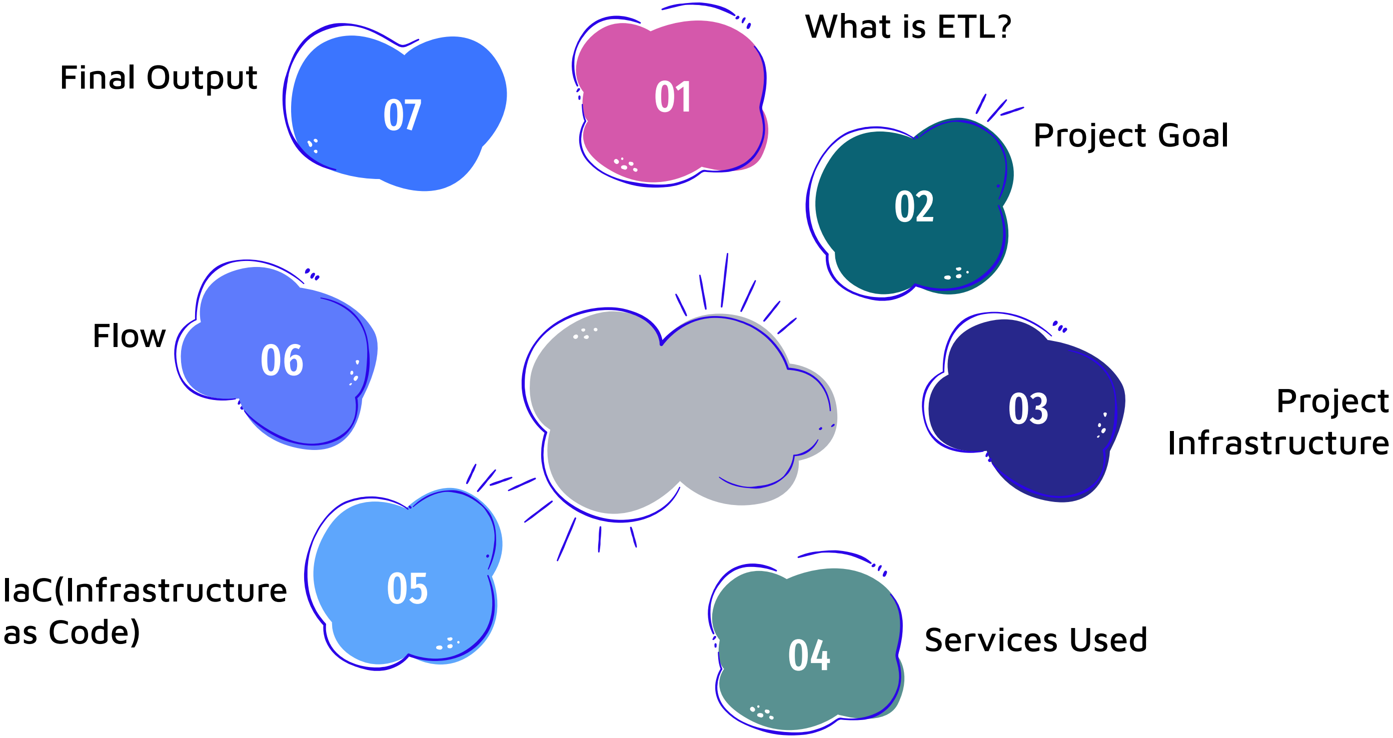
HTW Winter Semester 2023/24 Cloud Computing Presentation

Developing ETL Pipelines Using API's

Lead By : Prof. Nico Schönnagel
Siddharth Gupta , Akhilesh Parundekar, Tapan Solanki

