Analytics in AWS

Analytics Tensor

Mahesh KC

mahesh.kc@analyticstensor.com

https://analyticstensor.com

Amazon Elastic MapReduce (EMR)

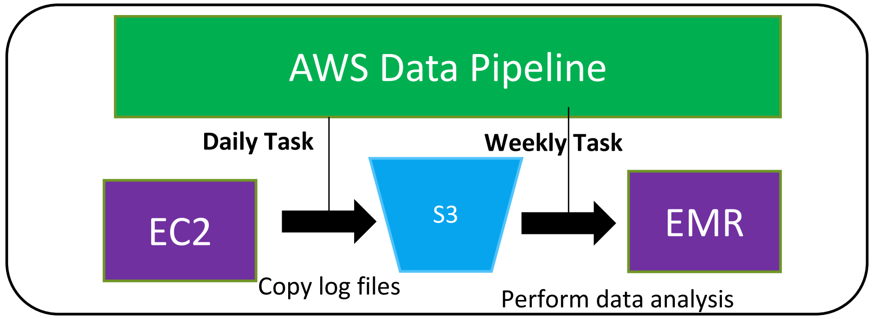
- Amazon Elastic MapReduce is based on Hadoop Framework.
- We can run Spark, Hive, Hbase, Presto.
- It gives us easy access to 1000 of nodes. We can provision single or thousands of instances.
- We can read data from any where either S3, DynamoDB, Redshift etc.
- We can easily integrate with other AWS Services.
- It is ideal services for data-intensive applications such as:
  - Data mining
  - Log Analysis
  - Simulation
  - Genomics

Data Lake and Analytics on AWS: https://aws.amazon.com/big-data/datalakes-and-analytics/

Analytics Tensor

AWS Data Pipeline

- AWS Data Pipeline is a web service that help to automate the movement and transformation of data.
- We can define data-driven workflows such that when one task is completed then other will be triggered automatically.
- It helps to move data between various data sources. For e.g. Moving data from DynamoDB to Redshift, S3 to RDS.
- It can also change the data type while transfer.
- It also supports sources outside of AWS such as on-premises databases (i.e. Oracle, MS-SQL)
- We can use Data Pipeline to execute SQL queries, execute custom applications running in EC2.
- It also support scheduling based on give time frame.
- It has built-in error handing.
- The execution logs of Data Pipeline is Stored in S3.



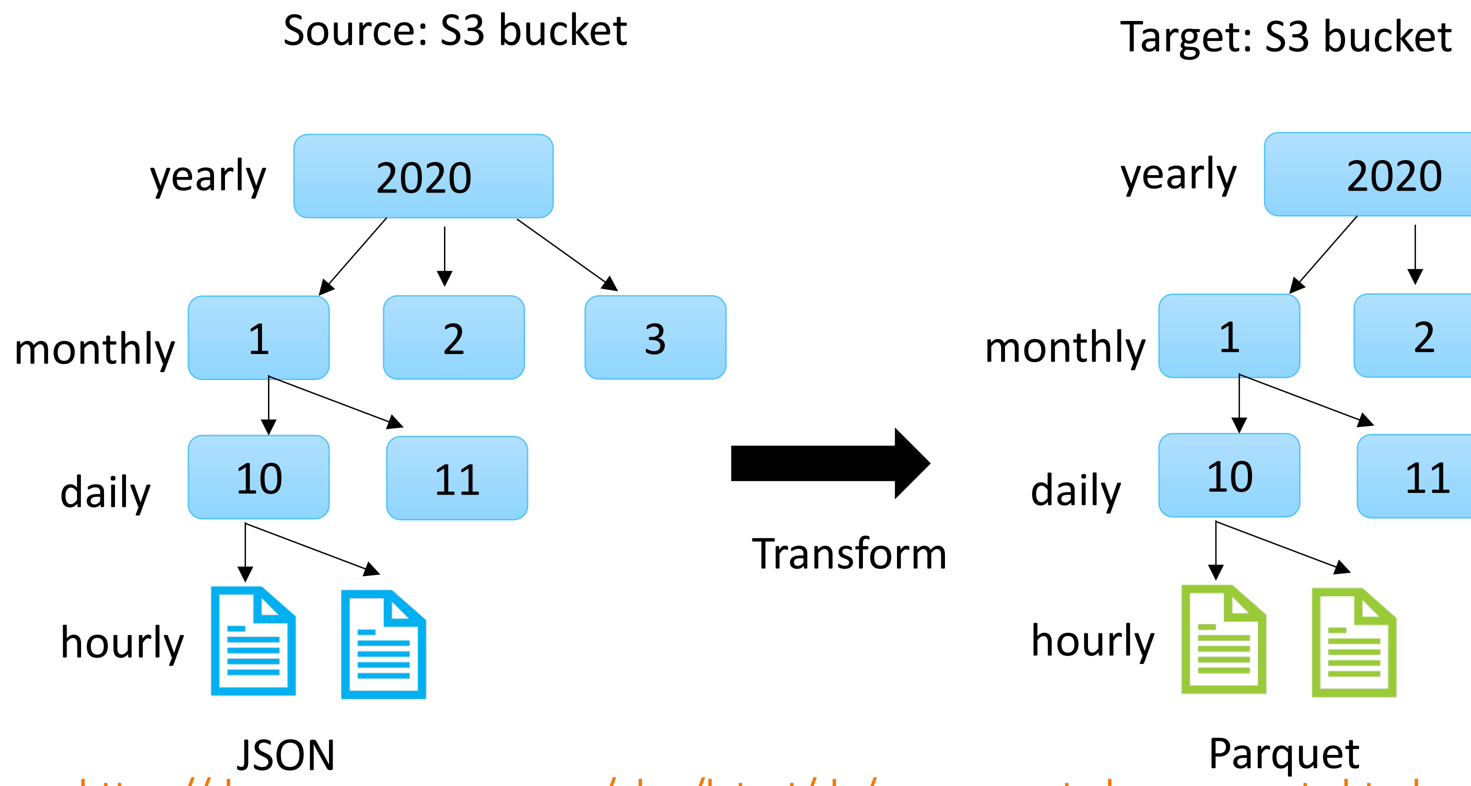
https://docs.aws.amazon.com/datapipeline/latest/DeveloperGuide/what-is-datapipeline.html

