User Manual: Data Indexing into Elasticsearch

Introduction

This guide provides instructions on using the Python script (exported from a Jupyter notebook) to index a processed dataset into Elasticsearch. The script reads data from a CSV file, preprocesses it, and uploads it to an Elasticsearch index in bulk.

---

1. Prerequisites

Requirements

1. **Elasticsearch**:
   - Ensure Elasticsearch is running and accessible.
   - Provide connection details (host, port, credentials, and certificate path) in the script.
2. **CSV Dataset**:
   - The dataset to index should be stored in a file named `final_features_flows.csv` within the `csv_files` directory.
   - Columns like `bidirectional_first_seen_ms`, `bidirectional_last_seen_ms`, `src_port`, `dst_port`, `src_ip`, and `dst_ip` will be dropped during preprocessing.
3. **Dependencies**:
   - Install required Python packages:

     ```
     pip install pandas elasticsearch
     ```

---

2. How to Use the Script

Step 1: Setup Elasticsearch Connection

- Update the connection parameters:
  - `elastic_host`: Elasticsearch server address.
  - `elastic_port`: Server port.
  - `elastic_user` and `elastic_password`: Credentials.
  - `elastic_ca_path`: Path to the CA certificate for secure HTTPS connections.

Step 2: Specify Dataset and Index

- Modify the following variables:
  - `csv_file_name`: Path to the CSV file (default is `csv_files/final_features_flows.csv`).
  - `index_name`: Elasticsearch index name where data will be uploaded (default is `network_flows_fan_encoded_final`).

Step 3: Run the Script

- Run the notebook in colab or jupyter :

```
2b_data_indexing.ipynb
```

- The script will:
  - Load and preprocess the dataset.
  - Delete the specified index in Elasticsearch (if it exists) and create a new one.
  - Batch index the data into Elasticsearch.

---

3. Detailed Workflow

3.1 Establish Elasticsearch Connection

- The script connects to Elasticsearch using the provided credentials and certificate.
- Connection is verified, and details are printed to confirm successful initialization.

3.2 Load and Preprocess Data

- The dataset is read from the specified CSV file (`final_features_flows.csv`).
- Columns like `bidirectional_first_seen_ms`, `bidirectional_last_seen_ms`, `src_port`, `dst_port`, `src_ip`, and `dst_ip` are removed before indexing to reduce redundancy.

3.3 Manage Elasticsearch Index

- The script checks if the specified index (`network_flows_fan_encoded_final`) exists:
  - If it exists, it is deleted.
  - A new index is created.

3.4 Index Data in Batches

- Data is prepared as a list of documents (`actions`), each representing a row in the dataset.
- Batches of 50 documents are indexed into Elasticsearch using `helpers.bulk()`.

---

4. Outputs

Console Logs

- Connection details and status:
  - Verification of Elasticsearch connection.
  - Successful deletion and creation of the index.
- Progress of data indexing:
  - Confirmation of completed indexing with the total documents indexed.

---

5. Customization Options

5.1 Change CSV File

- To index a different dataset, update `csv_file_name` with the new file path.

5.2 Modify Index Name

- Change `index_name` to specify a new Elasticsearch index.

5.3 Adjust Batch Size

- Modify the `batch_size` variable to change the number of documents indexed in each batch.

5.4 Add/Remove Preprocessing Steps

- Update the `drop` operation in the `df.drop(columns=...)` step to include or exclude more columns before indexing.

---

6. Troubleshooting

Common Issues

1. **Connection Errors**:
   - Verify Elasticsearch credentials, host, port, and certificate path.
   - Ensure the Elasticsearch server is running.
2. **Missing Dataset**:
   - Confirm the CSV file exists in the specified path.
3. **Indexing Errors**:
   - Check if the dataset contains unsupported data types or missing values that might cause issues during indexing.

---

7. Example Execution

Expected Console Output

```
Connected to Elasticsearch
Index network_flows_fan_encoded_final deleted.
Index network_flows_fan_encoded_final created with specified mapping.
Indexing in 'network_flows_fan_encoded_final' finished.
```

The index will be populated with data from the specified CSV file and ready for querying. For further customization or troubleshooting, refer to the inline comments in the script.