Table of Contents



01 What is ETL?

02 Project Goal

03 Project Infrastructure

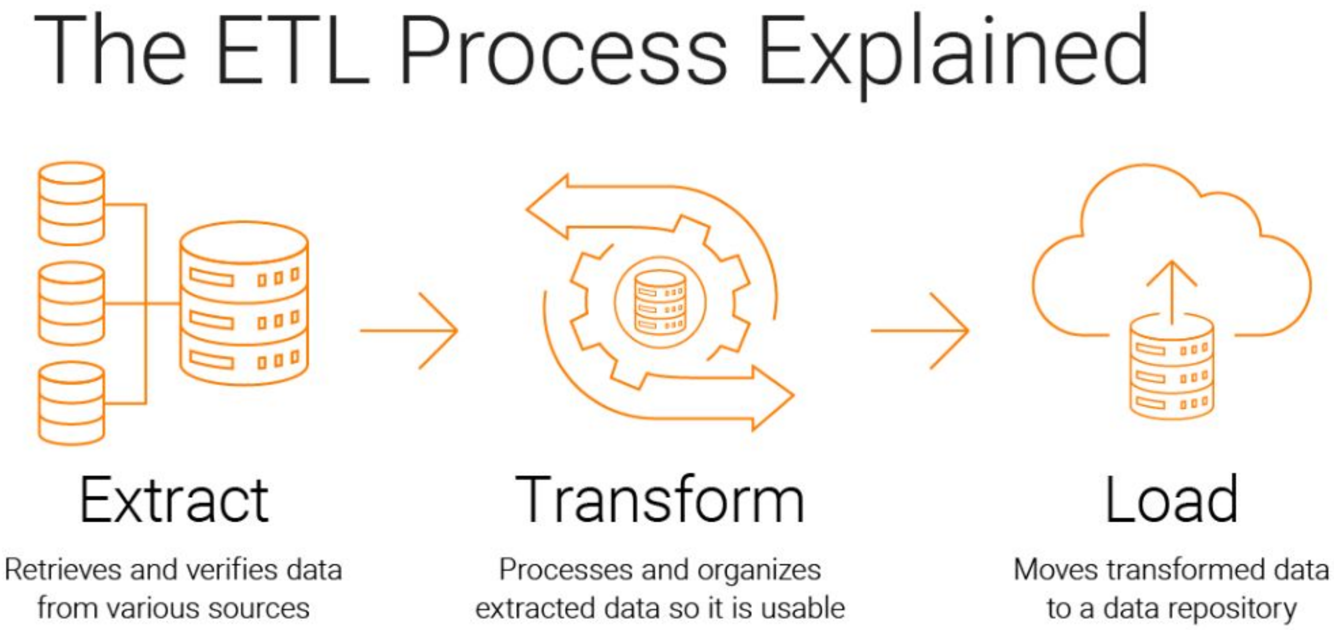
04 Services Used

05 IaC(Infrastructure as Code)

06 Flow

07 Final Output

What is ETL ?



The ETL Process Explained

Extract

Retrieves and verifies data from various sources

Transform

Processes and organizes extracted data so it is usable

Load

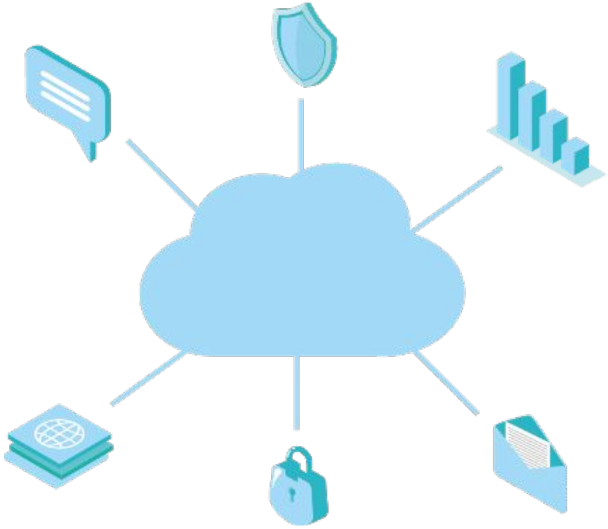
Moves transformed data to a data repository

