

Tech Experts


Semana 5

AWS Glue


- Glue Crawler: Descoberta e categorização de dados
 - Pode ser executado de forma manual(ad-hoc) ou agendado
- Data Catalog: Habilita consultas via Athena, EMR e redshift
 - Detecta partições Apache Hive
 - Built in classifiers para tipos populares
- ETL Jobs: podem ser escritos em pyspark ou scala, os códigos podem ser gerados com glue studio.
- Cases de uso:
 - Ler dados do s3, transformar e carregar no redshift ou outra tabela s3
 - Executar Discovery de dados no S3 para posterior transformação com EMR
 - Entre outros.

AWS Glue


Job: s3-glue-redshift Action ▾ Save Run job Generate diagram ?




Database Name s3-data
Table Name sales_records_csv




Transform Name ApplyMapping



Transform Name ResolveChoice



Transform Name DropNullFields



Connection Name glue-redshift-connect
Database Name

```
1 import sys
2 from awsglue.transforms import *
3 from awsglue.utils import getResolvedOptions
4 from pyspark.context import SparkContext
5 from awsglue.context import GlueContext
6 from awsglue.job import Job
7
8 ## @params: [TempDir, JOB_NAME]
9 args = getResolvedOptions(sys.argv, ['TempDir', 'JOB_NAME'])
10
11 sc = SparkContext()
12 glueContext = GlueContext(sc)
13 spark = glueContext.spark_session
14 job = Job(glueContext)
15 job.init(args['JOB_NAME'], args)
16
17 ## @type: DataSource
18 ## @args: [database = "s3-data", table_name = "sales_records_csv"]
19 ## @return: DataSource
20 datasource0 = glueContext.create_dynamic_dataframe(f"SELECT * FROM {database}.{table_name}")
21
22 ## @type: ApplyMapping
23 ## @args: [mapping = [{"region": "string"}]]
24 ## @return: ApplyMapping
25 applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [{"region": "string"}])
26
27 ## @type: ResolveChoice
28 ## @args: [choice = "make_cols", transformation_ctx = "applymapping1"]
29 ## @return: ResolveChoice
30 resolvechoice2 = ResolveChoice.apply(frame = applymapping1, choice = "make_cols", transformation_ctx = "applymapping1")
31
32 ## @type: DropNullFields
33 ## @args: [transformation_ctx = "resolvechoice2"]
34 ## @return: DropNullFields
35 dropnullfields3 = DropNullFields.apply(frame = resolvechoice2, transformation_ctx = "resolvechoice2")
```

Logs Schema

Athena

- Use schema do glue para consultar dados no S3
- Suporte para: XML, JSON, CSV, AVRO, PARQUET, ORC entre outros
- CloudTrail, ELB logs e vpc flow logs podem ser armazenados no S3 para serem consultados via Athena
- Schema on read
- Output da query pode ser utilizado por outros serviços para finalidades analíticas como visualização

Kinesis data streams

- Ingestão near real time em larga escala
- Múltiplos produtores e consumidores
- Retenção padrão: 24h – Máximo: 365 dias
- Produtores podem ser: EC2 rodando um código, lambda, iot devices, aplicações mobile, web entre outros
- Consumidores é semelhante, podem ser códigos rodando em EC2, lambda lendo os dados. AWS prove triggers para invocação de lambda assim que um dado chega no kinesis stream.

