Bigdata Pipeline with AWS

Author: Diksha Singh Tomer
Computer and Science Engineering
Banasthali University, India

Introduction

As we become a more digital society, the amount of data being created and collected is growing and accelerating significantly. Analysis of this ever-growing data becomes a challenge with traditional analytical tools. We require innovation to bridge the gap between data being generated and data that can be analyzed effectively.

The amount of (raw) data is too huge to store in raw format in any database system or data warehouse. One, definitely, wants to process/extract this data to get useful information to store and for further processing on this information. Data pipeline (or Bigdata pipeline) is a solution to collect, process and move this data from source system to data lakes or warehouses. Later the useful data can be sent to analytical and recommendation systems for further processing on it.

Amazon Web Services (AWS) provides a broad platform of managed services to help you build, secure, and seamlessly scale end-to-end big data applications quickly and with ease. Whether your applications require real-time streaming or batch data processing, AWS provides the infrastructure and tools to tackle your next big data project

This document depicts general terms about data pipeline, its types and some solutions to create such pipelines with AWS services.

Data Pipeline

Data pipelines refer to the general term of **movement of data** from one location to another location. The location from where the flow of data starts is known as a data source, and the destination is called as the data sink.

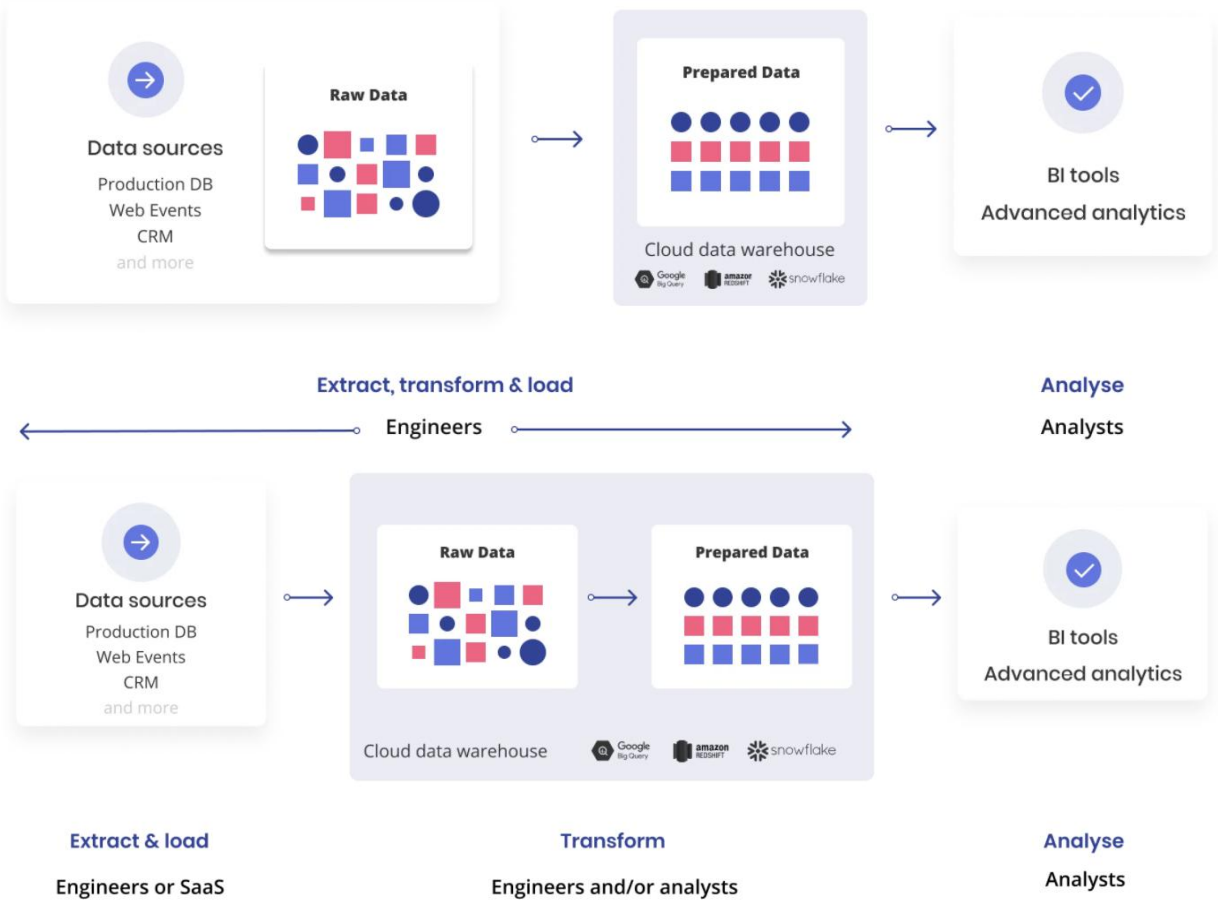
The data sources can be data stored in any of the AWS Big Data locations such as databases, data files, or data warehouses. Such data pipelines are called **batch data pipelines** as the data are already defined and we transfer the data in typical batches.