Project Goal

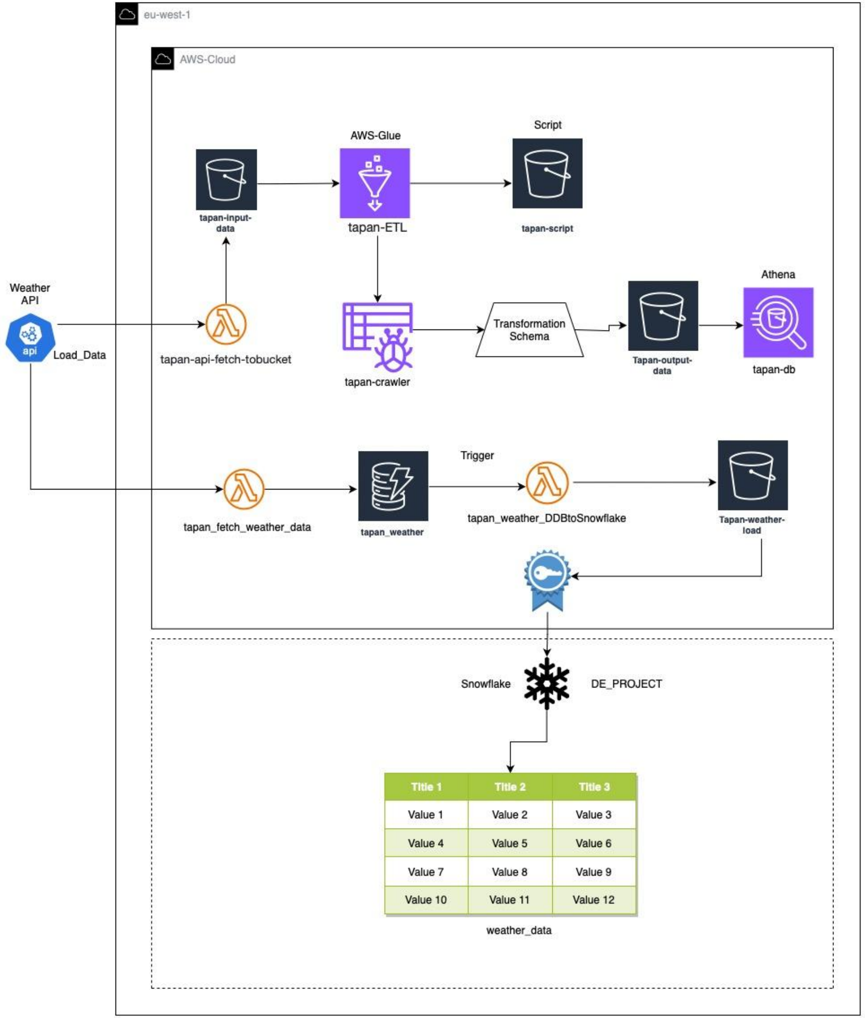
The overall goal of an ETL project using AWS is to efficiently and reliably integrate, transform, and load data into a target system for analysis and decision-making

This can lead to various benefits, such as:

- Improved business insights: Gain deeper understanding of customers, operations, and trends.
- Enhanced decision-making: Make data-driven decisions based on accurate and timely information.
- Increased efficiency: Automate data pipelines and reduce manual effort.
- Reduced costs: Optimize data storage and processing for cost-effectiveness.



Project Infrastructure



Title 1	Title 2	Title 3
Value 1	Value 2	Value 3
Value 4	Value 5	Value 6
Value 7	Value 8	Value 9
Value 10	Value 11	Value 12

