

***Data Engineering Training Batch-4***

**Project 2**

**Weather Data Analytics Pipeline using**

**Azure Databricks & Azure DevOps**

**Submitted by**

**Sumit Sanjay Jadhav**

**Saravana Kumar**

**Siddavatam Lakshmi Pranathi**

**Varsha Vardini**

**Date: 17/07/2025**

**Table of Contents**

|  |  |
| --- | --- |
| **Sr. No.** | **Topic** |
| **1.** | **Abstract** |
| **2.** | **Problem Statement** |
| **3.** | **Tools & Technologies Used** |
| **4.** | **Dataset Overview** |
| **5.** | **Architecture** |
| **6.** | **ETL Pipeline Breakdown** |
| **7.** | **ML Model & Evaluation** |
| **8.** | **Azure DevOps CI/CD** |
| **9.** | **Challenges Faced** |
| **10.** | **Outcome & Learnings** |
| **11.** | **Appendix** |

**1. Abstract**

This project focuses on building an end-to-end **Weather Data Analytics Pipeline** using Azure Data Factory, Azure Databricks (streaming), Delta Lake Medallion Architecture, and Azure DevOps CI/CD. It involves ingesting historical weather datasets, performing data transformations across Bronze, Silver, and Gold layers, training machine learning models for temperature prediction, and automating workflows using DevOps.

**2. Problem Statement**

The goal of this project is to design a modern data pipeline for processing historical weather data using cloud-native tools, and to build machine learning models for predicting weather parameters like temperature.

This project covers:

* ETL pipeline (Extract, Transform, Load) using Azure Data Factory + Azure Databricks (streaming)
* Feature engineering with Delta Lake (Bronze, Silver, Gold layers)
* Machine learning model training and tuning (RandomForest, Hyperopt)
* Integration via Azure DevOps

**3. Tools & Technologies Used**

* **Azure Data Factory** – For data ingestion
* **Azure Data Lake Storage Gen2** – For storage
* **Azure Databricks** – For streaming & transformations (PySpark + Delta Lake)
* **Scikit-learn + Hyperopt** – For machine learning
* **Azure DevOps** – For source control and CI/CD pipelines

|  |  |
| --- | --- |
| **Component** | **Service** |
| Data ingestion | Azure Data Factory |
| Storage | Azure Data Lake Storage Gen2 |
| Data processing | Azure Databricks (PySpark + Delta Lake) |
| Machine Learning | Databricks + Scikit-learn + Hyperopt |
| pipelines | Azure DevOps (Git + Pipelines) |

**4. Dataset Overview**

**📎 Dataset:** <https://www.kaggle.com/datasets/selfishgene/historical-hourly-weather-data>  
**Files Used:**

* city\_attributes.csv
* humidity.csv
* temperature.csv
* pressure.csv
* wind\_speed.csv
* wind\_direction.csv
* weather\_description.csv

**5. Architecture**

Kaggle CSVs → Azure Data Factory → ADLS Gen2

↓

Databricks Streaming (AutoLoader)