Kinesis data streams

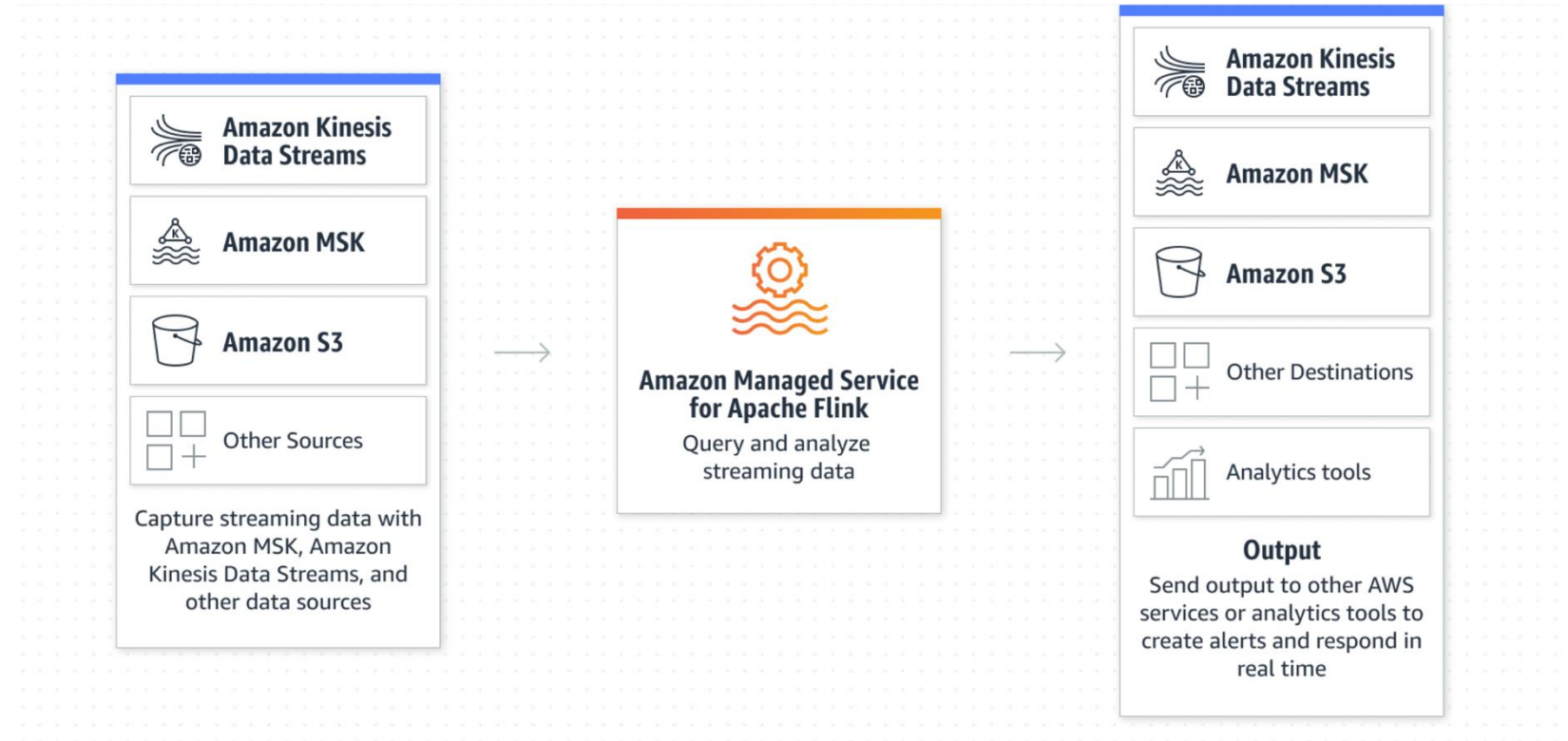
- Escalavel através de shards
 - Partições logicas que particiona dados pela chave de partição (todo evento kinesis deve possuir uma chave de partição)
 - 1MB/sec writing e 2MB/sec Reading / shard
 - 1000 put records/sec / shard
- Pode ser criptografado utilizando SSE através do KMS
- Não deve ser confundido com SQS:
 - 1 grupo produtor e 1 grupo consumidor
 - Use case com múltiplos produtores/consumidores: kinesis
 - Para desacoplamento e comunicação assíncrona: SQS
 - SQS não possui conceito de persistência. O evento é excluído após lido.
 - Large scale ingestion: Kinesis.

Kinesis firehose

- Recebe dados em streaming e armazena para futuras análises
- Pode ser plugado como um consumidor do kinesis streams
- Pode armazenar dados no S3, redshift, Elasticsearch servisse, splunk, entre outros.
- Dados podem ser transformados por uma lambda antes de serem armazenados
- Dados podem ser comprimidos usando gzip, zip ou snappy.

