Project DSI321: Real-Time Data Pipeline with Visualization

Bangkok Air Quality Dashboard with Streamlit, Prefect & LakeFS

Project Overview

This project is part of the DSI321: Big Data Infrastructure course, which focuses on building scalable data pipelines using modern tools. Our project implements a near real-time air quality monitoring system for Bangkok, utilizing hourly PM2.5 and AQI data collected from the Air4Thai API. The system forecasts both current and future air pollution levels, making it useful for public health awareness, urban planning, and environmental studies.

We employ a modern data architecture comprising a combination of Prefect (workflow orchestration), LakeFS (data versioning), Streamlit (interactive data visualization), and Docker (for reproducible deployment). Furthermore, we apply ARIMA time-series forecasting to predict air quality levels 6 hours in advance. All components are containerized for ease of deployment and reproducibility.

Key Features:

- **Real-Time Data Ingestion from LakeFS**
  Air quality data (PM2.5 and AQI) is stored in a version-controlled Parquet format using LakeFS. This ensures reproducibility and complete tracking of data updates.

- Real-time ingestion of AQI and PM2.5 data from Bangkok monitoring stations
- Forecasting of **both PM2.5 and AQI values** for each station using ARIMA
- Interactive dashboard to display live readings, forecasts, and pollution trends
- Geographic visualization with AQI heatmaps across Bangkok districts
- Fully containerized setup using Docker and Docker Compose
- Reproducible and version-controlled data pipelines using LakeFS
- Automated scheduling of ingestion and forecasting flows using Prefect

Tools & Technologies

- **Interactive Visualization via Streamlit**
  The dashboard features:
  - A color-coded interactive map showing AQI levels by district
  - Time-series charts for AQI trends over recent hours
  - A dropdown to select specific districts This project leverages modern open-source tools:
- **Data Version Control with LakeFS**
  LakeFS provides Git-like version control for data, enabling rollback, reproducibility, and auditability—crucial for time-sensitive environmental monitoring.
- **Prefect**: Python-based workflow orchestration and scheduling
- **LakeFS**: Git-like version control system for data lakes
- **Streamlit**: Framework for creating interactive dashboards in Python
- **Docker**: Containerization platform to ensure consistent environments

- **JupyterLab**: Notebook interface for data exploration and testing
- **ARIMA**: Statistical time-series forecasting model for AQI and PM2.5
- **Reproducible Deployment with Docker**
  All components (Streamlit, Prefect, LakeFS) are containerized for easy setup, consistent environments, and fast deployment.

Tech Stack Summary

Layer	Tools
Orchestration	Prefect
Containerization	Docker
Data Versioning	LakeFS
Visualization	Streamlit
Forecasting Model	ARIMA
Notebook IDE	JupyterLab
Data Source	Air4Thai PM2.5 API

Data Schema

The following table describes the structure of the **processed dataset** used for forecasting and visualization in the Streamlit dashboard. This schema is a refined version of the full dataset stored in LakeFS.

Column	Data Type	Description
`timestamp`	datetime	Timestamp of the measurement
`stationID`	string	Unique station identifier
`nameTH`	string	Station name in Thai
`areaTH`	string	Area name in Thai
`district`	string	District name
`lat`	float	Latitude
`long`	float	Longitude
`AQI.aqi`	int	Air Quality Index (0–500)
`PM25.value`	float	PM2.5 concentration (µg/m³)
`year`	int	Year of the measurement
`month`	int	Month of the measurement
`day`	int	Day of the measurement
`hour`	int	Hour of the measurement (0–23)

Dataset Quality Assurance

To ensure the reliability and accuracy of the dataset used in this project, we applied a comprehensive set of quality checks. These checks help maintain data integrity for both ingestion and forecasting workflows, especially when dealing with real-time air quality data from various stations.