↓

Bronze → Silver → Gold Delta Tables

↓

ML Training & Evaluation

↓

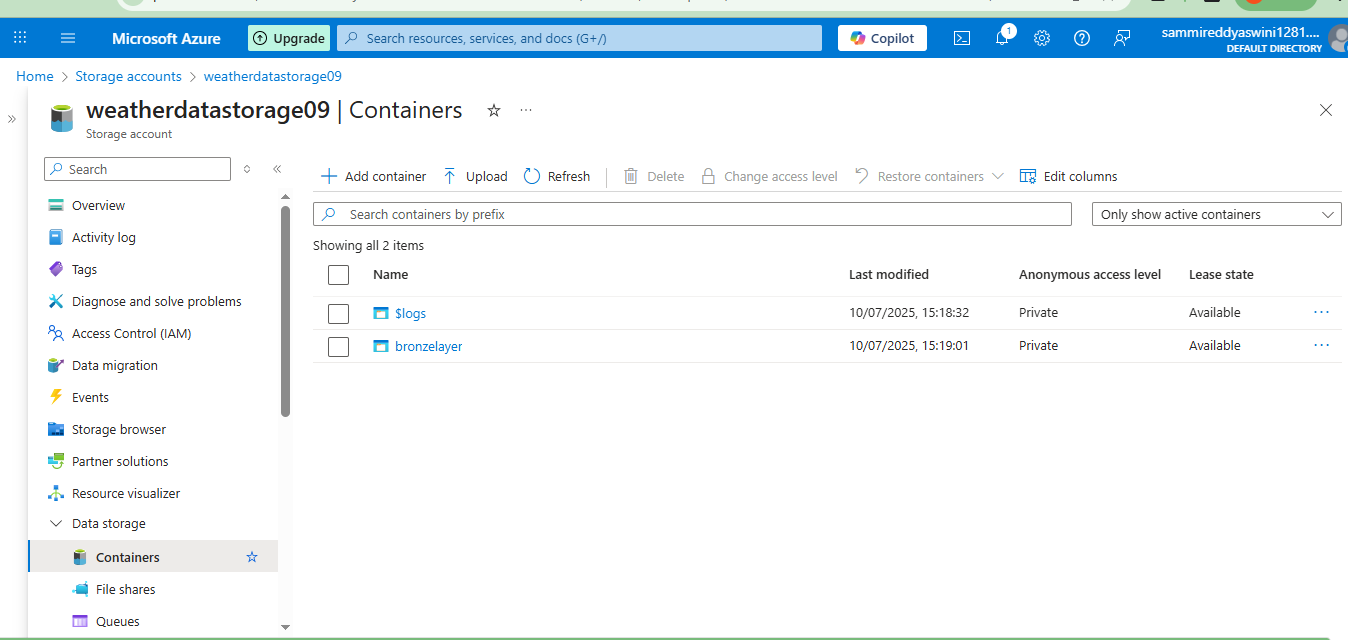
Azure DevOps Pipelines

**6. ETL Pipeline Breakdown:**

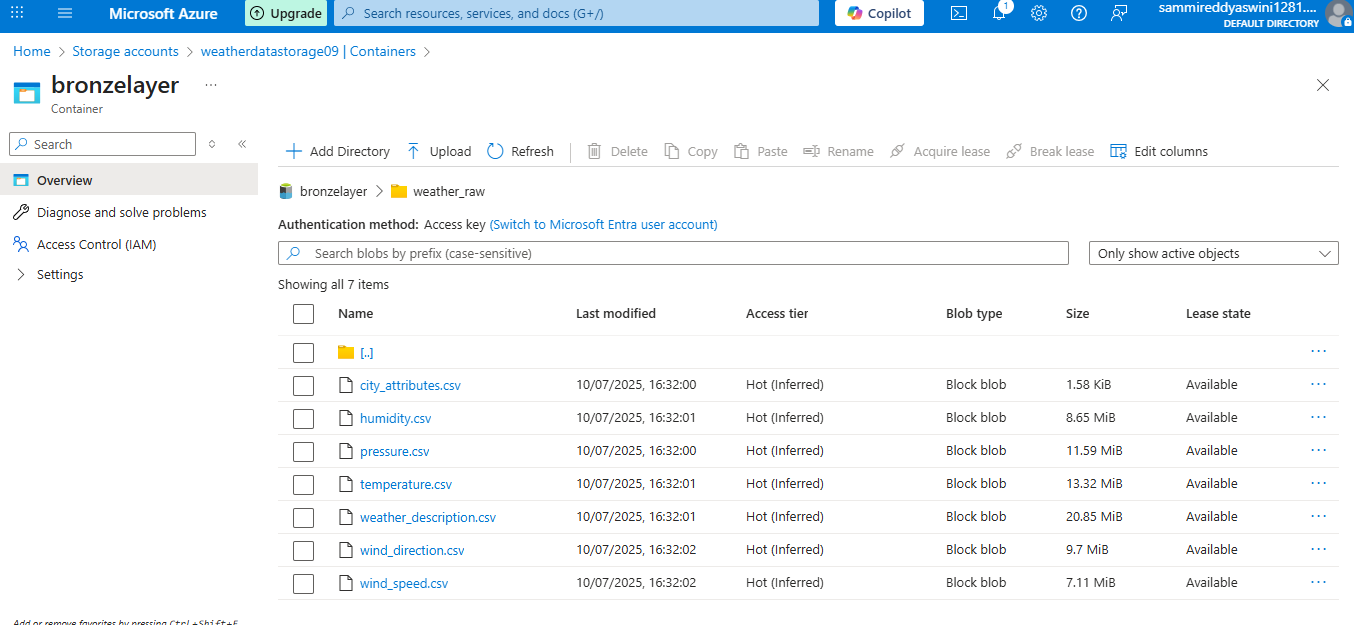
**Data Ingestion: Azure Data Factory → ADLS Gen2**

