**ActivePipe PoC**: **Building Event Driven Data Pipeline and Final Store**

The goal of this poc to build a pipeline and copy the source data into a final store i.e. a data lake. This is to mimic the data strategy shown below that was recommended to ActivePipe.

Graphical user interface, application

Description automatically generated

Fig: Recommended Data strategy

**Why this strategy?** AP’s data sources are in different places including MySQL & ElasticSearch. According to proposed data strategy idea is to bring data from various sources into a final data store efficiently using Kafka as message the bus. This will help dashboarding, analytics and ML easier and easier to maintain.

**What does this poc do?**

* + Integrating data sources with Apache Kafka
  + Copying data to a final store i.e. a data Lake.
  + Visualize the data in the final store data
  + Connect the final store to Tableau and visualize the data
  + An example for event driven, streaming and Lakehouse Architecture

**The Design**

This poc is built on event driven architecture and data streaming. The product is an example of Lakehouse Architecture which is gaining more popularity.

**Diagram

Description automatically generated**

**Fig: The Design**

This application reads data from S3 bucket s3://activepipe-poc and copies the content to kafka topic "ap\_property\_listing" in confluent cloud. This topic needs to be created with X partitions though not mandatory.   
This is the first part of the pipeline where the final place will be Databricks's DeltaLake

In the next part, the data that is in Kafka will be copied to Delta Lake. That will be done on Databrick's environment.

**Why Delta Lake?**

* Open source, no vendor lock-in, lake house architecture
* Your choice of storage, separates storage and compute
* Data is stored in parquet format which open source, supports structure and un-structured data
* Provides ACID transactions, time travel
* Supports both streaming and batch data ingestion

**A picture containing diagram

Description automatically generated**

**Fig: Delta lake**

**Components:**

Data flows from AWS S3 to Kafka and then to Delta Lake. This module has following components**.**

**AWS S3:** ActivePipe’s data files are manually uploaded to the buckets3://activepipe-poc. When a new file comes in, an event is created which in turn triggers a lambda (No code for S3 component)

Graphical user interface, text, application, email

Description automatically generated

Fig: ActivePipe data files

Graphical user interface, application

Description automatically generated

Fig: Event notification configuration.

**AWS Lambda:** It downloads the newly come in file, extracts the house listing data and copies them over to Kafka topic “ap\_property\_listing” on Confluent’s cloud env in avro format (Code will be shared, written in Java and Springboot).

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Databrick’s Delta Lake: It** is the final store aka data lake. Data from the kafka topic will be streamed into the Delta Lake’s bronze table. Cleansed data from the bronze table will be stored in Sales and Rent based silver tables. (Code will be shared, written in Python with Apache spark APIs)

There are three workspaces.

1. Ingest Property Listing: Ingest data from Kafka and store them in bronze table
2. Silver Tables: Read data from the bronze table, cleanse and copy them to silver tables.
3. Listing Aggregates: SQL queries to create aggregates and visualise them in charts and tables in Databricks env

Chart, bar chart

Description automatically generated

Fig: Property sales by year

Chart, bar chart

Description automatically generated

Fig: Property listing by year (Rent)

**Tableau**: After installing the delta lake driver, Delta Lake is connected to Tableau and data is retrieved through queries. There is no duplication of data. Data is visualised through maps and charts.

Connection to delta lake and table data:

Graphical user interface, application

Description automatically generated

Map

Description automatically generated

**Fig:** House sales data on the map

Chart, bar chart

Description automatically generated

Fig. Average sale price and sales by year