Whereas there are some data sources such as log files or streaming data from games or real-time application, such type of data is not well defined and may vary in structure as well. Such pipelines are called as **streaming data pipelines**. Streaming data requires a special kind of solution, as we have to consider late data records due to network latency or inconsistent data velocity.

ETL & ELT

We may also like to perform some operations/transformation on the data while it's going from the data source to a data sink, such kind of data pipelines have been given a special kind of names:

- **ETL (Extract Transform Load)**: In ETL data moves from the data source, to staging and then into the warehouse. All transformations are performed before the data is loaded into the warehouse.
- **ELT (Extract Load Transform)**: ELT offers a modern alternative to ETL where analysts load data into the warehouse before transforming it, supporting a more flexible and agile way of working.



Regardless of whether it's ETL or ELT, both processes involve the following three steps:

- **Extract:** Source data is extracted from the original data source in an unstructured format. In traditional ETL processes this data is put into a temporary staging repository.
- **Transform:** Additional transformation must be done to clean and model data before it can be practically useful for other analytics applications.
- **Load:** In the ELT model, data is copied then copied directly into the data warehouse without significant modification. In an ETL model, data would be transformed into a suitable format before loading it into the warehouse.

Batch vs Stream Data Pipeline solutions

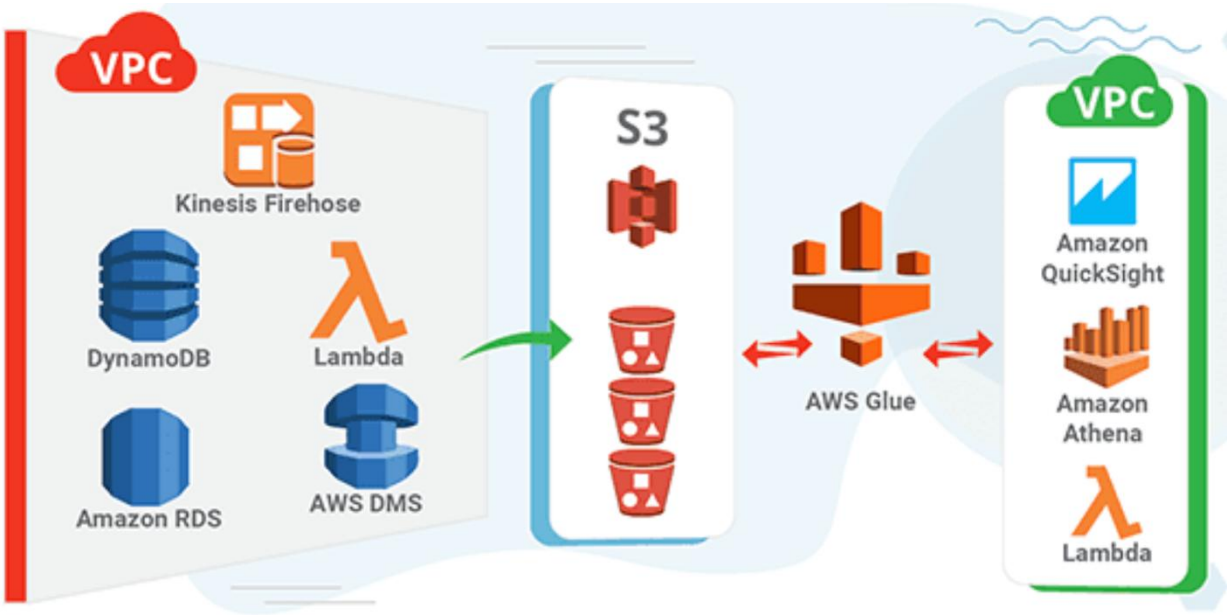
A batch data pipeline solution can be made with following AWS services:

- AWS Glue
- AWS Data Pipeline
- Storage and compute services (DynamoDB, RDS, S3, Redshift, EMR etc.)
- AWS Redshift Spectrum
- AWS Athena

A stream data pipeline solution can be made with following AWS services:

- AWS Kinesis Data Firehose
- AWS Kinesis Data Analytics

The different services creating data pipeline can be shown at stages with following image:



Below, we are talking more about services involved in batch data pipeline solutions.