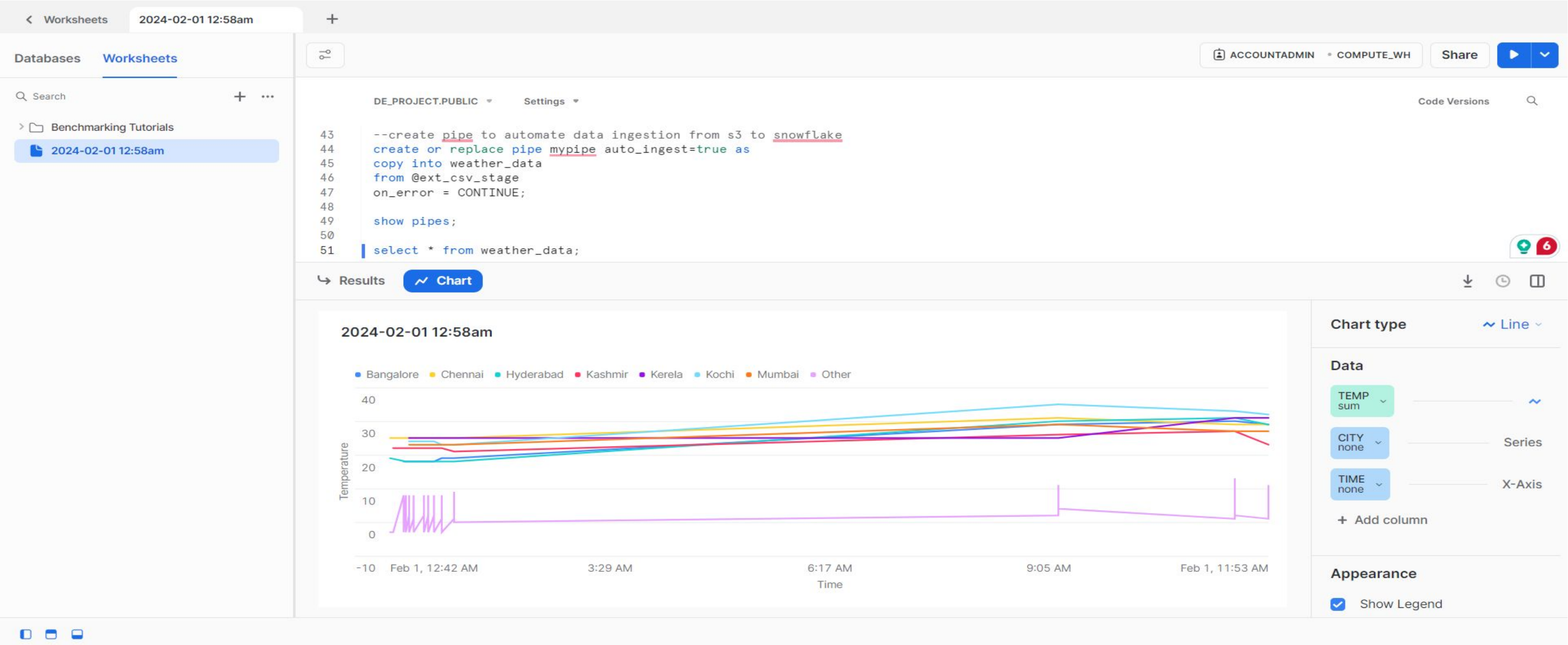
Services Used :

1. Weather API
2. S3 Bucket
3. Lambda Functions
4. AWS Glue
5. Amazon Athena
6. Amazon Crawler
7. Dynamodb
8. Snowflake
9. Cloud Formation for IAC

Amazon Services Overview:

1. **Weather API**: Get real time weather and geo data in .json format.
2. **Lambda Function 1**(*tapan_fetch_weather_data*): Get data from the API (**Extract**)and insert it into DynamoDB
3. **Amazon DynamoDB**(**tapan_weather**): To store the data we get from API. Good for handling large amount of data
4. **Lambda Function 2***(tapan_weather_DDBtoSnowflake*): **Transform** the data stored in DynamoDB in .csv format and put it in a S3 Bucket.
5. **Amazon S3 Bucket**(*tapan-weather-load*): Store the transformed data and pass it on to Snowflake DB
6. **Snowflake**: A DB service which helps us store, retrieve and view our data. Useful for Data Analysis. Here we **Load** our data
7. **AWS Glue:**(*tapan-ETL*) A fully managed extract, transform, and load (ETL) service that makes it easy for users to prepare and load their data for analysis.
8. **AWS Crawler:**(*tapan-crawler*) An AWS Glue component that automatically discovers, classifies, and extracts metadata from various data sources, facilitating efficient data cataloging and ETL processes.
9. **AWS Athena:**(*tapan-db*) A serverless, interactive query service that enables users to analyze data in Amazon S3 using standard SQL, without the need for infrastructure management.