Creating a data storage and container

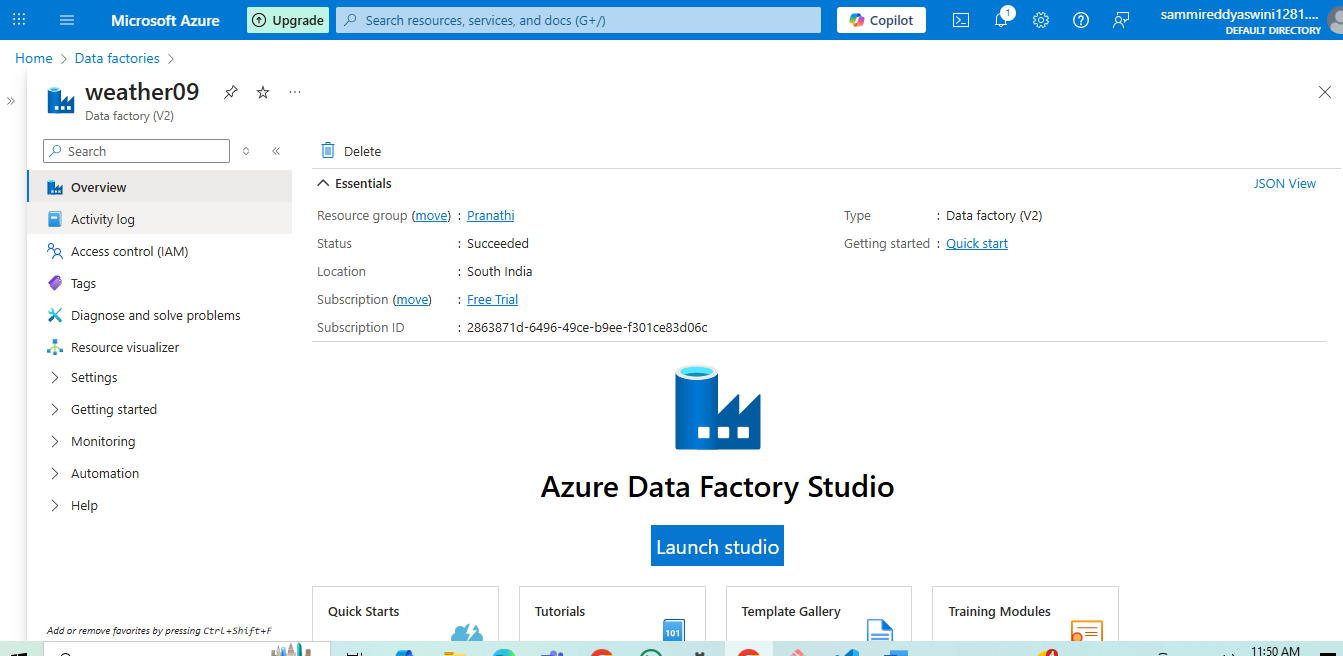
(**NOTE:** Enable hierarchy while creating storage account)



