Mini Project: IT Asset Data Operations & Insights

Overview

This mini-project gives you hands-on experience in **data engineering and analytics workflows** — cleaning raw data, indexing it into Elasticsearch, transforming and enriching it using Python, and finally building meaningful business insights through visualizations.

You will work with an intentionally "messy" dataset that simulates real-world enterprise data challenges.

Dataset

File Name: `it_asset_inventory_enriched.csv`

About the dataset:
Each record represents an IT asset with details such as hostname, country, operating system, lifecycle status, installation date, and other operational attributes.

PHASE 1 — Excel Data Cleaning

Objective

Clean the dataset in Excel and prepare it for ingestion into Elasticsearch.

Tasks

1. **Open the CSV file** in Excel.
2. **Remove duplicate rows** based on the `hostname` field.
    - Use → **Data → Remove Duplicates**
3. **Trim extra spaces** from text fields:
    - Apply `=TRIM(A2)` or use "Flash Fill".
4. **Handle blanks and missing values:**
    - Replace empty cells with `Unknown`
5. **Check date format** under `operating_system_installation_date` → ensure consistent `YYYY-MM-DD`.
6. Save the cleaned data as:
   `it_asset_inventory_cleaned.csv`

💡 *Tip:* Record every Excel function you use — it will go in your final README file.

PHASE 2 — Indexing Data to Elasticsearch Using Python

Objective

Write a Python script to load your cleaned CSV data into Elasticsearch.

Steps

1. Create a new **GitHub repository** named `data-operations-it-assets`.
2. Inside your project folder, create a script: **`index_data.py`**

4. Test your code to ensure data is indexed successfully.
5. Commit and push `index_data.py` to your GitHub repo.

PHASE 3 — Data Transformation & Enrichment in Elasticsearch

Objective

Enhance your indexed data using Elasticsearch scripting and transformations.

Tasks

Create another Python script — **`transform_data.py`** — to perform the following:

1. **Reindex data to another index**
2. **Add a derived field**:
   - `risk_level = "High"` if `operating_system_lifecycle_status` is `"EOL"` or `"EOS"`,
     else `"Low"`.
3. **Calculate system age** (in years) from the installation date.
4. **Delete records** that have missing hostnames or `Unknown` providers.
5. **Update existing records** with the new fields using `_update_by_query`.

PHASE 4 — Visualization and Insights

Objective

Build business insights and visualizations from your final dataset.

Tasks

1. Export or view data in **Kibana**.
2. Create charts such as:
    - **Assets by Country**
    - **Lifecycle Status Distribution**
    - **High vs Low Risk Assets**
    - **Top OS Providers**
3. Save screenshots of your dashboards in a folder:
   📁 `visualization_screenshots/`
4. Write short business insights — e.g.:

   "40% of assets are EOL — indicating an urgent need for OS upgrades, especially in INDIA and BRAZIL."

PHASE 5 — GitHub Submission

Your GitHub repository structure should look like this:

```
data-operations-it-assets
├── it_asset_inventory_cleaned.csv
├── index_data.py
├── transform_data.py
├── visualization_screenshots/
├── README.md
└── final_report.md
```

README.md must include:

- Overview of each phase
- Excel cleaning techniques used
- Python scripts and their purpose
- Screenshots of successful indexing, transformations, and visualizations
- Final business insights and learnings

Expected Outcome

By the end of this project, you will have:

✔ Cleaned real-world messy data using Excel functions
✔ Written Python scripts to index and enrich data in Elasticsearch
✔ Used Git for version control and collaboration
✔ Built visual dashboards for IT asset insights
✔ Derived meaningful business recommendations