- A minimum of **1,000 records** across all stations
- At least **24 hours of continuous data per station**
- More than **90% completeness** across all fields
- No columns with `object` **dtype** in the dataset
- No **duplicated rows** for any station

Full Quality Check Details
► View the full notebook here: **check_data_quality.ipynb** ► **View the full notebook here: check_data_quality.ipynb**

Getting Started

Follow the steps below to set up and run the Near Real-Time Air Quality Dashboard for Bangkok:

1. Clone the Repository

```
git clone https://github.com/khwkong/dsi321_2025.git
cd dsi321_2025
```

2. Launch All Services with Docker

Ensure Docker is running, then start all containers:

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

3. Access Local Services

Once the containers are up and running, access the following services via your browser:

- **LakeFS**: http://localhost:8001
- **JupyterLab**: http://localhost:8888
- **Prefect**: http://localhost:4200
- **Streamlit**: http://localhost:8502

** Default login for LakeFS**:

Username: `access_key`
Password: `secret_key`

Before proceeding, create a LakeFS repository (one-time setup):

```
lakectl repo create lakefs://dust-concentration
```

4. Upload Initial Data to LakeFS (Required Before Forecasting & Dashboard)

You'll need to upload initial `.parquet` data into LakeFS so that the dashboard and forecast pipelines can function properly.

First, open a shell inside the Jupyter container:

```
docker exec -it dsi321-jupyter-1 bash
```

Then, run the upload script:

```
python upload.py
```

This script will:

- Locate the most recent folder inside `/home/jovyan/data/data.parquet/year=*/month=*/day=*/`
- Upload the latest day's `.parquet` files to the `dust-concentration` repository in LakeFS
- Overwrite existing files if necessary

5. Generate Initial Forecast Data (Required for Dashboard)

Before the dashboard can display forecast data, ensure LakeFS contains both real-time and forecasted datasets.

Option A: Run Scripts Manually via CLI

Enter the Jupyter container shell:

```
docker exec -it dsi321-jupyter-1 bash
```

Then run the necessary scripts:

```
python getdata.py
python forecast.py
```

Option B: Trigger Flows from the Prefect UI

If you have already deployed the flows using `deploy.py` and `deploy_ml.py`, you can also trigger them manually from the Prefect UI.

Navigate to http://localhost:4200, select each flow, and click **"Quick Run"** to execute.

6. (Optional) Schedule Flows with Prefect

You can automate the ingestion and forecasting flows to run every hour using Prefect.

Deploy the Ingestion Flow (runs at minute 25 every hour)

Enter the Jupyter container shell:

```
docker exec -it dsi321-jupyter-1 bash
```

Then run this script:

```
python deploy.py
```

Deploy the Forecasting Flow (runs at minute 27 every hour)

Enter the Jupyter container shell:

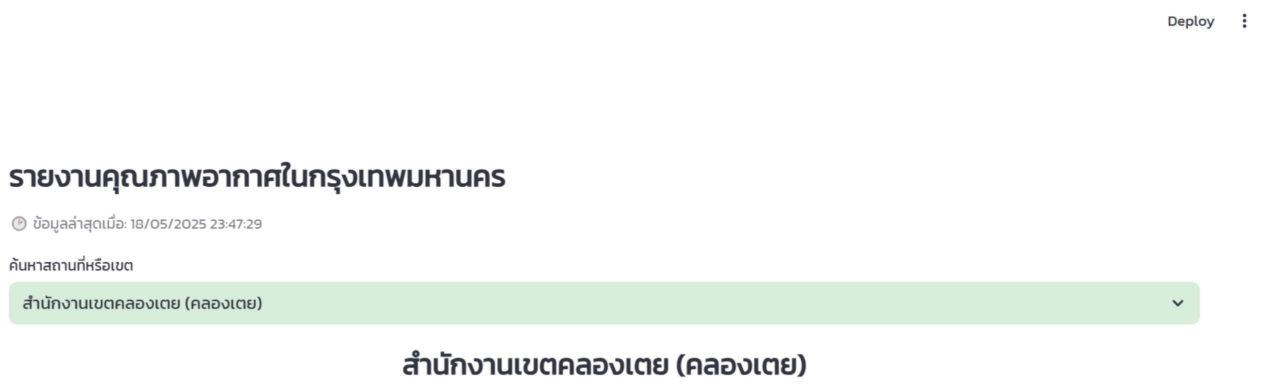
```
docker exec -it dsi321-jupyter-1 bash
```