AWS Glue as ETL service

AWS Glue is a serverless ETL job service. It is a fully managed ETL service that makes it simple and cost-effective to categorize your data, clean it, enrich it, and move it reliably between various data stores and data streams. While using this, we don't have to worry about setting up and managing the underlying infrastructure for running the ETL job.

AWS Glue has three main components:

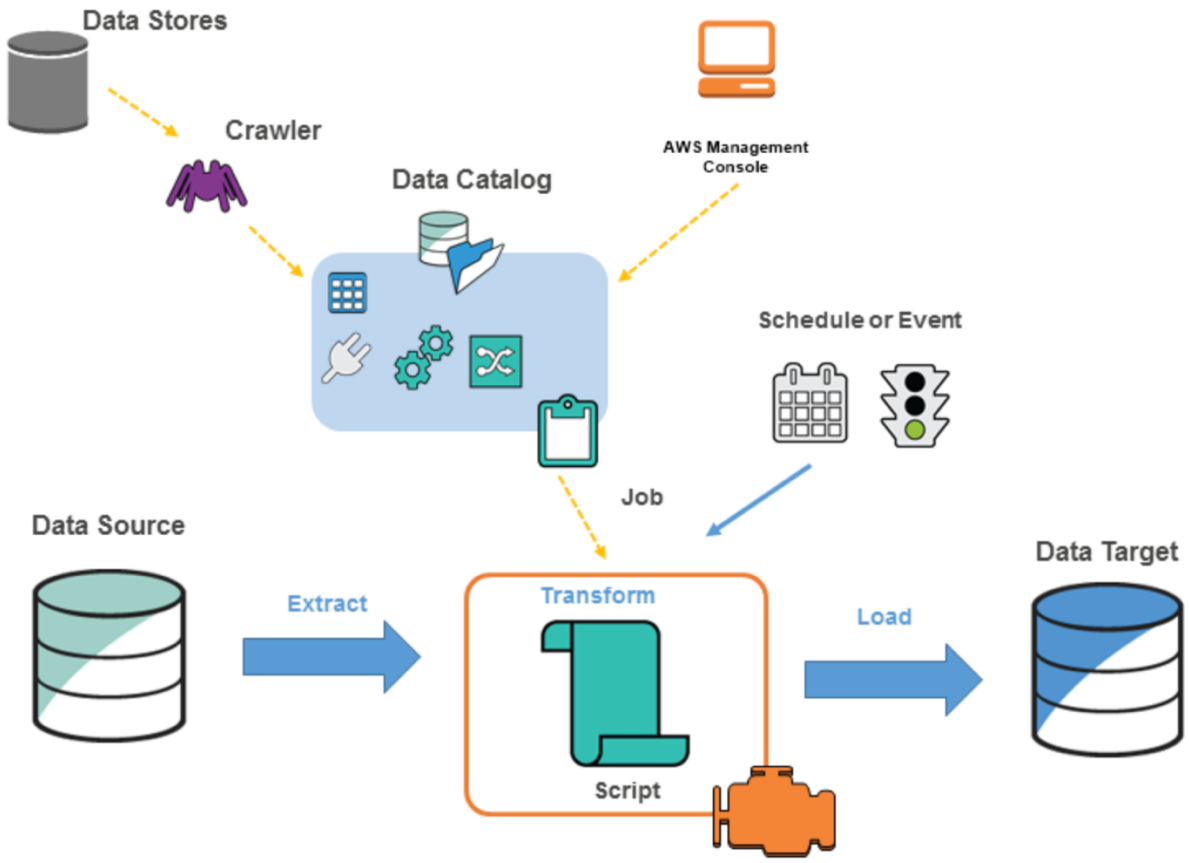
Data Catalog: Glue data catalog contains the reference to the data stores that are used as data sources and data sinks in our extract, transform, load (ETL) that we run via AWS Glue. When we defined a catalog, we need to run a crawler which in turn runs a classifier and infers the schema of the data source and data sink.

Glue provides with built-in classifiers for data formats. Crawlers store data in a metadata store which is an AWS RDS table so that it can be used again and again.

