Here is a step-by-step breakdown of the NYC Taxi Data Engineering project from your uploaded document, focused strictly on the project details:

### **1. Project Overview**

* Objective: End-to-end data engineering project utilizing NYC taxi data.
* Tools and Technologies:
  + **Azure Data Factory (ADF):** For creating parameterized pipelines to pull data dynamically from APIs.
  + **Databricks (PySpark):** For data transformation.
  + **Delta Lake:** For data storage and advanced features like versioning and time travel.
  + **Azure Data Lake Storage:** For organizing data in a Medallion architecture (Bronze, Silver, Gold layers).
  + **Parquet Files:** Efficient column-based file format for big data.
  + **Power BI:** For connecting and visualizing data.

### **2. Data and Architecture**

#### **Data Source:**

* NYC Taxi Data is sourced from a public website, available in Parquet format, with data from 2009 to 2023.

#### **Key Features:**

1. **Dynamic API Integration:**
   * Automate fetching monthly data directly from the website using an API.
   * Avoid manual downloads and uploads.
2. **Medallion Architecture:**
   * **Bronze:** Raw data ingestion.
   * **Silver:** Data transformation and cleaning.
   * **Gold:** Finalized and modeled data for reporting.
3. **Delta Lake Features:**
   * Data versioning.
   * Time travel capabilities.
   * Delta logs for transaction consistency.

### **3. Prerequisites**

1. A stable laptop or PC.
2. Azure Free Account (or Pay-as-you-Go).
3. Access to NYC Taxi data API and lookup files.

### **4. Project Phases**

#### **Phase 1: Setting Up Data Ingestion**

1. **Create an Azure Free Account:**
   * Navigate to [Azure Free Account](https://azure.microsoft.com/free).
   * Set up a $200 free credit account for resource creation.
2. **Create an Azure Resource Group:**
   * Define a resource group to manage project components.
3. **Set Up Azure Data Lake:**
   * Create a **Storage Account** and enable **Hierarchical Namespace** to convert it into a Data Lake.
   * Set up three containers:
     + **Bronze:** Raw data.
     + **Silver:** Cleaned data.
     + **Gold:** Processed data.
4. **Manually Upload Lookup Files:**
   * Create directories under Bronze (e.g., trip\_type, trip\_zone).
   * Upload small CSV files for zone lookup data.
5. **Create Azure Data Factory (ADF):**
   * Set up a cloud ETL service for building pipelines.

#### **Phase 2: Building Dynamic Pipelines**

1. **Understand Linked Services:**
   * Create connections to:
     + The NYC Taxi data API (Source).
     + Azure Data Lake (Destination).
2. **Create Dynamic Pipeline in ADF:**
   * Set up **Copy Data Activity**:
     + Define **Source Dataset** using NYC API.
     + Define **Sink Dataset** pointing to Azure Data Lake Bronze container.
3. **Parameterization for API Integration:**
   * Replace static month values in the API URL with a parameter.
   * Dynamically update the month using a loop (ForEach activity).
4. **Validation:**
   * Test the pipeline with a single month’s data.
   * Validate output files in the Azure Data Lake.

#### **Phase 3: Data Transformation in Databricks**

1. **Set Up Databricks Cluster:**
   * Integrate Databricks for processing large-scale data.
   * Enable PySpark for transformation.
2. **Bronze to Silver Transformation:**
   * Clean raw data from the Bronze layer.
   * Perform data enrichment and remove duplicates.
3. **Silver to Gold Transformation:**
   * Aggregate data into a reporting-friendly structure.
   * Save finalized data in Delta tables.

#### **Phase 4: Reporting and Visualization**

1. **Connect Power BI:**
   * Set up a direct connection to the Gold layer.
   * Create dashboards and visualizations for insights.
2. **Key Outputs:**
   * Monthly, zone-wise taxi trip analytics.
   * Aggregated reporting on trip data.

### **5. Key Learnings and Benefits**

1. Practical exposure to industry-relevant tools and techniques.
2. Hands-on understanding of dynamic pipeline creation.
3. Working knowledge of Delta Lake and Medallion Architecture.
4. Efficient use of big data formats like Parquet.

Would you like me to expand on any specific section or generate a tutorial for any phase?

