Homework: S4 - Integrate S3 with your FastAPI Application

Instructor: Miquel Farre

Objective

Complete the S4 assignment in the `bts-bdp-assignment` repository: implement the S3 integration endpoints so your FastAPI application downloads aircraft data and stores it in an AWS S3 bucket.

Setup

1. Fork the assignment repository: https://github.com/miquelfarre/bts-bdp-assignment
2. Clone your fork locally:

```
git clone https://github.com/YOUR-USERNAME/bts-bdp-
        assignment.git
cd bts-bdp-assignment
```

1. Create a virtual environment and install dependencies:

```
python -m venv .venv

# macOS / Linux
source .venv/bin/activate

# Windows
.venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt
```

1. Make sure the application starts:

```
uvicorn bdi_api.app:app --reload --host 0.0.0.0 --port 8080
```

Then visit http://localhost:8080/docs to verify.

What you need to do

Part 1: Create an S3 bucket

Create a bucket in your AWS Academy Lab with a name starting with `bdi-aircraft` followed by something unique (e.g. `bdi-aircraft-yourname`).

> The name must be globally unique across all AWS accounts.

Part 2: Implement the S4 endpoints

Open `bdi_api/s4/exercise.py` and implement the two endpoints:

- **`POST /api/s4/aircraft/download`**: Downloads aircraft data files from ADS-B Exchange and stores them in your S3 bucket under `raw/day=20231101/`
- **`POST /api/s4/aircraft/prepare`**: Reads the raw files from S3 and stores them locally in the `prepared` directory, so the S1 query endpoints keep working

Read the full instructions in `bdi_api/s4/README.adoc`.

Part 3: Write tests

Add your tests inside `tests/s4/`. Make sure they pass with:

```
pytest tests/s4/ -v
```

Running the application with your bucket

macOS/Linux:

```
BDI_S3_BUCKET=bdi-aircraft-yourname uvicorn bdi_api.app:app
        --reload --host 0.0.0.0 --port 8080
```

Windows (PowerShell):

```
$env:BDI_S3_BUCKET="bdi-aircraft-yourname"
uvicorn bdi_api.app:app --reload --host 0.0.0.0 --port 8080
```

Or add to a `.env` file in the project root (all platforms):

```
BDI_S3_BUCKET=bdi-aircraft-yourname
```

Prerequisites

Make sure you have AWS credentials configured as done in class:

- macOS/Linux: `~/.aws/credentials`
- Windows: `C:\Users\YOUR_USERNAME\.aws\credentials`

- For AWS Academy Learner Lab: credentials are available under **AWS Details > AWS CLI**

Verification

1. Run your application with your bucket name
2. Call `POST /api/s4/aircraft/download?file_limit=100` (via Swagger UI or curl)
3. Go to the S3 console and check that files appear under `raw/day=20231101/`
4. Call `POST /api/s4/aircraft/prepare`
5. Verify that the S1 query endpoints (`/api/s1/aircraft/`, etc.) return data

Deliverables

1. Push your code to your GitHub fork
2. Take a screenshot of your GitHub commit history and upload it to campus
3. Take a screenshot of your S3 bucket showing the files in `raw/`

Evaluation (10 points)

1. **The application does not compile/start -> 0 automatically**
2. **Virtual environments are not used -> 0 automatically**
3. **Code** (4 points): I'll run a series of validation tests against your application. Percentage of passed tests determines the score.
4. **Code quality and best practices** (2 points):
    - Code linting with `ruff`
    - Tests you have implemented inside `tests/s4/` (coverage and completeness)
    - Screenshot of your S3 bucket directory with the 1000 raw files
    - Having failing tests results in a 0 for this section
5. **Questions about the practice** (4 points): You will be asked questions about how you implemented your solution. You must be able to explain your code and the decisions you made.