Project Report: Renewable Energy Usage Analysis with AWS S3, Snowflake, and Tableau

Project Overview

The objective of this project was to build an **end-to-end cloud-based analytics pipeline** to ingest renewable energy usage data, transform it for KPI generation, and visualize insights on cost efficiency and adoption trends.

We used:

- **AWS S3** for scalable storage and data ingestion.

- **Snowflake** for cloud-native data warehousing and transformation.

- **Tableau** for creating interactive dashboards and publishing to the cloud.

Key Outcomes:

- Processed **3,000+ renewable energy records** from multiple regions.

- Generated **10+ Key Performance Indicators (KPIs)**, including cost efficiency, adoption rates, and regional usage trends.

- Identified **Wind** as the most cost-efficient energy source at **$0.39/kWh**.

- Found **post-2020 adopters** saved **2.5% more** on average.

2. Technology Stack & Rationale

Technology	Purpose	Why Chosen
AWS S3	Store and stage raw CSV datasets before processing.	Highly scalable, easy integration with Snowflake, pay-per-use.
Snowflake	Store and transform datasets using SQL.	Automatic scaling, support for semi-structured data, zero-copy cloning.
Tableau	Data visualization & dashboard creation.	Industry-standard BI tool, strong Snowflake integration, fast interactivity.

3. Data Pipeline Execution

Step 1 – Uploading Data to AWS S3

The dataset was stored in an **S3 bucket** named renewable-energy-usage.

We used the **AWS CLI** to upload the CSV:

```
aws s3 cp Renewable_Energy_Usage_Sampled.csv s3://renewable-energy-usage/
```

AWS S3 served as the **staging area** for Snowflake ingestion.

Step 2 – Creating Snowflake Integration

We created an **AWS S3 Storage Integration** in Snowflake to establish a secure connection:

```
CREATE STORAGE INTEGRATION s3_integration

TYPE = EXTERNAL_STAGE

STORAGE_PROVIDER = S3

ENABLED = TRUE

STORAGE_AWS_ROLE_ARN = 'arn:aws:iam::123456789012:role/snowflake_s3_access'

STORAGE_ALLOWED_LOCATIONS = ('s3://renewable-energy-usage/');
```

Then we **updated the trust policy** in AWS IAM to grant Snowflake access.

Step 3 – Creating an External Stage in Snowflake

```
CREATE STAGE renewable_stage

STORAGE_INTEGRATION = s3_integration

URL = 's3://renewable-energy-usage/';
```

Step 4 – Loading Data into Snowflake

We created a table to store the dataset:

```
CREATE OR REPLACE TABLE renewable_energy_usage (

    Household_ID STRING,

    Region STRING,

    Country STRING,

    Energy_Source STRING,

    Monthly_Usage_kWh FLOAT,

    Year INT,

    Household_Size INT,

    Income_Level STRING,

    Urban_Rural STRING,

    Adoption_Year INT,

    Subsidy_Received STRING,

    Cost_Savings_USD FLOAT
);
```

We then loaded data from S3:

```
COPY INTO renewable_energy_usage

FROM @renewable_stage/Renewable_Energy_Usage_Sampled.csv

FILE_FORMAT = (TYPE = 'CSV' FIELD_OPTIONALLY_ENCLOSED_BY='"' SKIP_HEADER=1);
```

4. Data Understanding & Cleaning

We performed **exploratory data analysis** using Snowflake SQL:

-- Check unique energy sources

SELECT DISTINCT Energy_Source FROM renewable_energy_usage;

-- Check for missing values

SELECT COUNT(*) - COUNT(Monthly_Usage_kWh) AS missing_usage

FROM renewable_energy_usage;

Findings:

- No missing values in key fields.

- Energy sources included: Solar, Wind, Hydro, Geothermal, Biomass.

- Data covered 2010–2024 adoption years.

5. Data Transformation & KPI Generation

KPI 1 – Cost per kWh by Energy Source

```
SELECT Energy_Source,

    ROUND(AVG(Cost_Savings_USD / Monthly_Usage_kWh), 2) AS Cost_per_kWh

FROM renewable_energy_usage

GROUP BY Energy_Source

ORDER BY Cost_per_kWh;
```

Insight: Wind energy had the lowest cost per kWh ($0.39).

KPI 2 – Savings Difference for Post-2020 Adopters

```
SELECT CASE WHEN Adoption_Year > 2020 THEN 'Post-2020' ELSE 'Pre-2020' END AS
Adoption_Period,

      ROUND(AVG(Cost_Savings_USD), 2) AS Avg_Savings

FROM renewable_energy_usage

GROUP BY Adoption_Period;
```

Insight: Post-2020 adopters saved 2.5% more.

KPI 3 – Regional Usage Trends

```
SELECT Region, SUM(Monthly_Usage_kWh) AS Total_Usage

FROM renewable_energy_usage

GROUP BY Region

ORDER BY Total_Usage DESC;
```