4o

##### **You said:**

Yes expand all the topics according to content given in the pdf

1/2

###### **ChatGPT said:**

ChatGPT

Here’s an expanded step-by-step guide for the **NYC Taxi Data Engineering Project**, based on the detailed content of your document. Each phase is elaborated for clarity while staying focused on project-specific instructions:

## **1. Project Overview**

This end-to-end project focuses on the **NYC Taxi dataset** and aims to provide hands-on learning of in-demand tools and technologies. The project builds real-time, industry-level data pipelines and uses advanced processing techniques.

### **Key Features**

* **Dynamic Data Ingestion:** Automate data fetching directly from APIs using Azure Data Factory.
* **Transformation with Databricks (PySpark):** Use advanced big data frameworks for cleaning and transformation.
* **Delta Lake Storage:** Utilize features like data versioning, time travel, and transaction logs.
* **Medallion Architecture:** Follow industry-standard practices for organizing data.
* **Column-Based Data Formats (Parquet):** Optimize storage and processing for big data.

## **2. Data and Architecture**

### **Data Source**

* NYC Taxi Data API provides monthly trip data in **Parquet format**. Additional **CSV lookup files** include zone details.
* The dataset spans multiple years, including data from **2009 to 2023**. This project uses the **2023 Green Taxi Trip Data** for demonstration.

### **Data Architecture**

The project follows a **Medallion Architecture**:

1. **Bronze Layer:** Raw data ingestion.
2. **Silver Layer:** Cleaned and enriched data.
3. **Gold Layer:** Aggregated and modeled data for analysis and reporting.

## **3. Prerequisites**

### **Hardware and Tools**

* A computer with stable internet.
* Azure Free Account (includes $200 free credit for resource creation).

### **Software Requirements**

1. **Azure Portal Access:**
   * Set up services like Azure Data Factory and Azure Data Lake.
2. **Databricks Account:**
   * For big data transformations using PySpark.
3. **Power BI (Optional):**
   * To visualize the Gold-layer data.

## **4. Step-by-Step Process**

### **Phase 1: Setting Up Azure Environment**

#### **Step 1.1: Create Azure Free Account**

1. Go to [Azure Free Account](https://azure.microsoft.com/free).
2. Set up a free trial account with $200 credits.
3. Configure billing and basic details.

#### **Step 1.2: Create Resource Group**

1. Log into the Azure Portal.
2. Search for **Resource Groups** and click **Create**.
3. Name your resource group (e.g., NYC-Taxi-Project-RG).
4. Select a region and finalize creation.

#### **Step 1.3: Configure Azure Data Lake**

1. Go to the Azure Marketplace and create a **Storage Account**.
2. Enable **Hierarchical Namespace** to convert the storage into a **Data Lake**.
3. Create three containers:
   * **Bronze:** For raw data.
   * **Silver:** For cleaned data.
   * **Gold:** For processed data.

#### **Step 1.4: Upload Lookup Files**

1. In the Bronze container, create two directories:
   * **trip\_type**: For trip type data.
   * **trip\_zone**: For zone data.
2. Upload the corresponding CSV files manually.

### **Phase 2: Dynamic Data Ingestion with Azure Data Factory**

#### **Step 2.1: Create Azure Data Factory**

1. Go to the Azure Portal, search for **Data Factories**, and click **Create**.
2. Name your Data Factory (e.g., NYC-Taxi-ADF).
3. Use the same resource group and region.
4. Finalize creation and launch the ADF Studio.

#### **Step 2.2: Establish Linked Services**

1. **Create API Connection (Source):**
   * In ADF, go to the **Manage Tab** > **Linked Services** > **New**.
   * Select **HTTP** as the connection type.
   * Provide the **Base URL** of the NYC Taxi API (e.g., https://example.net/).
2. **Create Data Lake Connection (Destination):**
   * Select **Azure Data Lake Gen2** as the connection type.
   * Choose the corresponding storage account.
   * Test and validate both connections.

#### **Step 2.3: Build Copy Data Pipeline**

1. Go to the **Author Tab** in ADF Studio.
2. Create a new pipeline and name it (e.g., NYC-Web-To-DataLake).
3. Add a **Copy Data Activity**:
   * **Source Dataset:** NYC Taxi API data (Parquet format).
   * **Sink Dataset:** Azure Data Lake Bronze container.

