use-case-recommender-system :

ASSUMPTIONS :

- the data scientist has built the first version of the model on a jupyter notebook using clickstream data processed as part of 01-uc-bigdataprocessing
- the model is required to be deployed in a production setup for continuous training and scoring daily
- the model needs be expanded to multiple geo regions
- the model outputs should be available in both offline and online mode (real-time prediction service)

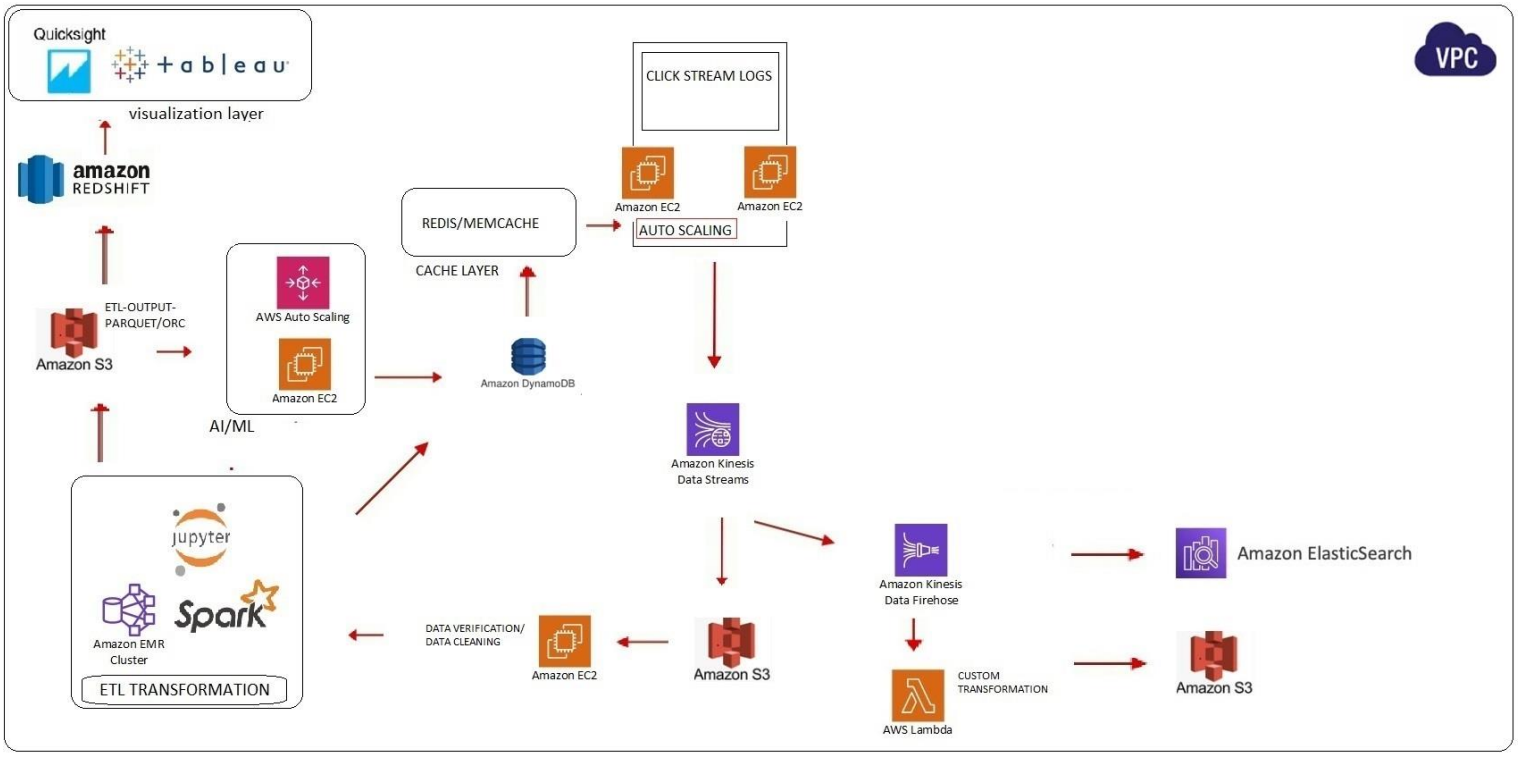
CONSIDERATIONS :

- to be implemented on AWS stack
- we are not looking at the modelling approach but more interested in deploying and running the recommendation engine in a scalable manner
- the model prediction service has to scale for millions of users.

OUTPUT :

- the output should show the deployment and the Implementation view as design document with all core architecture considerations and components.

DESIGN DIAGRAM :



Solution Specification:

1. Incoming data - Website - clickstream incoming logs.
2. Message Queue - AWS Kinesis Streams as alternative to Apache Kafka. Managed by AWS, Kinesis serverless scalable real time data streaming service.
3. Message relayed to Kinesis Data Firehouse –
    a. Incoming stream data, apply lambda custom transformations – store data in S3, analyze Realtime logs elastic-search.
    b. Store raw data over S3, EC2 instance – Verify file atomicity, check to convert file-formats or apply necessary permissions, Transfer over EMR system.
4. EMR System – Transfer over to HDFS or EMRFS, Spark ETL Jobs -
    a. EMR configure to auto-scale.
    b. store output towards S3 as columnar file-formats like ORC/Parquet.
    c. parsed Key data points - transfer to DynamoDB.
5. EC2 ML - Memory optimized instances - R4, R5.
    a. ML Job – models applied over ETL output data.
    b. EC2 instances - If its required, can be put in auto-scaling group.
    c. Store recommendation key points over DynamoDB.
6. Additional cache layer can be added for strong performance.
7. Redshift – to be used as data warehouse.