Then run this script:

```
python deploy_ml.py
```

> These flows will execute automatically every hour if the Prefect Worker is active and new data is available in LakeFS.

Streamlit Dashboard Overview



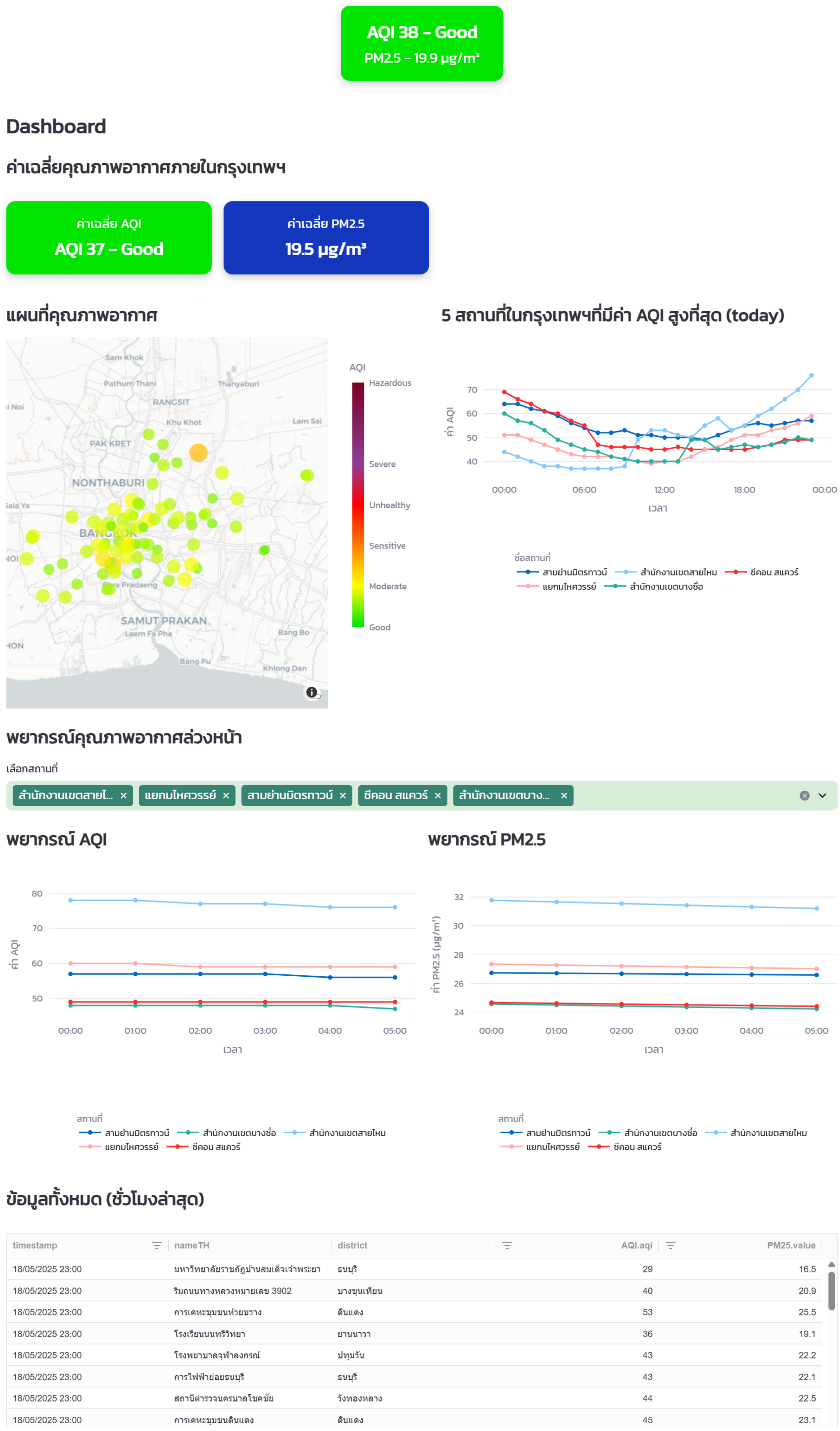
รายงานคุณภาพอากาศในกรุงเทพมหานคร

ข้อมูลล่าสุดเมื่อ: 18/05/2025 23:47:29

ค้นหาสถานที่หรือเขต

สำนักงานเขตคลองเตย (คลองเตย)

สำนักงานเขตคลองเตย (คลองเตย)



AQI 38 – Good

PM2.5 – 19.9 µg/m³

Dashboard

ค่าเฉลี่ยคุณภาพอากาศภายในกรุงเทพฯ

ค่าเฉลี่ย AQI

AQI 37 – Good

ค่าเฉลี่ย PM2.5

19.5 µg/m³

แผนที่คุณภาพอากาศ

5 สถานที่ในกรุงเทพฯที่มีค่า AQI สูงที่สุด (today)

พยากรณ์คุณภาพอากาศล่วงหน้า

เลือกสถานที่

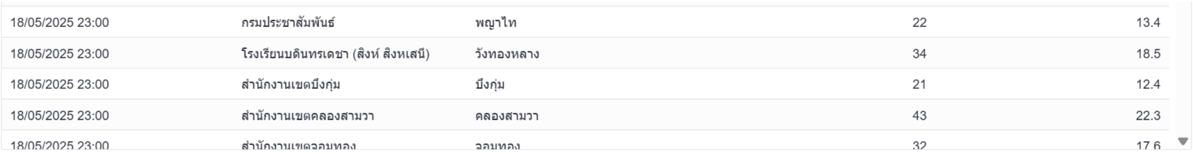
สำนักงานเขตสายไ... × แยกมไหศวรรย์ × สามย่านมิตรทาวน์ × ซีคอน สแควร์ × สำนักงานเขตบาง... ×

พยากรณ์ AQI

พยากรณ์ PM2.5

ข้อมูลทั้งหมด (ชั่วโมงล่าสุด)