AWS Glue

- It is a fully-managed, serverless extract-transform-load (ETL) service.
- Glues has three main components:
  - Data Catalog: It is a centralize metadata store for all the data in AWS. With catalog, it has crawler which reads/open the datasets, extract the metadata, extracts structure, extracts organization and load the metadata into the catalog. The catalog is Hive Metastore compatible. Catalog can also be integrated with other AWS services such as RedShift, Athena, EMR to understand what they are querying.
  - Serverless Job: It is a serverless job execution system. At the core of job execution, it uses Apache Spark to run all the jobs. When we submit the ETL job written in Python and Scala, then it provision all the machine, configure the network, spin-up machine. We have to pay only for the time the job ran. It also has auto-generate ETL code, so that we don’t have to write the code to do transformation.
  - Orchestration: It is used to integrate or stich multiple jobs in linear fashion or DAG. It also has monitoring and alerting system for jobs. It can also be integrated with other external system beyond AWS. It has flexible scheduling.
- Beyond ETL, it also has integration with Amazon SageMaker and Zeppelin Notebooks to perform data science and exploration.
- Glue also provides Python shell. Using Python, we can use for run SQL based ETL like running query in Redshift, build custom connector to pull data from API in Python Shell or run small to medium ML task.

https://aws.amazon.com/glue/

AWS Glue Example



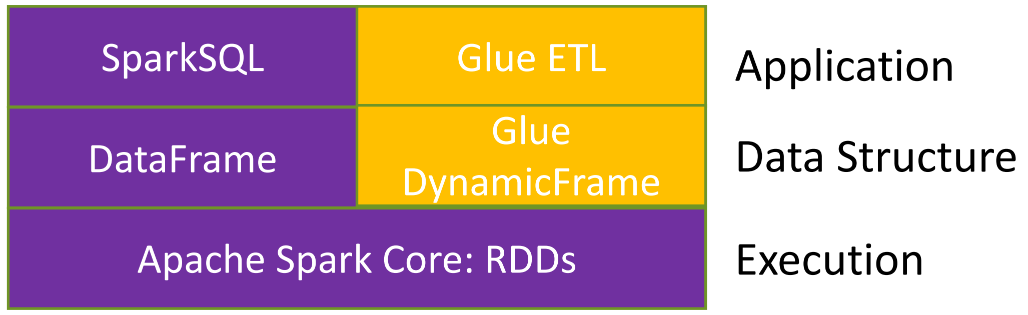
For analytics purpose we want to convert the JSON file to Parquet file and store in other S3 bucket maintaining the file hierarchy with the given partition based on year, month and day.