Final Data : Snowflake



```
--create pipe to automate data ingestion from s3 to snowflake
create or replace pipe mypipe auto_ingest=true as
copy into weather_data
from @ext_csv_stage
on_error = CONTINUE;

show pipes;

select * from weather_data;
```

Chart type: Line

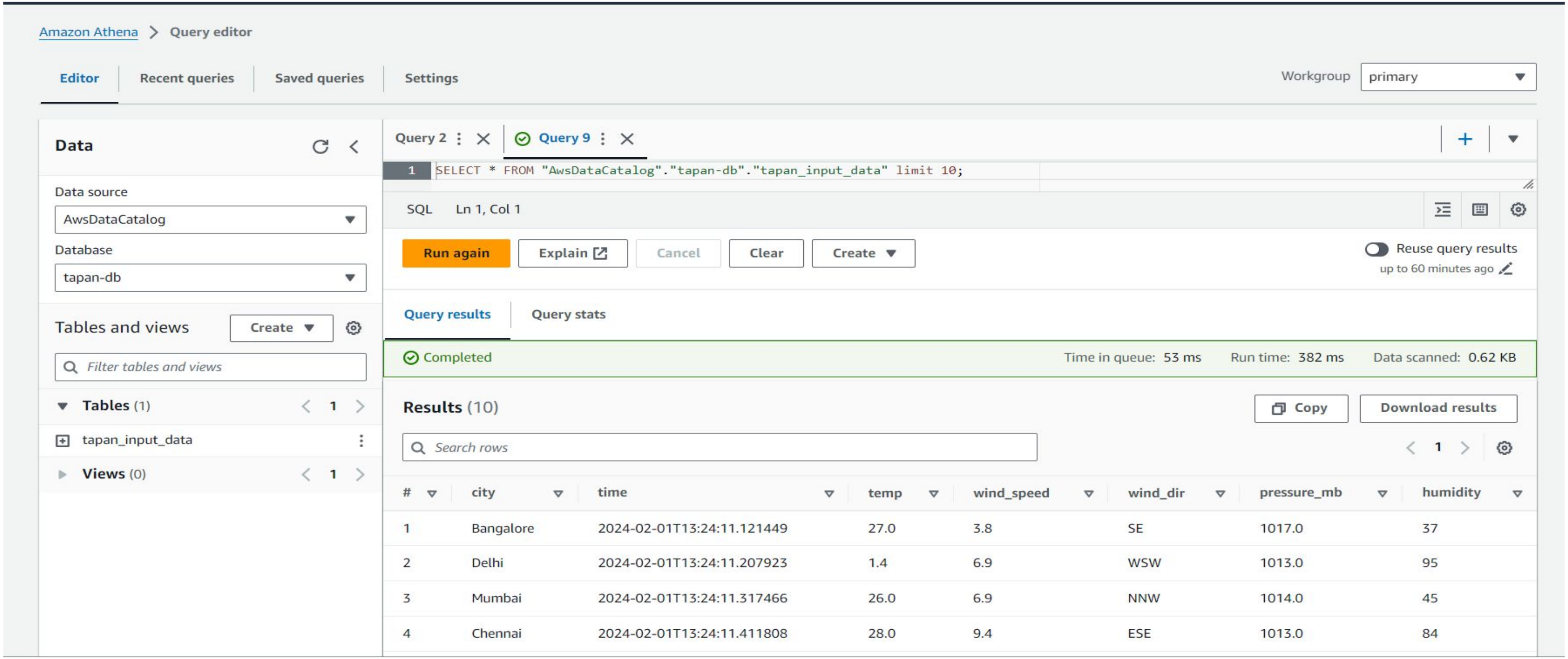
Data

- TEMP sum
- CITY none: Series
- TIME none: X-Axis

Appearance

☑ Show Legend

Final Data : Athena



Workgroup: primary

Data source: AwsDataCatalog

Database: tapan-db

```
SELECT * FROM "AwsDataCatalog"."tapan-db"."tapan_input_data" limit 10;
```

Completed   Time in queue: 53 ms   Run time: 382 ms   Data scanned: 0.62 KB

Results (10)