ETL engine: it is the heart of AWS Glue. It performs the most critical task of generating and running the ETL job. In the ETL job generation part, the ETL engine provides us with a comfortable GUI using which we can select any of the data stores in the data catalog and define the source and sink of the ETL job. Glue provides us with some built-in transformation as well. ETL engine generates the corresponding pypspark / scala code. We can edit the ETL job code and customize it as well. ETL engines manage all the infrastructure (launching the infrastructure, underlying execution engine for the code, on-demand job run, cleaning up after the job run). The default execution engine is Apache Spark.

Glue Scheduler: Glue scheduler is more or less like a CRON. We can periodically schedule jobs or run jobs on-demand based on some external triggers, or the job can be triggered via AWS Lambda functions.

The following diagram shows the architecture of an AWS Glue environment:

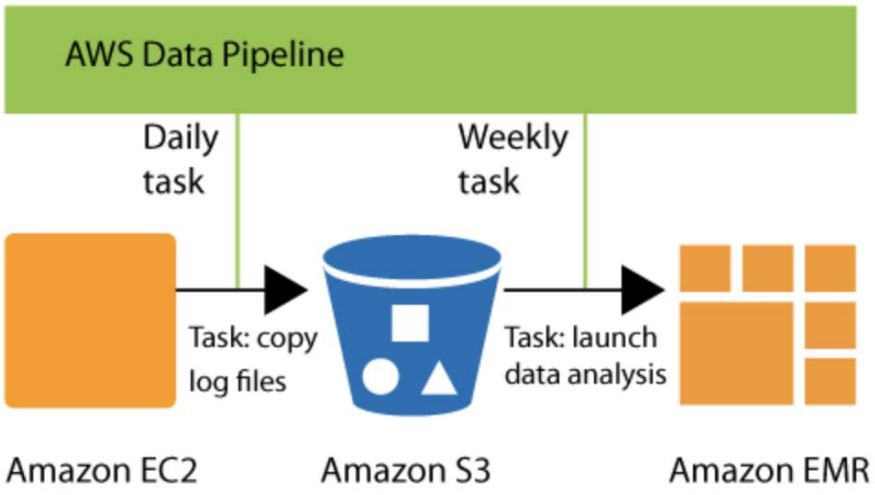


AWS Data Pipeline service

AWS Data Pipeline is a web service from AWS that you can use to automate the movement and transformation of data. With this, you can define **data-driven workflows**, so that tasks can be dependent on the successful completion of previous tasks. This service makes it easy for you, to design extract-transform-

load (ETL) activities using structured and unstructured data, both on-premises and in the cloud, based on your business logic.

An example use case can be that, you can use AWS Data Pipeline to archive your web server's logs to Amazon Simple Storage Service (Amazon S3) each day and then run a weekly Amazon EMR (Amazon EMR) cluster over those logs to generate traffic reports.



AWS Data Pipeline performs several functions such as:

- Guaranteeing resource convenience
- Managing inter-task dependencies
- Retrying transient failures or timeouts in individual tasks
- Making a failure notification system

Some uses of AWS Data Pipeline:

- Copy ETL data to Amazon RedShift
- Analyze ETL Unstructured knowledge
- Load AWS Log knowledge to Amazon Redshift
- Copy knowledge from the user on-premises knowledge store to AWS cloud
- Backup dynamo dB tables to S3 for disaster recovery functions

Storage and compute services

Amazon DynamoDB: Fully managed NoSQL database with fast performance.
Amazon RDS: It is a fully managed relational database that can accommodate large datasets. Has numerous options for the database you want, like AWS aurora, Postgres, MS SQL, MariaDB.
Amazon Redshift: Fully managed petabyte-scale Data Warehouse.
Amazon S3: Low-cost highly-scalable object storage.
Compute Services

Amazon EC2: Service for scalable servers in AWS data center, can be used to build various types of software services.
Amazon EMR: Service for distributed storage and compute over big data, using frameworks such as Hadoop and Apache Spark.

AWS Redshift Spectrum

Redshift Spectrum is an extension of AWS Redshift, the data warehouse service. Redshift Spectrum is a **combination of computing and storage**. It extends the data warehouse to S3 which is inexpensive storage as compared to Redshift, so what we can do now is we can store the less frequently queried data in S3, and whensoever we need to query this data we can make use of Redshift spectrum. This is useful and cost-effective. It uses the compute from the current Redshift cluster only.