Amazon Managed Service for Apache Flink

- Analytics Real Time
- Flink Studio
- Apache Beam
- SQL



AWS datasync

- Sincronia de seus dados no ambiente on premisses com ambiente AWS
- Pode ser sincronizado com:
 - S3
 - EFS
 - FSx for Windows
- Pode ser feito real time ou agendado, com limite de uso de rede.

Snowball

- Transferencia física dos dados para AWS
- **Snowball:**
 - Encrypted using KMS.
 - 50 TB ou 80 TB.
 - Economico para 10TB a 10
- **Snowball Edge:** This is like Snowball, but it comes with both storage and computes capability.
- **Snowmobile:** This is a portable data center within a shipping container on a truck. This allows you to move exabytes of data from on-premises to AWS.

AWS EMR

- Cluster hadoop/spark gerenciado pela AWS
- Processamento distribuído
- Pode ser cluster de longa duração ou ad hoc
- Pode ser habilitado auto scaling
- Utiliza EC2 em seu background, em uma única zona de disponibilidade
 - (Glue utiliza EMR no seu background, porém sem necessidade de entender questões operacionais do EMR)
- Use case: parecido com glue, mas o principal é quando você tem seus dados de input no S3 e o output também, visto que ele possui menos facilidades de conectores, como é o caso do glue.
- Possui EMRFS, extensão do HDFS para buscar dados diretamente no S3.

AWS Batch

- Processamento batch baseado em eventos
- Longo tempo de duração
- Pode executar um script ou um executável (ex: jar)
- Executa no ECS ou EC2
- Fila de Jobs
- *If you get a question in exams on an event-style workload that requires flexible compute, a higher disk space, no time limit (more than 15 minutes), or an effective resource limit, then choose AWS Batch.*

Q1

A machine learning (ML) specialist has more than 1 TB of objects that are stored in an Amazon S3 bucket. The objects are named with a subpath under a common S3 path. The ML specialist wants to group the objects for batch loading into an Amazon EMR cluster for processing.

Which solution will meet these requirements with the LEAST amount of effort?

- A. Use recursive partitioning in AWS Glue
- B. Deploy a Apache Flink as a preprocessor of EMR Cluster
- C .Use managed data identifier within Amazon Macie on Amazon S3
- D .Active AWS Lambda on S3 object writes to send the objects do Amazon Data Streams for analysis by Amazon Managed Service for Apache Flink (previsously knwon as Amazon Kinesis Data Analytics)

Q1

A machine learning (ML) specialist has more than 1 TB of objects that are stored in an Amazon S3 bucket. The objects are named with a subpath under a common S3 path. The ML specialist wants to group the objects for batch loading into an Amazon EMR cluster for processing.

Which solution will meet these requirements with the LEAST amount of effort?

- A. Use recursive partitioning in AWS Glue
- B. Deploy a Apache Flink as a preprocessor of EMR Cluster
- C. Use managed data identifier within Amazon Macie on Amazon S3
- D. Active AWS Lambda on S3 object writes to send the objects to Amazon Data Streams for analysis by Amazon Managed Service for Apache Flink (previously known as Amazon Kinesis Data Analytics)

Q2

- A company is setting up a system to manage all of the datasets that it stores in Amazon S3. The company wants to automate running transformation jobs on the data and maintaining a catalog of the metadata about the datasets.
- Which solution will meet these requirements with the LEAST operational overhead?
- A. Create an EMR provisioned cluster with Apache Hive installed. Then, create a Hive metastore and a script to run transformation Jobs on a schedule
- B. Create an AWS Glue crawler to populate an AWS Glue Data Catalog. Then, create a Glue ETL job and set up a schedule for data transformation Jobs.
- C. Create an EMR provisioned cluster with Apache Spark installed. Then, create a Hive metastore and a script to run transformation Jobs on a schedule
- D. Create a SageMaker Jupyter notebook instance that transforms the data. Then, create a Hive metastore and a script to run transformation Jobs on a schedule