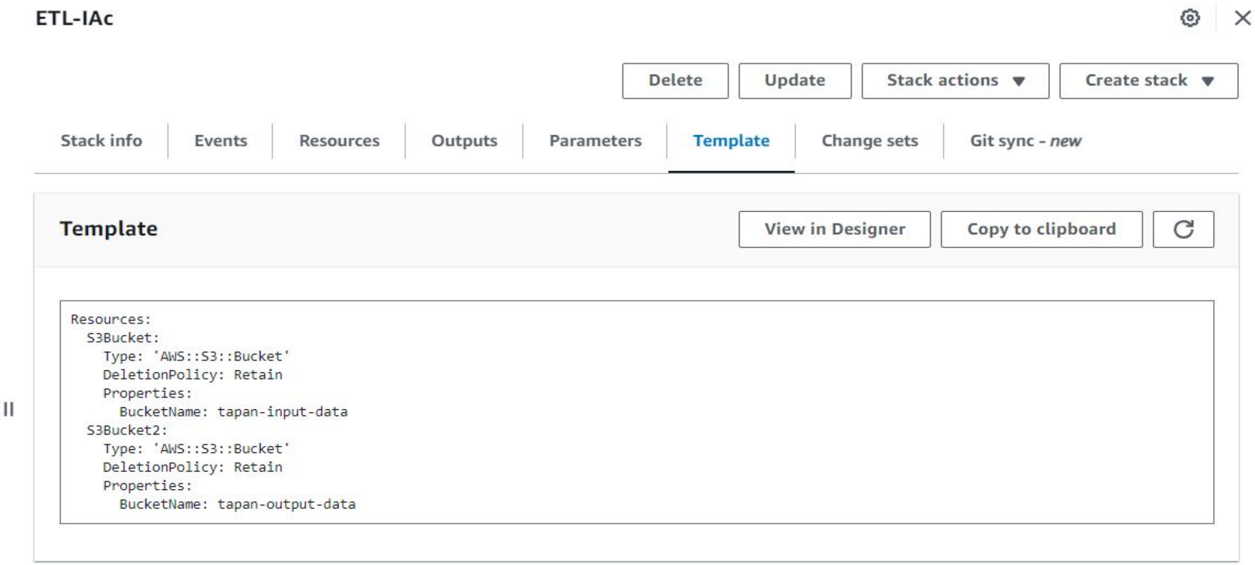
#	city	time	temp	wind_speed	wind_dir	pressure_mb	humidity
1	Bangalore	2024-02-01T13:24:11.121449	27.0	3.8	SE	1017.0	37
2	Delhi	2024-02-01T13:24:11.207923	1.4	6.9	WSW	1013.0	95
3	Mumbai	2024-02-01T13:24:11.317466	26.0	6.9	NNW	1014.0	45
4	Chennai	2024-02-01T13:24:11.411808	28.0	9.4	ESE	1013.0	84

IaC (Infrastructure as Code)

In our infrastructure we have used Cloud Formation for implementing Infrastructure as code.

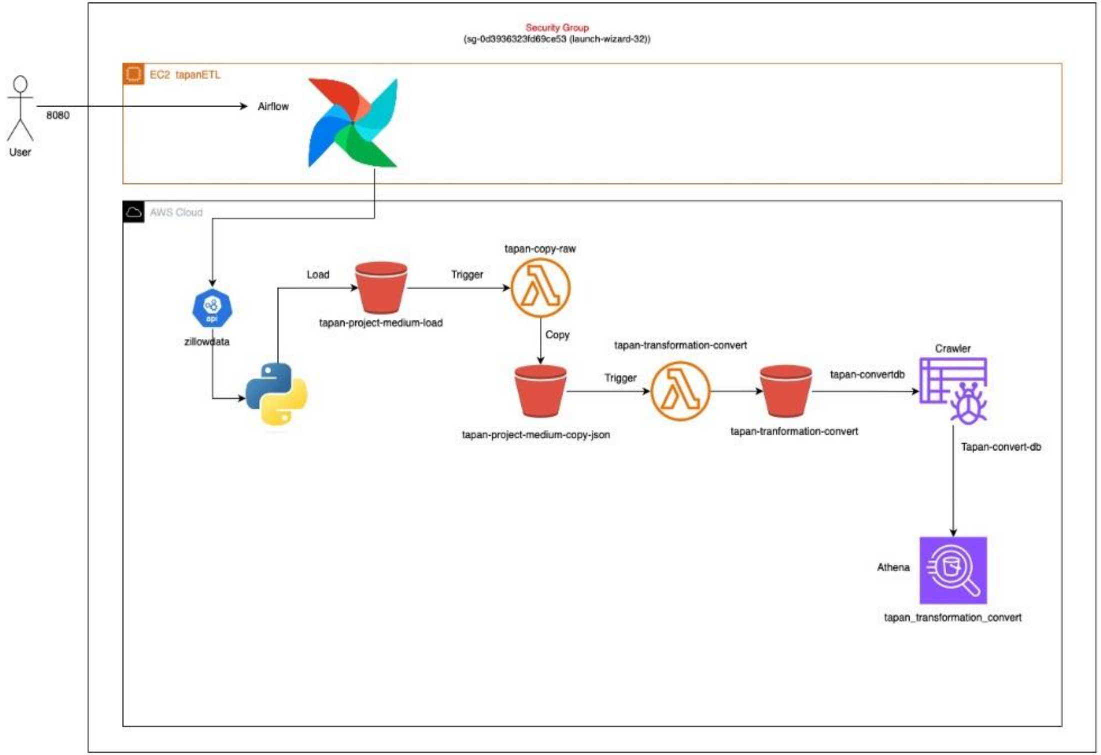
Created two S3 Buckets using this code.