timestamp	nameTH	district	AQI.aqi	PM25.value
18/05/2025 23:00	มหาวิทยาลัยราชภัฏบ้านสมเด็จเจ้าพระยา	ธนบุรี	29	16.5
18/05/2025 23:00	ริมถนนทางหลวงหมายเลข 3902	บางขุนเทียน	40	20.9
18/05/2025 23:00	การเคหะชุมชนห้วยขวาง	ดินแดง	53	25.5
18/05/2025 23:00	โรงเรียนนนทรีวิทยา	ยานนาวา	36	19.1
18/05/2025 23:00	โรงพยาบาลจุฬาลงกรณ์	ปทุมวัน	43	22.2
18/05/2025 23:00	การไฟฟ้าย่อยธนบุรี	ธนบุรี	43	22.1
18/05/2025 23:00	สถานีตำรวจนครบาลโชคชัย	วังทองหลาง	44	22.5
18/05/2025 23:00	การเคหะชุมชนดินแดง	ดินแดง	45	23.1



18/05/2025 23:00	กรมประชาสัมพันธ์	พญาไท	22	13.4
18/05/2025 23:00	โรงเรียนบดินทรเดชา (สิงห์ สิงหเสนี)	วังทองหลาง	34	18.5
18/05/2025 23:00	สำนักงานเขตบึงกุ่ม	บึงกุ่ม	21	12.4
18/05/2025 23:00	สำนักงานเขตคลองสามวา	คลองสามวา	43	22.3
18/05/2025 23:00	สำนักงานเขตจอมทอง	จอมทอง	32	17.6

The dashboard provides a city-wide overview of real-time and forecasted air quality in Bangkok.