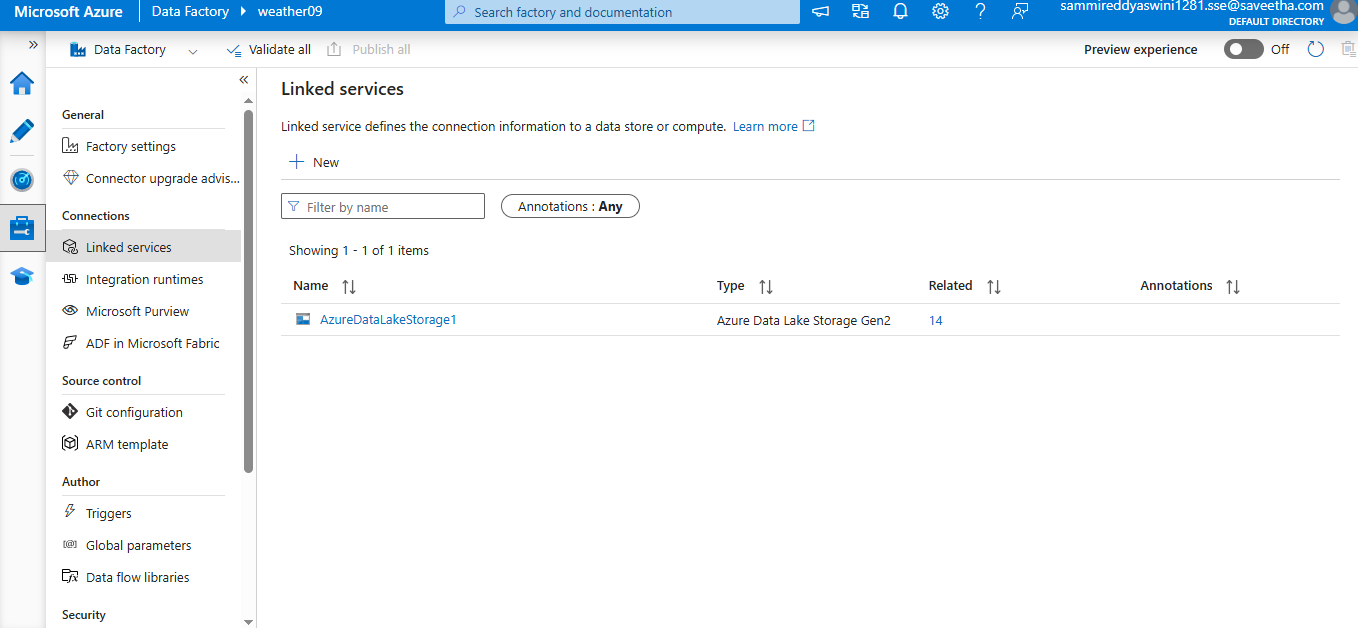
Upload data to the specified directory



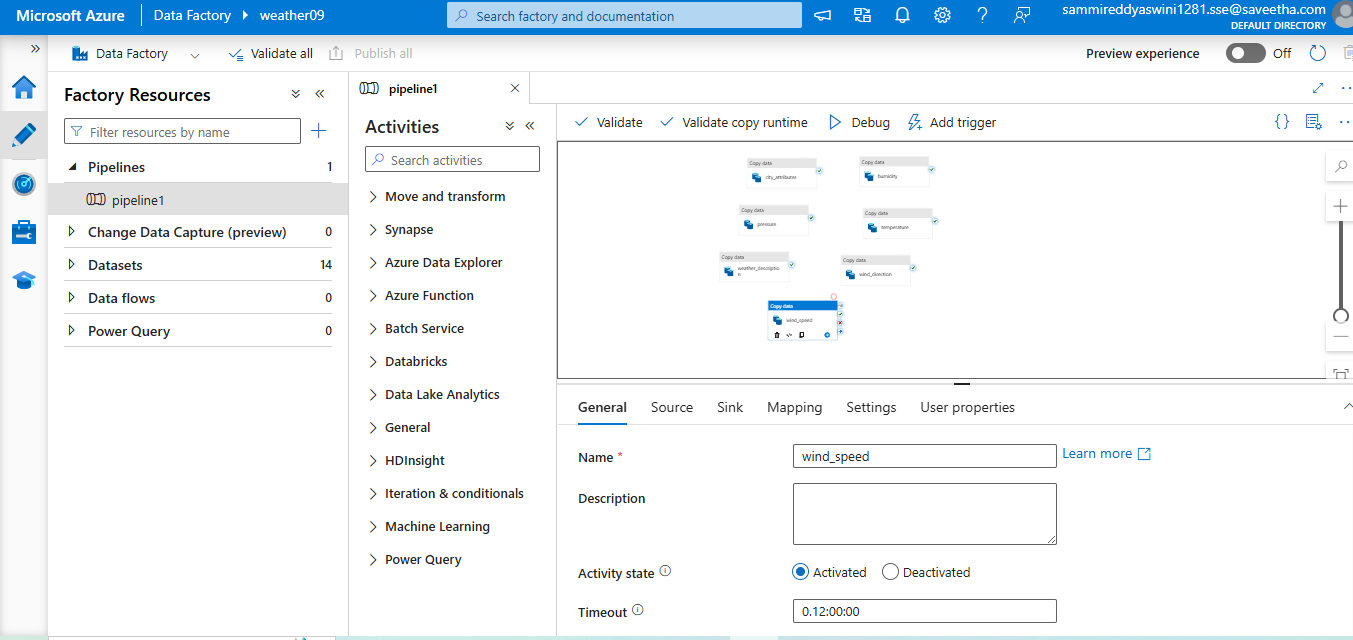
Create a Data Factory



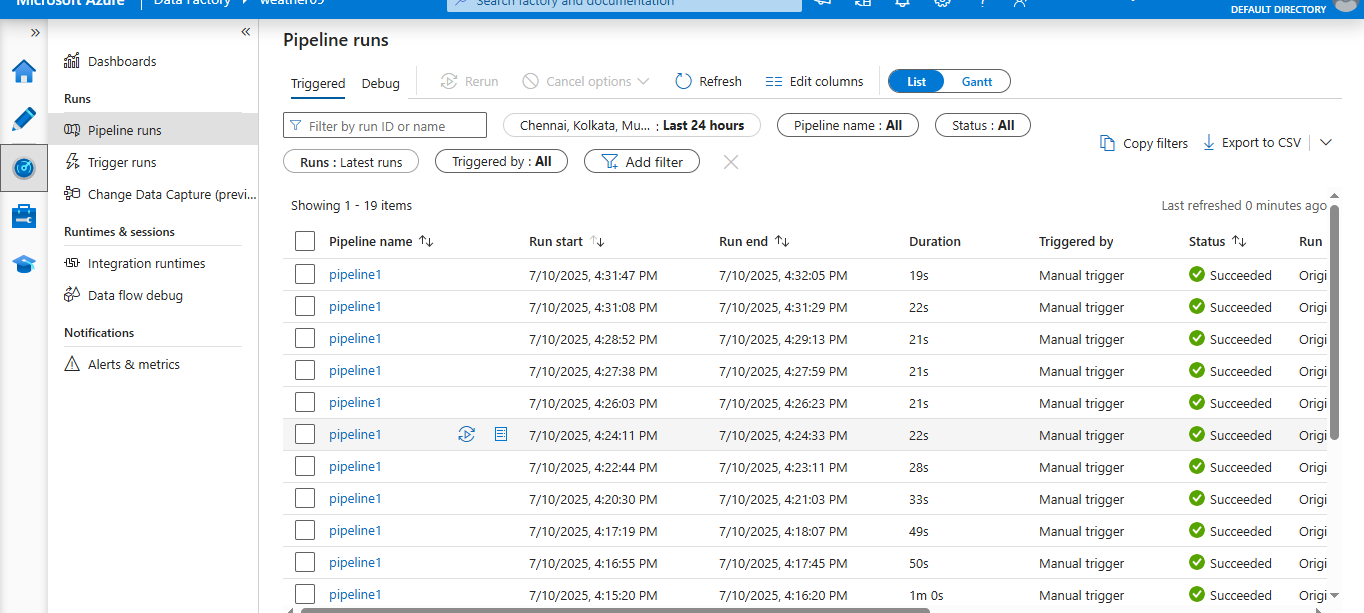
Launch Studio - Created a Linked Service



Create a Pipeline & copy the data



Pipeline Succeeded



**ETL Pipeline (Bronze → Silver → Gold)**

**1. Bronze Layer: Raw Ingestion**

* Used Azure Data Factory with 7 *Copy Data* activities (one per CSV)
* Stored raw data into ADLS container: bronzelayer/weather\_raw
* Used Databricks Autoloader to read .csv files and store as Delta Tables in /bronze

**2. Silver Layer: Transform**

* Unpivoted city-wide columns into long format (location, value)
* Joined humidity and temperature on datetime, location
* Stored result to /silverlayer/

**3. Gold Layer: ML Features**

* Loaded and unpivoted pressure.csv, wind\_speed.csv
* Joined with silver to form final feature set
* Stored to /goldlayer/

ETL Workbook: [ETL](https://colab.research.google.com/drive/1zXLrq7yMg5Kppn6kk35EqLAK95AzauIj)

**7. ML Model & Evaluation**

**Machine Learning**

**Model Objective**

Predict **temperature** using features: humidity, pressure, wind\_speed

**Models Trained**

* **Linear Regression**
* **Random Forest Regressor** (best performing)

**Optimization**

* Used **Hyperopt** to tune:
  + n\_estimators: [50, 100, 200]
  + max\_depth: [5, 10, 15]