- tapan-input-data
- tapan-output-data



```
Resources:
  S3Bucket:
    Type: 'AWS::S3::Bucket'
    DeletionPolicy: Retain
    Properties:
      BucketName: tapan-input-data
  S3Bucket2:
    Type: 'AWS::S3::Bucket'
    DeletionPolicy: Retain
    Properties:
      BucketName: tapan-output-data
```

ETL: Pipeline Using Airflow


Security Group
(sg-0d3936323fd69ce53 (launch-wizard-32))
EC2 tapanETL
User
8080
Airflow
AWS Cloud
zillowdata
Load
tapan-project-medium-load
Trigger
tapan-copy-raw
Copy
tapan-project-medium-copy-json
Trigger
tapan-transformation-convert
tapan-tranformation-convert
tapan-convertdb
Crawler
Tapan-convert-db
Athena
tapan_transformation_convert

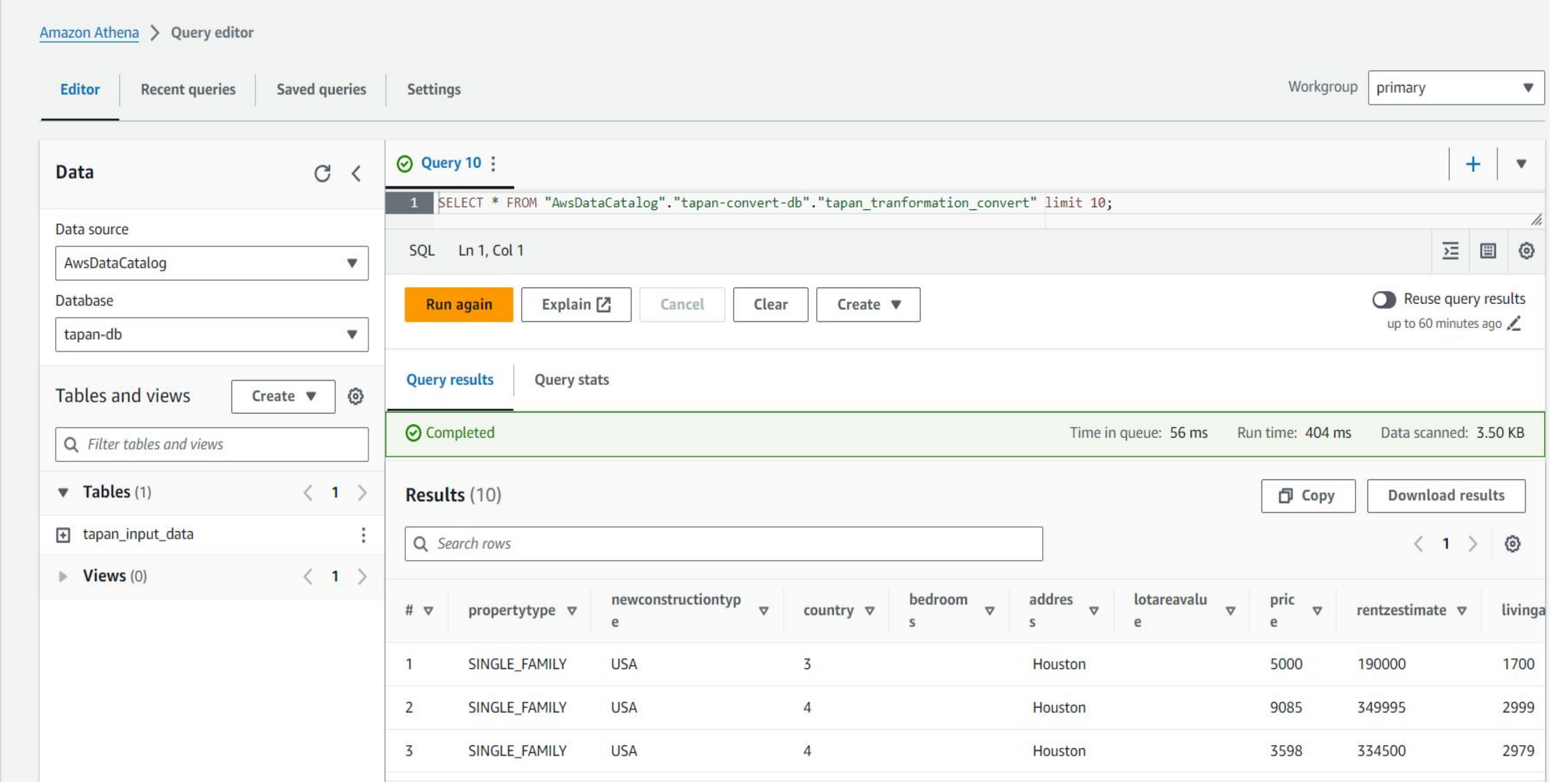
Services Used:

- Zillow Rapid API
- EC2 : Medium Instance
- Apache Airflow
- S3 Buckets
- Lambda Functions
- AWS Crawler
- Athena

Amazon Services Overview:

1. **Zillow API**: Get real time Housing data in .json format.
2. Airflow : We are using Python operators to fetch and load data into S3 bucket.
3. **S3 Bucket** (*tapan-project-medium-load*): Get data from the API (**Extract**) and getting loaded in this bucket.
4. **Lambda Function (tapan_copy_raw)**: To copy the data we get from API to new S3 Bucket.
5. **S3 Bucket** *(tapan-project-medium-copy-json*): json stored in previous Bucket get copied to this Bucket.
