Technical task

Create a data pipeline using Apache Airflow to perform an ETL (Extract, Transform, Load) process for zoo animal data. The pipeline should:

1. Extract animal data from multiple CSV files.
2. Transform the data with various operations.
3. Aggregate and validate the data.
4. Load the final transformed and validated data into a new CSV file.

Task Details

1. **Setup Apache Airflow:**
   - Install Apache Airflow.
   - Configure a basic Airflow setup with a `DAG` (Directed Acyclic Graph).
2. **Extract Tasks:**
   - Create tasks to read data from two CSV files named `zoo_animals.csv` and `zoo_health_records.csv`.
   - The `zoo_animals.csv` file will contain the following columns: `animal_id`, `animal_name`, `age`, `species`.
   - The `zoo_health_records.csv` file will contain the following columns: `animal_id`, `checkup_date`, `health_status`.
3. **Transform Tasks:**
   - Create tasks to perform the following transformations:
     1. Merge the data from `zoo_animals.csv` and `zoo_health_records.csv` based on `animal_id`.
     2. Filter out animals where the `age` is less than 2 years.
     3. Convert the `animal_name` column to the title case.
     4. Ensure the `health_status` column contains only "Healthy" or "Needs Attention".
4. **Aggregation and Validation Tasks:**
   - Aggregate the data to count the number of animals in each `species` and the number of "Healthy" vs "Needs Attention" statuses.
   - Validate the aggregated data to ensure the counts are correct and no data is missing.
5. **Load Task:**
   - Create a task to write the final transformed, aggregated, and validated data to a new CSV file named `final_zoo_data.csv`.
6. **Airflow DAG:**
   - Define a DAG in Airflow that schedules and orchestrates the Extract, Transform, Aggregate, Validate, and Load tasks.
   - Ensure the tasks are dependent on each other in the correct order: Extract -> Transform -> Aggregate -> Validate -> Load.