#### **Step 2.4: Parameterize the Pipeline**

1. **Add Parameters for Dynamic Data Fetching:**
   * Define a parameter (e.g., P\_Month) for the month in the API URL.
   * Replace the static month value in the URL with this parameter.
2. **Configure ForEach Activity:**
   * Use a **ForEach Activity** to loop through months (01 to 12).
   * Dynamically update the P\_Month parameter with each iteration.

#### **Step 2.5: Validate and Run**

1. Test the pipeline by debugging a single month.
2. Check output in the Bronze container.
3. Execute the pipeline to fetch data for all months dynamically.

### **Phase 3: Data Transformation in Databricks**

#### **Step 3.1: Set Up Databricks**

1. Launch a Databricks workspace and create a cluster.
2. Import the raw data from the Bronze container.

#### **Step 3.2: Bronze to Silver Transformation**

1. Use PySpark scripts to:
   * Remove duplicates and handle missing data.
   * Standardize data formats (e.g., date-time).
   * Save the transformed data to the **Silver Layer**.

#### **Step 3.3: Silver to Gold Transformation**

1. Perform advanced transformations:
   * Aggregations (e.g., trip counts by zone).
   * Calculations (e.g., average trip fare per zone).
2. Save the modeled data in **Delta Tables** in the Gold layer.

### **Phase 4: Reporting and Visualization**

#### **Step 4.1: Connect Power BI**

1. Establish a connection between Power BI and the Azure Data Lake.
2. Import data from the Gold layer.

#### **Step 4.2: Build Dashboards**

1. Create visualizations such as:
   * Zone-wise trip distribution.
   * Monthly revenue analysis.
2. Share insights with stakeholders.

## **5. Key Concepts Covered**

1. **Dynamic Pipelines:** Automating API data ingestion with parameters.
2. **Medallion Architecture:** Structuring data for efficiency.
3. **PySpark in Databricks:** Big data processing and transformations.
4. **Delta Lake:** Advanced storage features like time travel.
5. **Visualization with Power BI:** Presenting data insights.

Would you like further explanation on a specific phase or a sample script for any part of the project?

2/2

4o

##### **You said:**

I want a formal project report so that i can share it pn linkedin

###### **ChatGPT said:**

ChatGPT

Here’s a professional project report draft for your **NYC Taxi Data Engineering Project** that you can share on LinkedIn. This format highlights your work, skills, and tools effectively.

# **Project Report: NYC Taxi Data Engineering Project**

### **Abstract**

This project showcases an **end-to-end data engineering pipeline** built to process NYC Taxi trip data using modern data engineering tools and methodologies. The project follows the **Medallion Architecture**, leverages **cloud services like Azure**, and incorporates advanced features such as **Delta Lake** for data versioning and **dynamic API integration**. The ultimate goal is to prepare data for analytics and reporting in **Power BI**.

### **Objective**

To design and implement an automated data pipeline capable of fetching, processing, and storing large-scale data in an industry-standard format, while enabling data transformations and analytics-ready outputs.

### **Tools and Technologies Used**

1. **Azure Data Factory (ADF):** For dynamic pipeline creation.
2. **Azure Data Lake Storage Gen2:** For raw and processed data storage.
3. **Databricks (PySpark):** For data transformation and cleaning.
4. **Delta Lake:** For efficient data storage with features like time travel and versioning.
5. **Parquet Format:** For optimized big data storage and retrieval.
6. **Power BI:** For data visualization and reporting.

### **Dataset Overview**

* **Source:** NYC Taxi and Limousine Commission (TLC) Open Dataset.
* **Format:** Monthly Parquet files (Green Taxi Trip Records).
* **Years:** Focused on 2023 data for this project.
* **Additional Files:** Zone lookup files in CSV format.

### **Architecture Overview**

The project adopts a **Medallion Architecture**, dividing the data pipeline into three layers:

1. **Bronze Layer:** Raw data ingestion directly from APIs.
2. **Silver Layer:** Cleaned and transformed data.
3. **Gold Layer:** Aggregated and analytics-ready data.

Each layer serves a specific purpose, ensuring data quality and consistency as it flows through the pipeline.