6. **Lambda Function** (**tapan-transformation-convert**): Transform json data to csv and pass it to S3 Bucket.
7. **S3 Bucket** (*tapan-transformation-convert*): Generated csv file will get stored in this Bucket for further use.
8. **AWS Crawler**:(*tapan-convert-db*) An AWS Glue component that automatically discovers, classifies, and extracts metadata from various data sources, facilitating efficient data cataloging and ETL processes.
9. **AWS Athena**:(*tapan_transformation_convert*) A serverless, interactive query service that enables users to analyze data in Amazon S3 using standard SQL, without the need for infrastructure management.

Final Output : using Apache AirFlow



Workgroup: primary

Data

Data source: AwsDataCatalog

Database: tapan-db

Tables and views

Filter tables and views

- tapan_input_data

Query 10

```
SELECT * FROM "AwsDataCatalog"."tapan-convert-db"."tapan_tranformation_convert" limit 10;
```

SQL Ln 1, Col 1

Completed Time in queue: 56 ms Run time: 404 ms Data scanned: 3.50 KB

Results (10)

Search rows

#	propertytype	newconstructiontype	country	bedrooms	address	lotareavalue	price	rentzestimate	livinga
1	SINGLE_FAMILY	USA	3		Houston		5000	190000	1700
2	SINGLE_FAMILY	USA	4		Houston		9085	349995	2999
3	SINGLE_FAMILY	USA	4		Houston		3598	334500	2979

THANK YOU