**Youtube Data Analysis**

dataset from kaggle -

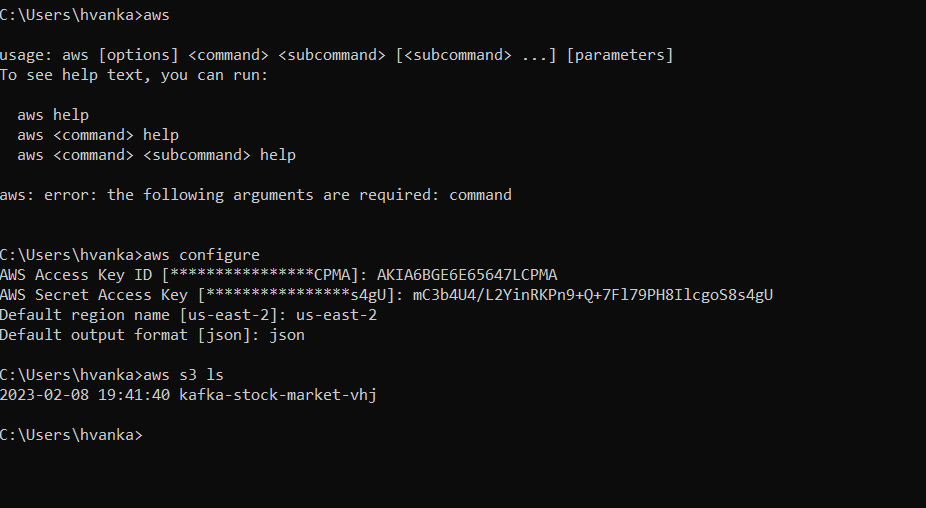
<https://www.kaggle.com/datasets/datasnaek/youtube-new>

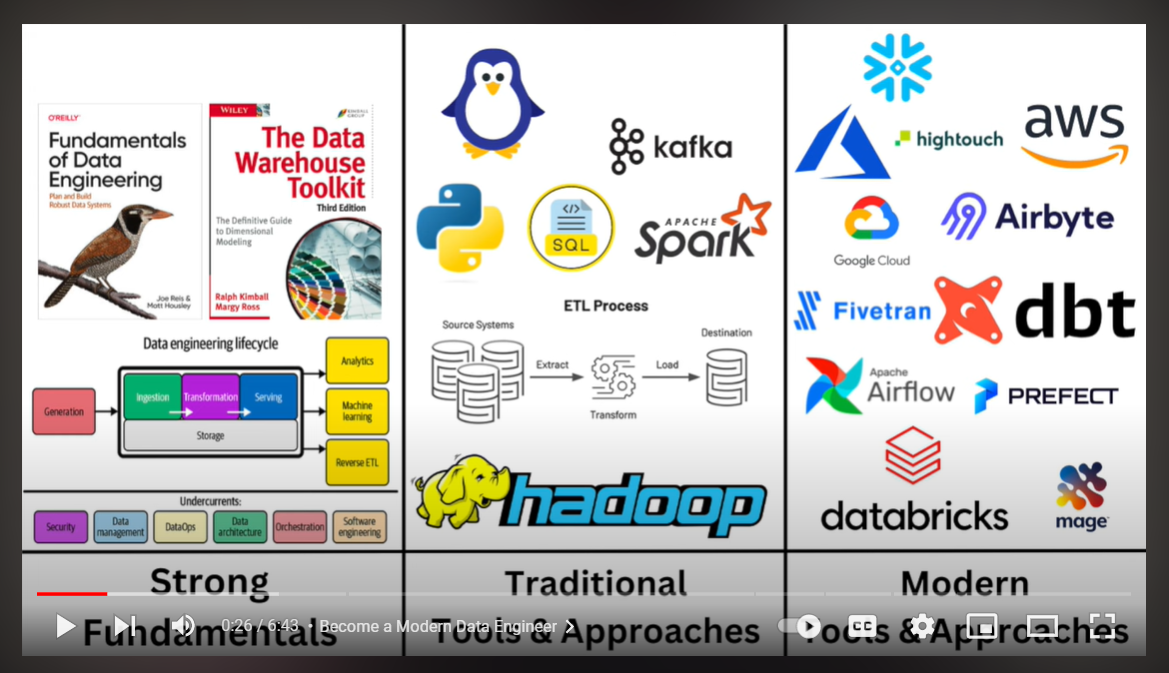
Graphical user interface, diagram, application

Description automatically generated

* data ingestion – from data sources
* data lake - design and build new data architecture
* aws cloud - provider
* etl design - extract, tranform, load data efficiently
* scalability - data architecture scale efficiently
* reporting - build businesss inteligence tier, dashboards
* build data lake from scraatch in s3
* joining structured and semi-structured data
* lake house architecture design
* data lake vs data warehouse
* etl in aws glue spark jobs
* amazon sagemaker jupter notebooks
* amazon sns for alerting
* sql using amazon athena and spark sql
* bi dashboards
* youtube
* what is trending - factors inlcudes user interactions
* no of shares, views, likes, comments
* not most viewes videos
* setup - aws account
* IAM user and groups
* s3 - buckets
* transfer data from windows to s3 in commandline using hive pattern commands
* lake house architecure
* all types of data stored - structured or unstructred data or etc.
* aws glue catalog
* glue -> crawler -> data catalog -> job(schedule or event)-> transform -> load to get the target data
* de\_youtube\_cleansed\_table
* de\_youtube\_cleansed
* Streamline Youtube analysis on structured and semi-structured youtube videos data from data ingestion to building etl pipeline to visualize data in QuickSight.

