes, no. of s3 get requests are based on no. of objects fetched and in a single get request the entire object is fetched. That means if fetched data is in single file that will be one get request and if the same data is getting fetched from 10 files will that be 10 get requests.

* **Can we assign default workgroup to a user so he/she need not to change it in settings. As we tried last time, were not able to achieve. We were able to give access to workgroup but faced issues while setting default workgroup for user.**

Check how to use Athena Workgroups (change the default workgroup of user) for tracking the Athena usage per user.

MIND: You can create different workgroups and assign these workgroups to user via

IAM Policies.

<https://docs.aws.amazon.com/athena/latest/ug/example-policies-workgroup.html>

New workgroups "test", "test2" are created in spicepay account and a managed policy is attached with data\_lake\_user to restrict its access only to "test" workgroup. Now data\_lake\_user can run queries only in "test", not in "primary" and "test2".

* **What is maximum concurrency that can be requested for Athena, we curraently have 50 for our production setup.**

Check the concurrency (query execution) limit of Athena and EMR.

MIND: In Athena concurrency quotas are defined as the number of queries that can be submitted to the service concurrently. By default, this value is 20 which means 20 queries can be submitted at a time in an account to Athena. This is the soft limit and can be increased by requesting increase in Athena service quota, the raise varies from use-case to use-case. Thing to note is that this does not ensure that they will be executed concurrently, as Athena is a shared service the execution depends on resource availability.

In EMR, every application has its own way of tuning concurrency by some application-

level and yarn-level configurations like for Presto the default concurrency value is 16 and you can increase by power of 2.

Max concurrency is not defined.

 In Athena concurrency quotas are defined as the number of queries that can be submitted to the service concurrently. By default this value is 20 which means 20 queries can be submitted at a time in an account to athena. This is the soft limit and can be increased by reqesting increase in athena service quota, the raise varies from use-case to use-case.

Thing to note is that this does not ensure that they will be executed concurrently, as athena is a shared service the execution depends on resource availability.

In EMR, the step concurrency can be set to a max. of 256 and as far as query execution concurrency is concerned, every application has its own way of tuning concurrency by some application-level and yarn-level configurations, more on this will be reverted by support team.

* **How to get the Athena queries**

MIND: You can get the query logs with execution time and data scanned for a particular Workgroup from the history tab.

In the history tab in Athena console there are logs of queries run in a particular workgroup in last 45 days. These logs contain fields like the query itself, the amount of data scanned by it and run time of query.

* **Security controls for accessing AWS console (restriction based on user, hostname etc. apart from IP based restriction)**

MIND: Currently only IP based restrictions can be applied.

Security controls for accessing AWS console (restriction based on user, hostname etc. apart from IP based restriction)

  Finding about the above point are as follows.

1. User login can be restricted using Source IP Address by using aws:SourceIp  condition.

2. It can also be restricted on region basis.

3. Restricting user login on the basis of Geographical location is not possible. It can only be through IP Address

As of now only these two methods are available for restricting user.

* **Cost of the data transformation job:**
  + Data: sm-csv-data/aeps\_trans\_res (33 GB)
  + Cluster:  1Master : m4.xlage , 10 Core Node ( c4.2xlarge ), 10 Task Node (c4.2xlarge) , All Spot instance used.
  + Execution Time: 10 Minute
  + Cost: USD **0.413**(running cluster compute cost)
* **Regarding User access: User level access/access restriction for querying data**

**through EMR cluster. For Athena, you have already suggested using data lakes.**

MIND: We have shown you user level access/restrictions for Athena using Data Lake Formation. Integration of EMR cluster with Data Lake Formation is in beta.

<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-lake-formation.html>

You can restrict access in EMR to glue catalogue tables/columns by assigning role to EMR cluster which has specific permission at table/column level. But this access is not at user level.

* **Usage pattern: Ability to identify and analyze per user usage patter for Athena as**

**well as EMR cluster.**

MIND: Yes, you can identify and track data scanned by user queries in Athena. Please find attached the document for the same. For EMR, you are not charged for the data scan, the following will be the charges in different cases:

Case 1. Data is in EMR- No charges

Case 2. EMR and S3 bucket in same region- Data Retrieval charges (List and Get)

Case 3. EMR and S3 bucket in different regions- Data transfer charges + Data Retrieval

charges (List and Get) Catalog request charges for querying glue catalog which are nominal will be applicable for all the cases. They are free for the first million requests per month and then $1.00 per million requests above 1M in a month.

* **Possibility of incremental changes.**

MIND: Yes, using Apache Hudi (issue is, it will not do the changes in the source dataset, it will create new dataset with changes)

<https://aws.amazon.com/blogs/aws/new-insert-update-delete-data-on-s3-with-amazon-emr-and-apache-hudi/>

* **Performance difference for querying data through EMR and Athena under stress testing.**

MIND: We couldn’t find any performance benchmarking of EMR and Athena, we need to test it yourself. You can find below tips for optimizing performance for Athena.

<https://aws.amazon.com/blogs/big-data/top-10-performance-tuning-tips-for-amazon-athena/>

* **Any tool available for performance testing using parallel query execution.**

MIND: No, out of the box tool available.

* **Also, as per my understanding, even if we are using EMR to query data, indirectly we are going to Athena only. This also means that we have s ame limits as of Athena even querying through EMR. Please correct if mistaken.**

MIND: No, this is not correct. EMR has own it set of tools, like Spark SQL, Hive, Presto etc. for querying data in HDFS or S3 (EMRFS). These tools don’t use Athena (which is a

AWS managed querying engine based on Presto)