**Evaluation Metric**

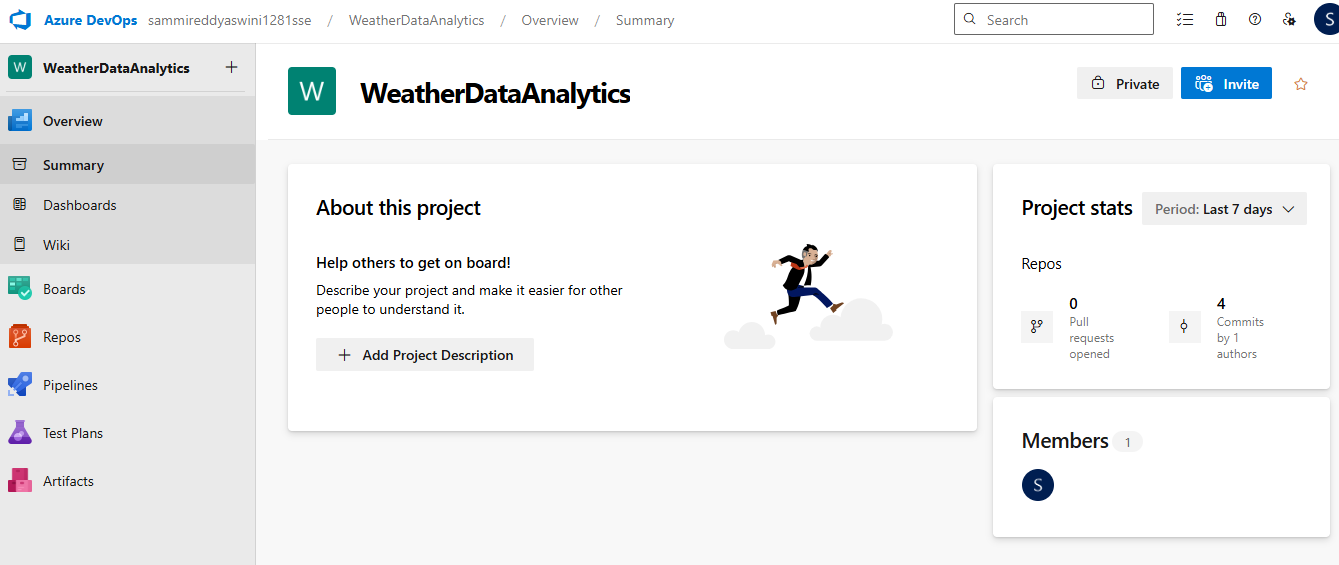
* **RMSE (Root Mean Squared Error)**

ML Workbook: [ML\_notebook](https://colab.research.google.com/drive/15T732YNMOGsn1Winka3tDOaJTxWdFT1r)

