Big Data Project - Infrastructure & Data Generation

This report documents the completion of infrastructure setup and synthetic data generation for the Big Data Project, including MinIO deployment, data generation with quality issues, and successful data ingestion into the bronze layer.

---

1. Environment Setup

1.1 Project Structure

The project has been organized with the following directory structure:

```
BD_Project/
├── docker-compose.yml
├── Scripts/
│   ├── generate_weather_data.py
│   ├── generate_traffic_data.py
│   └── upload_to_minio.py
└── Data/
    ├── weather_raw.csv
    └── traffic_raw.csv
```

---

2. MinIO Infrastructure Setup

2.1 Docker Installation

MinIO container image has been pulled from the official repository:

```
docker pull minio/minio
```

2.2 Docker Compose Configuration

A `docker-compose.yml` file has been created to orchestrate MinIO deployment with the following specifications:

- **API Port**: 9000
- **Console Port**: 9001
- **Persistent Storage**: minio_data volume
- **Credentials**:
  - Username: `bdprojectfcds4`
  - Password: `bdprojectfcds4`

2.3 MinIO Deployment

MinIO service has been deployed and verified:

```
docker-compose up -d
```

2.4 Bucket Creation

Three data layer buckets have been created via MinIO Console (`http://localhost:9001`):

Bucket	Purpose
bronze	Raw, unprocessed data ingestion
silver	Cleaned and validated data
gold	Final merged and simulation-ready data

---

3. Synthetic Data Generation

3.1 Weather Dataset

A synthetic weather dataset containing **5,000 records** has been generated with intentional data quality issues to simulate real-world conditions:

Data Characteristics: - Missing values: 10–15% - Duplicate records: 5% - Inconsistent date formats - Temperature outliers (unrealistic values) - Invalid numerical values (negative humidity, negative wind speed) - Categorical inconsistencies

Output File: `Data/weather_raw.csv`

3.2 Traffic Dataset

A synthetic traffic dataset containing **5,000 records** has been generated with similar data quality patterns:

Data Characteristics: - Missing area/district information - Negative or invalid speed values - Extreme vehicle count outliers - Malformed date entries - Duplicate records - Inconsistent categorical values

Output File: `Data/traffic_raw.csv`

---

4. Data Upload to MinIO (Bronze Layer)

4.1 Upload Implementation

A Python script (`upload_to_minio.py`) has been developed to handle data ingestion with the following capabilities:

- Authentication and connection to MinIO service
- Batch upload of weather and traffic datasets to the bronze bucket
- Comprehensive error handling for invalid credentials and missing buckets
- Upload verification and logging

4.2 Issue Resolution

Initial connectivity issues encountered:

Issue	Resolution
InvalidAccessKeyId error	Verified credentials match docker-compose.yml configuration
Authentication failures	Confirmed MinIO access keys and password alignment

4.3 Upload Status

Successfully completed: - `weather_raw.csv` uploaded to bronze bucket - `traffic_raw.csv` uploaded to bronze bucket - Data verified in MinIO Console

---

5. Deliverables to Team Member 2

The following artifacts have been prepared for handoff:

Deliverable	Details
MinIO Access Credentials	URL, username, password
Raw Datasets	weather_raw.csv, traffic_raw.csv
Generation Scripts	generate_weather_data.py, generate_traffic_data.py
Infrastructure Configuration	docker-compose.yml
Documentation	Complete setup and execution guide
