**Architecture & Flow**

**Flow as per my assumption for batch processing**

Uses NiFi on EC2. Nifi can fetch files from ftp and sftp location as well data from different databases.

It will validate the files like type of file, size etc. and Store into AWS S3 Bucket.

* Once new file in S3 bucket a lambda connected with s3 bucket will execute and run AWS Glue crawler. And subsequently ETL job in AWS Glue.
* An script written in PYSpark or scala will perform ETL (Sum of readings) and store in Aurora db(Postgre)
* Aurora db can be visualize with different tools like tableu.

**AWS Services Used**

* IAM
* VPC
* S3 Bucket
* EC2
* AWS glue Crawler, connection etc
* AWS Glue ETL Job in Pyhton
* Aurora DB(Postgre)
* Cloudwatch logs
* Nifi on EC2

**Why Used these tools**

Nifi:-

1. A visual tool to fetch files and data, perform validation and can do some ETL also.
2. Can fetch thousands of files and store it in S3 bucket
3. Can easily handle failover scenario

S3 Bucket :-

1. To store files and data in data lake in different format.
2. Can be easily integrate with other AWS Services

AWS Glue :-

1. For batch job on spark with multiple nodes.
2. Can be integrated with cloudwatch, lambda etc
3. ETL Jobs can be scheduled or on demand or can be triggered with event based.
4. Can be MapReduce with help of spark dataframe.
5. Provide flexibility to write job code in Scala or python
6. Can use sagemaker or zeppelin notebook.
7. Can create job with visual tool
8. Monitoring of jobs

Aurora Postgre:- Benefits of RDBMS with 3x higher throughput than RDS.

AWS Services Could not able to use due to time and other limitation

* AWS Lambda
* Elk Stack for logging purpose
* Wanted to use Datapipeline or EMR but not allowed on free tier
* Logs and metadata related to files from Nifi in either postgre or in any nosql db
* Visualization tool like quicksight
* Validation, Exception handling etc could able to implement due to time limitation
* Could not able to create cloud formation template