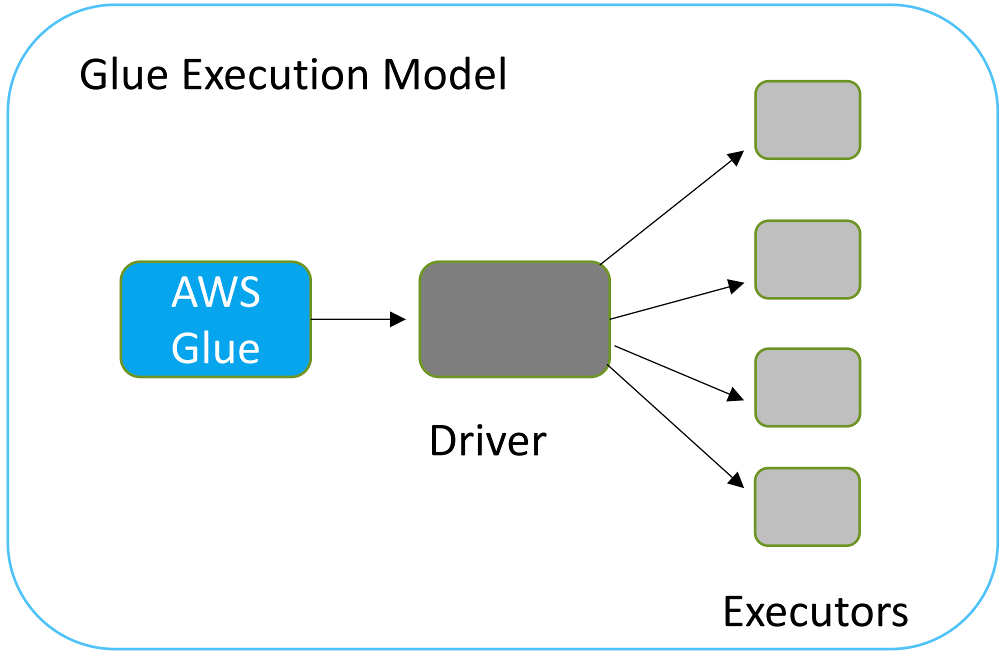
Crawler will automatically picks the top level attributes in the data and other nested attributes. It will keep track of all the file partition, discover and register into data catalog.

https://docs.aws.amazon.com/glue/latest/dg/components-key-concepts.html
https://docs.aws.amazon.com/glue/latest/dg/aws-glue-programming-python-samples-legislators.html

Glue Internal

- Glue is built on top of Apache Spark core.
- DynamicFrame is used to optimize ETL. DataFrame is the basic core data structure for SparkSQL. Transformation on top of DynamicFrames for data cleaning and restructuring.
- DataFrame is like structure table that has schema, each row has same structure and suited for SQL-like analytics.
- DynamicFrames doesn't need schema up-front. It is designed for processing semi-structured data. e.g. JSON, Avro, Apache Logs. We can convert DynamicFrame into DataFrame and vice-versa.
- Glue take Spark script either in Python or Scala and send it to the drivers. The drivers broken into stages. Job are divided into stages. 1 stage X 1 partition is equal to 1 task. Driver schedules tasks on executors. 2 executors per DPU.
- All the ETL metrics are derived from the underlying Spark metrics. Metrics such as memory usage, bytes read/write, cpu load, bytes shuffled, needed executors etc. is send to AWS CloudWatch every 30 sec.





