**AWS- on the path of making ETL simpler with EMR and Glue services**

AWS Glue service is a server-less ETL(Extraction, Transformation, Loading) tool based on pay as you go model with very little infrastructure required to be setup. The effort required in execution and monitoring of ETL jobs is close to none. In the case where data is structured, Glue crawlers can be sed to infer schema, identify file formats and populate the metadata content into Glue’s Data Catalogue. On the basis of specific criteria, Glue automatically generates code in Python/Scala and this code rune on Spark engine which gets automatically configured. Further, Glue gives a feature of monitoring and scheduling of these jobs in easy-to-use GUI.

EMR is a big data service offered by AWS with the USP of analyzing and processing of high volumes of data. In this managed service the developers configure their EC2 machines on which Hadoop components are installed giving full flexibility to its use. Use cases range variedly from data scientists using it to run ML jobs, data analysts using it to run queries on Presto to building streaming applications using [Spark](https://aws.amazon.com/emr/features/spark/) and Kinesis. All in all, one can replace Glue with EMR but it is vice versa is not possible as EMR is having a lot more capabilities than Glue.

# **Glue service Components:**

* **Data catalogs**

This component of Glue namely data catalog is responsible for holding the structure and the metadata of the dataset.

* **Databases**

This Glue component is used for creating and accessing the database for different targets and sources.

* **Tables**

Single or multiple tables are created in the database, these are then used by destination and source.

* **Crawlers and Classifiers**

The function of the crawler is retrieval of data from source by making use of built in and custom classifiers. In the output section, metadata tables are created by the crawlers and classifiers.

* **Job**s

Any business logic which is written to complete ETL process is called a job. In the backend, it is basically Pyspark, Scala and Apache spark code for doing the required transformations in the dataset.

* **Triggers**

Although jobs can be started on demand but to begin with a specific ETL job on a specific time or in response to an event, triggers are required.

* **Development Endpoints**

For testing, development and debugging of job, an endpoint is needed by the developers. This component of Glue is very useful to achieve this and provide a secure development environment.

* **Additional Services in EMR:**
* **Hadoop**

It is a big data open-source framework which is used to store and run applications having massive amounts of data stores, require enormous power of processing and for storing data and can deal with big number of concurrent jobs.

* **Tez**

Tez is used to increase the speed of MapReduce framework while maintaining its scalability feature. It is used as an extended framework while making big data applications which require high batch performance and are interactive. YARN in Hadoop is responsible to coordinate this.

* **Zookeeper**

In any distributed system, there are a large number of clusters and a central service is required to manage naming, configuration information and syncing activities along all clusters. This is where Zookeeper comes into place. It is open-source project by Apache with the objective of making it easier to manage and propagate changes in a reliable manner.

* **Hive**

Hive is made on Apache Hadoop, it is used as a data warehouse to perform query and analysis on big data. It proves a SQL like interface for performing the query operations across multiple file systems and databases which can be integrated with Hadoop.

* **Spark Core**

This is the most important API of spark which is used for writing code for big data processing in Python(Pyspark), Java and Scala languages.

* **Spark SQL**

This is an easier path for non-developers who are keen to use spark. This is basically an programming interface which is used to create Hive based SQL queries which are then translated to spark core in a seamless manner.

* **Hue**

This is a web application giving users an interface to analyze data using SQL. Natively, it is set to work with Presto, Hive and SparkSQL.

* **Zeppelin**

It is a web notebook tool which is open source. It is required for running the data pipeline operations in orchestration using technologies like Spark, SparkSQL, Bash and Hive. It also offers feature to collaborate , do graphical visualizations and basic level scheduling of queries.

* **Ganglia**

It is a software to see live and recorded statistics like load averages of CPU and network utilization etc. The level of scalability and distributed monitoring in clusters and networks of high computing systems is commendable.

* **TensorFlow**

It is a widely used software library which is open-source and mainly used to perform numerical computations that require high performances. This is commonly used in the fields of ML and deep learning.

* **Pig**

This high-level application is used to create programs for scripting which run on Hadoop, there is option to add it in EMR cluster.

* **Sqoop**

This console-mode application is based on Java and is used to transfer big data between non-Hadoop and Hadoop sources of data like data warehouses, NoSQL databases and relational databases.

* **Advantages of Glue over EMR**
* **Serverless service**

Being a bigdata integration service which is truly serverless, Glue saves you from the task of building and maintenance of infra. Amazon itself provides for and manages the servers and one does not have to do all the configurations like in EMR.

* **Automation of ETL code**

AWS Glue automatically generates the ETL pipeline code in Python and Scala depending on the data sources and destinations selected. This feature helps in operation of data integrations and also in the process of parallelizing big data workloads.

* **Data visibility and clarity increased**
* This is advantage of AWS Glue Data Catalog which acts as a metadata repository which stores all the information for the data sources be it RDS, Amazon S3, or Amazon Redshift, it helps to keep a tab on all the data assets.
* **Endpoints for developers**

For users preferring to create their custom ETL scripts manually, the AWS Glue gives the facility of developer endpoints for facilitating the complete process of development.

* **Scheduling of jobs**

AWS Glue service give easy-to-use interface and tools for creation of jobs and following them up on the basis of some schedule, event-based triggers and also on-demand.

* **Paying on the go**

The Glue service does not force one to get into long-term commitments through any subscription plans but one can minimize the costs by only making payments of the services or resources as per one’s usage needs.

* **Limitations of Glue over EMR**
* **Technical know-how important**

Certain aspects of AWS Glue service are not friendly for non-tech users. Like, the tasks are run usually on Spark engine which is why one has to be well versed with it to decode the errors in any ETL job. Also, knowledge of Python and Scala is undoubtedly the central part which is not the case with EMR.

* **Dual language support**

Python and Scala are the languages currently supported by AWS Glue so to customizing the ETL code written in any other language is not possible. Whereas in ETL Ruby, Pearl, PHP, Nodejs and C++ is there.

* **Limited set of integrations**

AWS Glue is designed only to work with other AWS services. Which means it is not possible to integrate it with third party tools or platforms lying outside of AWS ecosystem.

In conclusion, one should note that as AWS Glue is serverless so it tends to be on a little costlier side than AWS EMR. On comparing similar configurations of cluster across the two, AWS Glue is more expensive but not by a very large margin.