AWS Athena as query engine

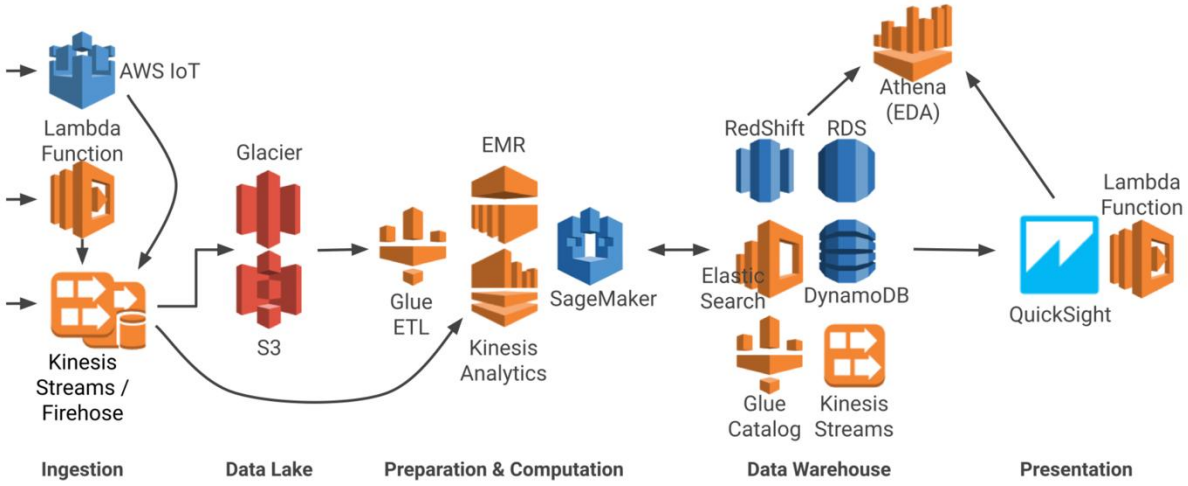
AWS Athena is a query engine that runs over the data stored in S3. Athena charges you per query basis only. Athena runs the presto framework, by Facebook.

In case you don't have a Redshift data warehouse and you want to query the data stored in S3, so Athena is the tool that you are looking for. Athena can even be integrated with your BI tools such as Tableau. Athena saves the need to set up a whole ETL process yourself and to keep the entire computer cluster running for on spot queries to the data.

AWS Kinesis Data Firehose as streaming data pipeline solution

It is a service that serves as a tool for the ingestion of streaming data from various data sources to the data sinks in a secure way. It can handle an ample amount of data stream workloads and scale accordingly. It works in following steps:

Streaming data pipeline architecture with AWS



Pricing

The pricing of different services in above solutions varies with usage, type of solution, frequency of data and region.

The overview of pricing of different services is given in below table. Here, US East region is being considered. The cost of storing data in data-sources (like S3, DynamoDB) and cost for related monitoring services (CloudWatch logs and events) are not being considered, as it depends on amount of data being generated and processed.

Service	Pricing details
AWS Data pipeline	High Frequency: $1 per month, Low Frequency: $0.6 per month, Inactive pipelines: $1.00 per month
AWS Glue	$0.44 per DPU-Hour, billed per second, with a 1-minute minimum for each ETL job $0.44 per DPU-Hour, billed per second, with a 10-minute minimum for each provisioned development endpoint
AWS Redshift Spectrum	Dense Compute DC2 (dc2.large: $0.25 per Hour, dc2.8xlarge: $4.80 per Hour) Dense Storage DS2 (ds2.xlarge: $0.85 per Hour, ds2.8xlarge: $6.80 per Hour)
AWS Athena	$5.00 per TB of data scanned (by using columnar data format and partitions,

	you can save from 30% to 90% on your per-query costs)
AWS Kinesis data firehose	Data Ingestion: First 500 TB/month: $0.029, Next 1.5 PB/month: $0.025, Next 3 PB/month: $0.02 Data Format Conversion, per GB: $0.018 Per GB processed to VPC: $0.01 Per hour, per AZ for VPC delivery: $0.01
Amazon Kinesis Data Analytics	Kinesis Processing Unit, per Hour: $0.11 per hour Running Application Storage, per GB-month (50 GB of running application storage is assigned per KPU): $0.10 per GB-month Durable Application Backups, per GB-month: $0.023 per GB-month

The pricing of either batch or stream data pipeline solutions can be explained later considering business use case, type of data to be stored, amount of generated and processed data.

Some best practices

- Define clear IAM roles to protect your data and resources among various users
- The volume of data expected.
- The velocity of data, the rate at which it is coming.
- Variety of data that the pipeline will be supporting.
- The validity of data in the pipeline.
- Create separate VPC to keep the data, resources, and pipeline protected.
- Monitor the pipeline using AWS CloudWatch.