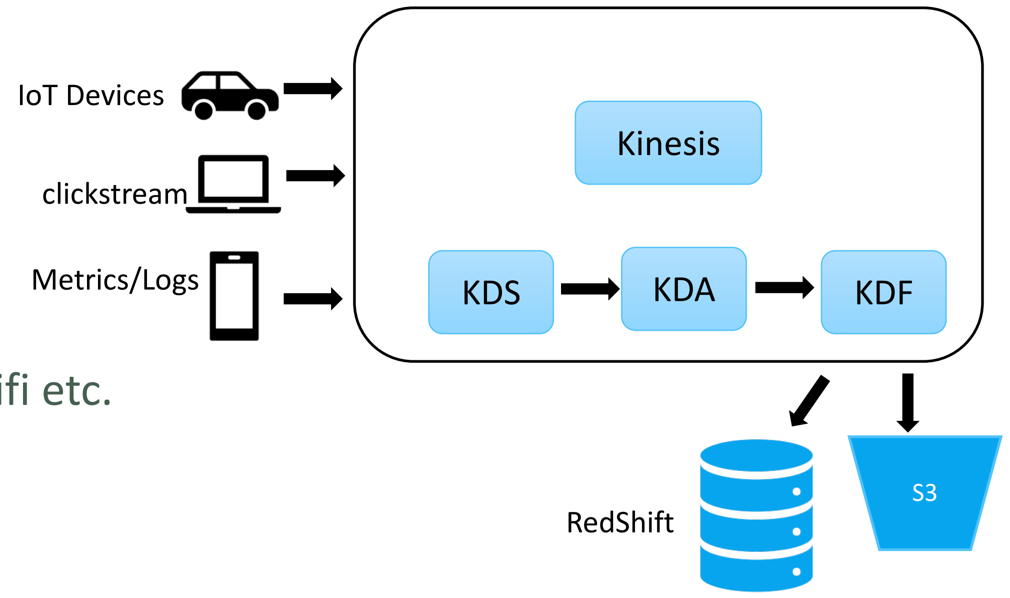
Demo & Use Case of Glue

- Demo
- Use Cases:
  - Retrieve data form input partition
  - Perform Data type validation
  - Perform Flattening
  - Relationalize/Explode
  - Save in Parquet Format.

Check out the image for multiple use-cases: https://aws.amazon.com/glue/

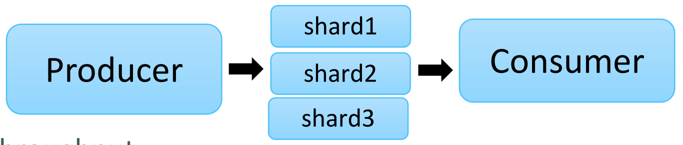
AWS Kinesis

- Kinesis is a fully managed service.
- Kinesis is an alternative to Apache Kafka.
- Great for application logs, metrics, IoT, clickstreams.
- Great for real-time big data.
- Great for streaming processing frameworks such as Spark, Nifi etc.
- Data is automatically replicated synchronously to 3 AZ's.
- Kinesis provides multiple services:
  - Kinesis Data Streams: Low latency streaming ingest at scale.
  - Kinesis Video Streams: Streaming video from connected devices to AWS.
  - Kinesis Data Analytics: Performs real-time analytics on streams using SQL.
  - Kinesis Data Firehouse: Load streams into S3, Redshift, Elastic Search and Splunk.



AWS Kinesis (cont.)

- Kinesis streams are divided into ordered Shards/Partitions.
- Data retention is 24 hours by default and can go up to 7 days.
- We have the ability to reprocess and replay the data until data retention time frame.
- Multiple applications can consume the same stream.
- Real-time processing with scale of throughput.
- Once the data is inserted in Kinesis, it can’t be deleted. So, it is immutable.
- One stream is made of many different shards.
- Billing is per shard provisioned. We can have multiple share as we want.
- Records are ordered per shard.
- Record is produced to the shard. It is made up of data blob.
- Data blob size is up to 1MB and serialized as bytes. The record has its record key.
- Sequence number is added by Kinesis after the data is added to the shard.
- Producer can only produce 1MB/s or 1000 message/s at write per shard. Exceeding the limit will give “provisioned throughput exception”.
- There are two type of consumer:
  1. Classic:
     - 2MB/s at read per shard across all consumers.
     - 5 API calls per second per shard across all consumers.
  2. Consumer Enhanced Fan-Out:
     - 2 MB/s at read per shard, per enhanced consumer.
     - No API calls needed (push model)



AWS Kinesis (cont.)

Terminology: https://docs.aws.amazon.com/streams/latest/dev/key-concepts.html
Demo: https://docs.aws.amazon.com/streams/latest/dev/tutorial-stock-data-kplkcl2.html
Use case: https://www.youtube.com/watch?v=jKPlGznbfZ0. Watch video from 27:00 mins- end

Architecture Video: https://aws.amazon.com/this-is-my-architecture/?sc_channel=EL&sc_campaign=Explainer_2018_vid&sc_medium=YouTube&sc_content=video1964&sc_detail=TMA&sc_country=US&tma.sort-by=item.additionalFields.airDate&tma.sort-order=desc&awsf.category=*all