Q2

- A company is setting up a system to manage all of the datasets that it stores in Amazon S3. The company wants to automate running transformation jobs on the data and maintaining a catalog of the metadata about the datasets.
- Which solution will meet these requirements with the LEAST operational overhead?
- A. Create an EMR provisioned cluster with Apache Hive installed. Then, create a Hive metastore and a script to run transformation Jobs on a schedule
- B. Create an AWS Glue crawler to populate an AWS Glue Data Catalog. Then, create a Glue ETL job and set up a schedule for data transformation Jobs.
- C. Create an EMR provisioned cluster with Apache Spark installed. Then, create a Hive metastore and a script to run transformation Jobs on a schedule
- D. Create a SageMaker Jupyter notebook instance that transforms the data. Then, create a Hive metastore and a script to run transformation Jobs on a schedule

Q3

- A machine learning (ML) specialist is setting up an ML pipeline. The objective is to enable the ETL part of the pipeline to activate ML training jobs in Amazon SageMaker. Specifically, the ML specialist intends to use batch jobs for ETL. The solution must integrate with SageMaker without the use of additional services.
- Which solution will meet these requirements?
- A. Use AWS Elastic Beanstalk to provision AWS batch
- B. Use Apache Flink on EMR for the ETL
- C . Use Glue for the ETL
- D . Use Amazon Kinesis Data Streams for the ETL

Q3

- A machine learning (ML) specialist is setting up an ML pipeline. The objective is to enable the ETL part of the pipeline to activate ML training jobs in Amazon SageMaker. Specifically, the ML specialist intends to use batch jobs for ETL. The solution must integrate with SageMaker without the use of additional services.
- Which solution will meet these requirements?
- A. Use **AWS Elastic Beanstalk** to provision AWS batch
- B. Use **Apache Flink** on EMR for the ETL
- C . **Use Glue for the ETL**
- D . Use **Amazon Kinesis Data Streams** for the ETL

Q4

- A Mobile Network Operator is building an analytics platform to analyze and optimize a company's operations using Amazon Athena and Amazon S3. The source systems send data in .CSV format in real time. The Data Engineering team wants to transform the data to the Apache Parquet format before storing it on Amazon S3. Which solution takes the LEAST effort to implement?
- A. Ingest .CSV data using Apache Kafka Streams on Amazon EC2 instances and use Kafka Connect S3 to serialize data as Parquet
- B. Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Glue to convert data into Parquet.
- C. Ingest .CSV data using Apache Spark Structured Streaming in an Amazon EMR cluster and use Apache Spark to convert data into Parquet.
- D. Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Kinesis Data Firehose to convert data into Parquet.

Q4

- A Mobile Network Operator is building an analytics platform to analyze and optimize a company's operations using Amazon Athena and Amazon S3. The source systems send data in .CSV format in real time. The Data Engineering team wants to transform the data to the Apache Parquet format before storing it on Amazon S3. Which solution takes the LEAST effort to implement?
- A. Ingest .CSV data using Apache Kafka Streams on Amazon EC2 instances and use Kafka Connect S3 to serialize data as Parquet
- B. Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Glue to convert data into Parquet.
- C. Ingest .CSV data using Apache Spark Structured Streaming in an Amazon EMR cluster and use Apache Spark to convert data into Parquet.
- D. Ingest .CSV data from Amazon Kinesis Data Streams and use Amazon Kinesis Data Firehose to convert data into Parquet.

Q5

- A manufacturing company has structured and unstructured data stored in an Amazon S3 bucket. A Machine Learning Specialist wants to use SQL to run queries on this data.
Which solution requires the LEAST effort to be able to query this data?
- A. Use AWS Data Pipeline to transform the data and Amazon RDS to run queries.
- B. Use AWS Glue to catalogue the data and Amazon Athena to run queries.
- C. Use AWS Batch to run ETL on the data and Amazon Aurora to run the queries.
- D. Use AWS Lambda to transform the data and Amazon Kinesis Data Analytics to run queries.

Q5

- A manufacturing company has structured and unstructured data stored in an Amazon S3 bucket. A Machine Learning Specialist wants to use SQL to run queries on this data.
Which solution requires the LEAST effort to be able to query this data?
- A. Use AWS Data Pipeline to transform the data and Amazon RDS to run queries.
- **B. Use AWS Glue to catalogue the data and Amazon Athena to run queries.**
- C. Use AWS Batch to run ETL on the data and Amazon Aurora to run the queries.
- D. Use AWS Lambda to transform the data and Amazon Kinesis Data Analytics to run queries.