Components:

- **Station Selector**: Choose a station to view details
- **Real-time Scorecard**: Latest AQI and PM2.5 for selected station
- **Citywide Averages**: Average AQI and PM2.5 across all stations
- **Color Map**: Map of AQI levels with color-coded bubbles
- **Line Chart**: AQI Line Chart for the most polluted station
- **Forecast Line Chart**: Multi-station forecast for AQI and PM2.5
- **Data Table**: All current readings from every station

Forecasting Logic (ARIMA)

We forecast **both AQI and PM2.5** values for each station using manually configured ARIMA models (`order=(1, 0, 1)`), implemented with the `statsmodels` package. Forecasts are generated hourly and stored in LakeFS:

```
lakefs://dust-concentration/main/forecast/forecast.parquet
```

Key Points:

- Forecast horizon: 6 hours into the future per station
- Separate ARIMA(1,0,1) models are trained for both PM2.5 and AQI
- Stations with fewer than 24 hourly records are skipped
- Outlier stations (e.g., with constant data) are excluded
- Forecasts are rounded (AQI) or kept as float (PM2.5) and saved back to LakeFS
- Forecast results are visualized in the Streamlit dashboard

Repository Structure

```
.
├── data/                           # Data directory with Parquet and schema
│   ├── data.parquet/                 # Partitioned Parquet files (LakeFS-style)
│   │   └── year=2025/
│   │       └── month=5/
│   │           └── day=XX/
│   │               └── hour=XX/
│   │                   └── <uuid>.parquet
│   ├── SCHEMA.md                     # Dataset schema documentation
│   └── check_data_quality.ipynb      # Notebook for validating data
quality
│
├── img/                            # Images for README or dashboard
preview
|   └── dashboard_demo.png/           # Screenshot of the Streamlit
dashboard
|
├── pipeline/                       # Python scripts for ingestion &
forecasting
│   ├── bangkok_districts.geojson     # GeoJSON file for Bangkok map
visualization
│   ├── deploy.py                     # Prefect flow: fetch real-time data
from API
│   ├── deploy_ml.py                  # Prefect flow: run ARIMA forecasts
per station
│   ├── forecast.py                   # ARIMA model for forecasting
│   ├── getdata.py                    # Script to retrieve and transform
API data
│   └── savedata.py                   # Dowload entire LakeFS repository
contents to local
|   └── upload.py                     # Upload latest day's `.parquet`
files to LakeFS
│
├── prefect/                        # Prefect-related configs and Docker
setup
│   ├── Dockerfile.jupyter            # Dockerfile for JupyterLab
environment
│   ├── Dockerfile.prefect-worker     # Dockerfile for Prefect flow worker
│   ├── requirements.txt              # Python dependencies for flows
│   └── wait-for-server.sh            # Helper script to wait for services
│
├── visualization/                  # Streamlit dashboard implementation
│   ├── .streamlit/
│   │   └── config.toml                 # Streamlit UI configuration
(theme, layout)
│   └── app.py                        # Main Streamlit dashboard
application
│
├── .gitignore                      # Git ignored files list
├── LICENSE                         # Project open-source license (MIT)
├── README.md                       # Main project documentation
└── docker-compose.yml              # Docker Compose to run all services
```

Contact

If you have any questions or want to learn more, feel free to reach out:

Developer: Pornnapin Thienreak

Email: pornnapin2004@gmail.com

DSI321: BIG DATA INFRASTRUCTURE | 2025

Container Docker | Version Control LakeFS | Orchestration Prefect | Dashboard Streamlit | Forecasting ARIMA