Dataset -> Data Ingestion (S3) -> Lambda -> Data Catalog -> Job-> Athena -> Quicksight.





# To copy all JSON Reference data to same location:

aws s3 cp . s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_data/ --recursive --exclude "\*" --include "\*.json"

# To copy all data files to its own location, following Hive-style patterns:

aws s3 cp CAvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=ca/

aws s3 cp DEvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=de/

aws s3 cp FRvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=fr/

aws s3 cp GBvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=gb/

aws s3 cp INvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=in/

aws s3 cp JPvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=jp/

aws s3 cp KRvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=kr/

aws s3 cp MXvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=mx/

aws s3 cp RUvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=ru/

aws s3 cp USvideos.csv s3://de-youtube-analysis-using-raw-data/youtube/raw\_statistics\_region\_data/region=us/

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, application

Description automatically generated

Diagram

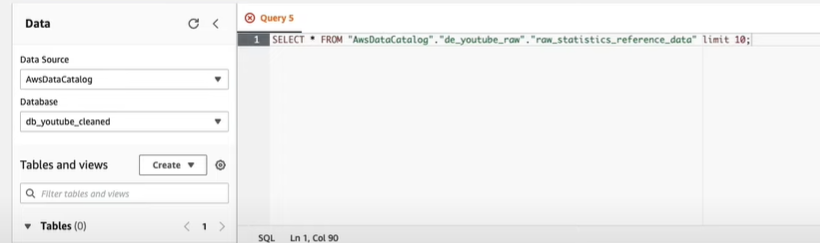
Description automatically generated

Diagram

Description automatically generated

Graphical user interface, application

Description automatically generated



Json data failed to get queried.

Json SerDe libraries -

Serialized and deserialized

Data cleaning semi- structured pipeline

Json

-> json s2 bukcet

-> aws lambda, json to apache parquet

-> s3 cleansed bucket / aws data wrangler

-> amazon athena / glue catalog

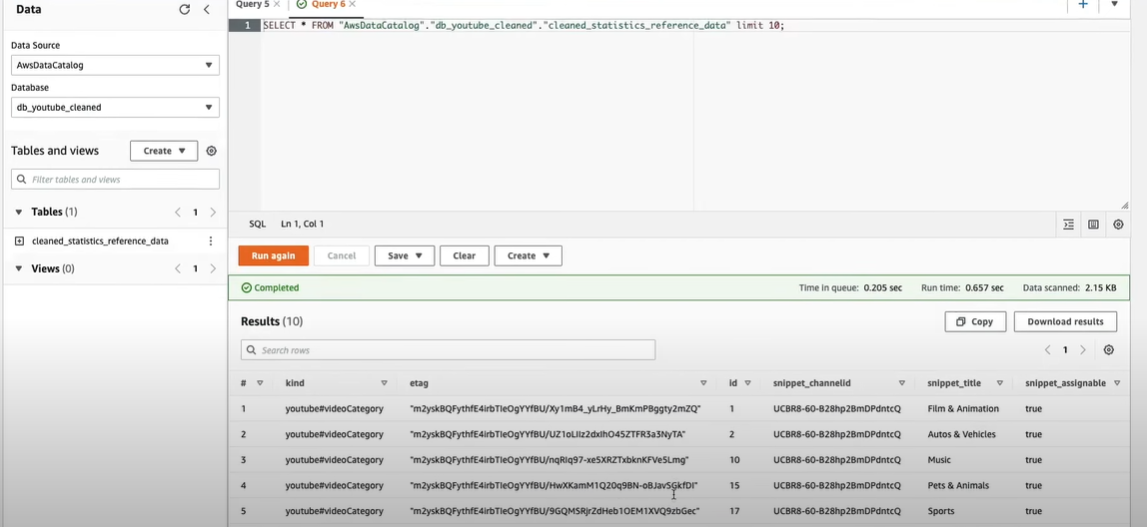
* Created lambda python file and assigned outputs to get created such as glue catalog, database, s3 parquet file.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated



* Changing the data type string to bigint -> preprocessing
* After that recreating the parquet/ database catalog by deleting the previous one.
* Instead of querying data using join operations or any operations and retrieving the data, we creating the **job and event**
* While create add target format as parquet and it create spark job gives default code and insert data with our specific requirements

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Building etl pipeline

Diagram

Description automatically generated

Graphical user interface, chart

Description automatically generated with medium confidence

* Now building the reporting version
* Now finally on to quicksight to visualize the reports - on choosing database data source.

Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, chart, application, pie chart

Description automatically generated