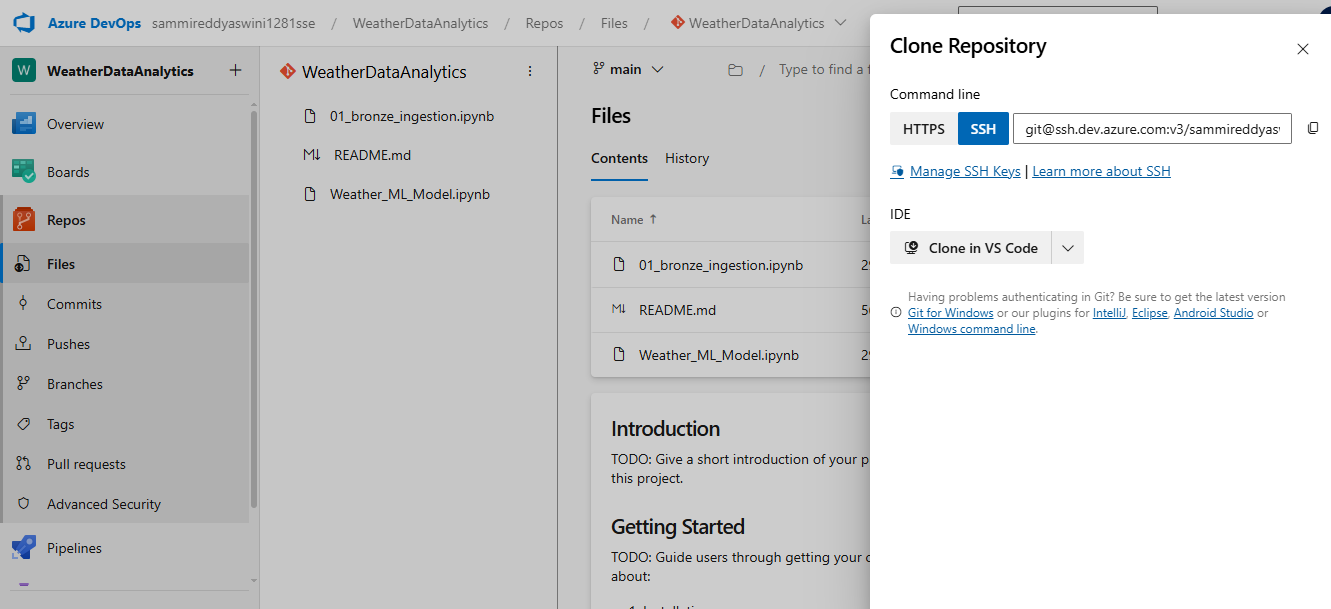
**8. Azure DevOps CI/CD**

**Azure DevOps**

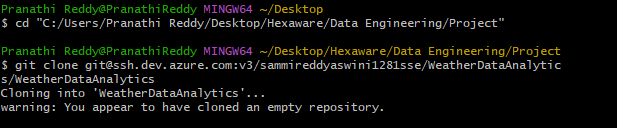
Create an Project in the organization

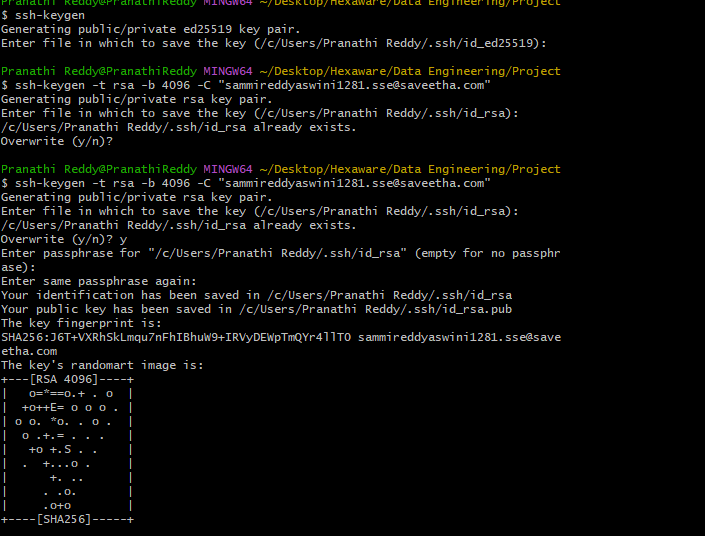
****

Clone Repository

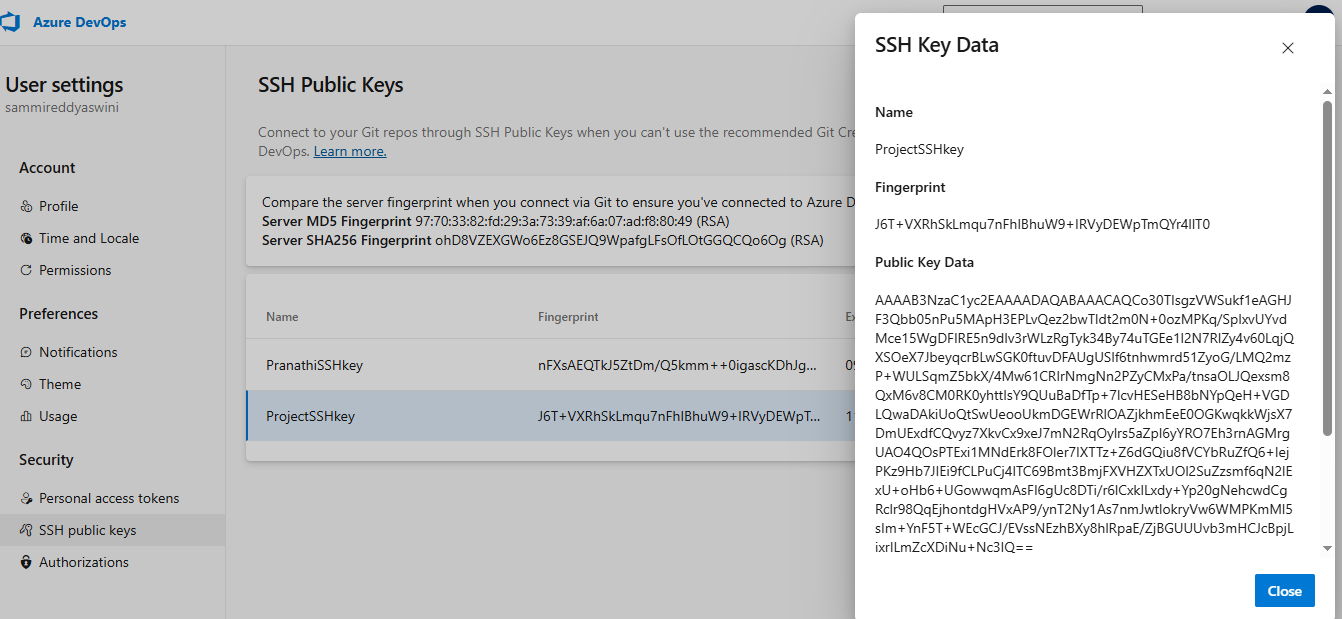
****

Cloning git

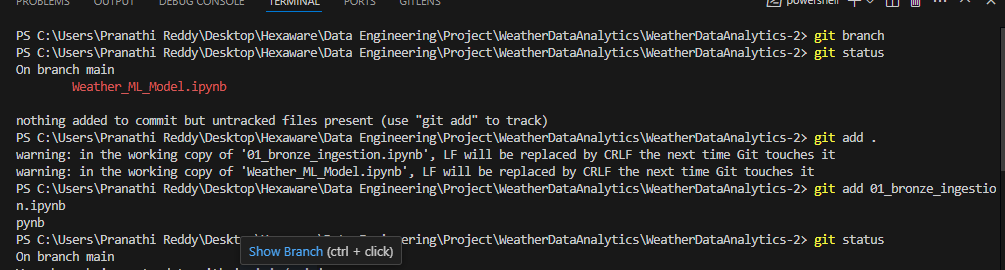


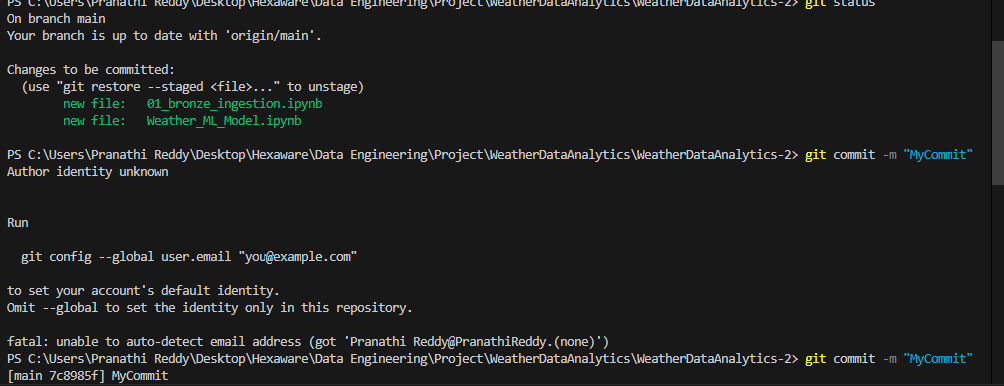


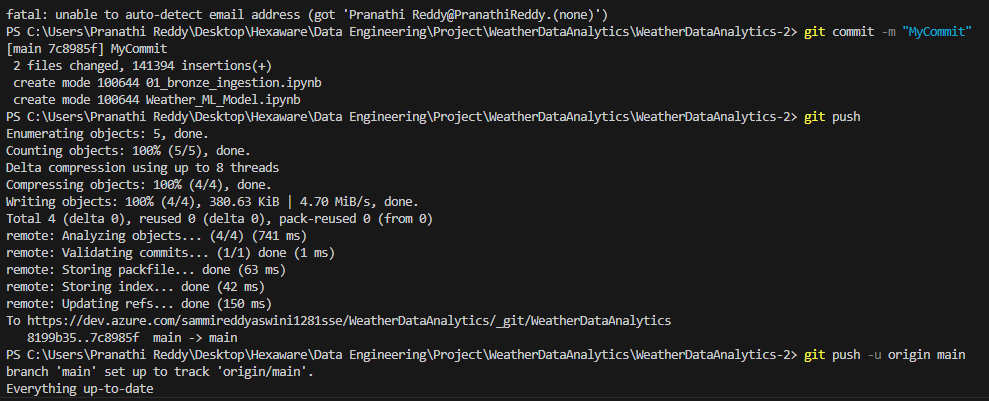
Update SSH Key



Add the notebooks and commit them







**9. Challenges Faced**

* Managing schema evolution while streaming
* Ensuring correct unpivoting and joining of multiple wide-format datasets
* Configuring SSH keys for Git in Azure DevOps
* Tuning ML model hyperparameters for optimal RMSE
* Databricks CLI permission issues

**10. Outcome & Learnings**

* Built a production-grade weather analytics pipeline
* Learned to use Delta Lake’s Bronze-Silver-Gold architecture
* Applied AutoLoader for real-time ingestion
* Understood ML model lifecycle from training to optimization
* Gained hands-on experience with DevOps and CI/CD pipelines

**12. Appendix:**

We can see in the commit in devops