### **Step-by-Step Implementation**

#### **Phase 1: Environment Setup**

1. **Azure Free Account:**
   * Created a free account with $200 credits for resource setup.
2. **Resource Group:**
   * Created a dedicated resource group to manage all project components.
3. **Azure Data Lake Setup:**
   * Created a storage account with **Hierarchical Namespace** enabled.
   * Configured containers for **Bronze**, **Silver**, and **Gold** layers.
4. **Manual Uploads:**
   * Uploaded zone lookup files in CSV format to the **Bronze Layer** for enrichment.

#### **Phase 2: Data Ingestion with Azure Data Factory (ADF)**

1. **Dynamic API Integration:**
   * Used Azure Data Factory to build a dynamic pipeline that fetches monthly Parquet files directly from the NYC Taxi API.
2. **Linked Services:**
   * Configured connections to:
     + The API (source).
     + Azure Data Lake (destination).
3. **Dynamic Pipeline Creation:**
   * Created a **Copy Data Activity** to move data from the API to the **Bronze Layer**.
   * Parameterized the pipeline using a **ForEach Activity** to loop through months dynamically.
4. **Validation:**
   * Verified data ingestion for all 12 months of 2023.

#### **Phase 3: Data Transformation in Databricks**

1. **Cluster Setup:**
   * Configured a Databricks cluster to process large-scale data.
2. **Bronze to Silver Transformation:**
   * Used PySpark to:
     + Remove duplicates.
     + Handle missing values.
     + Standardize data formats.
   * Saved the cleaned data to the **Silver Layer**.
3. **Silver to Gold Transformation:**
   * Aggregated data by zone and month.
   * Calculated metrics such as average fare and trip counts.
   * Stored the processed data in **Delta Tables** in the **Gold Layer**.

#### **Phase 4: Reporting and Visualization**

1. **Power BI Integration:**
   * Established a direct connection to the Gold Layer Delta tables.
2. **Dashboards:**
   * Created visualizations, including:
     + Monthly trip distribution by zone.
     + Revenue trends and averages per trip.

### **Key Features and Highlights**

1. **Dynamic Data Pipelines:** Automated API data ingestion using Azure Data Factory.
2. **Medallion Architecture:** Ensured structured and incremental data processing.
3. **Delta Lake:** Incorporated advanced features like time travel and versioning.
4. **Big Data Formats:** Utilized Parquet files for optimal performance.
5. **Advanced Visualization:** Delivered actionable insights through Power BI dashboards.

### **Challenges Faced and Solutions**

1. **Dynamic API Integration:**
   * Challenge: Manual data fetching was not scalable.
   * Solution: Parameterized pipelines with dynamic API URLs and ForEach Activity in ADF.
2. **Data Transformation at Scale:**
   * Challenge: Cleaning and transforming large datasets.
   * Solution: Leveraged PySpark on Databricks for distributed data processing.
3. **Data Storage Efficiency:**
   * Challenge: Managing storage for large-scale data.
   * Solution: Used Parquet and Delta Lake for optimized storage and retrieval.

### **Learning Outcomes**

1. Gained hands-on experience with cloud-native tools like Azure Data Factory and Databricks.
2. Mastered the Medallion Architecture for real-world data engineering projects.
3. Implemented dynamic pipelines to automate data ingestion tasks.
4. Learned advanced Delta Lake concepts like time travel and data versioning.

### **Project Repository**

The code and configurations for this project are available on GitHub:  
[GitHub Repository Link] *(Insert your GitHub link here)*

### **Future Enhancements**

1. Extend the pipeline to include real-time data ingestion.
2. Incorporate advanced machine learning models for predictive analytics.
3. Automate the dashboard updates in Power BI for live reporting.

### **Conclusion**

This project demonstrates the complete lifecycle of a data engineering pipeline, showcasing automation, scalability, and modern data engineering practices. It is a practical example of how to handle large-scale data using cutting-edge technologies, making it a valuable addition to any aspiring data engineer's portfolio.

Let me know if you'd like me to refine any section further or include more details!

4o

##### **You said:**

Make a linkedin post

###### **ChatGPT said:**

ChatGPT

Here’s a professional LinkedIn post draft for showcasing your **NYC Taxi Data Engineering Project**:

🚀 **Just Completed an End-to-End Data Engineering Project!** 🚀

I am thrilled to share my latest project: the **NYC Taxi Data Engineering Project**, where I built a complete pipeline to process and analyze large-scale data using industry-standard tools and architectures.

This project focused on dynamically fetching, transforming, and storing NYC Taxi trip data while following best practices for **big data engineering**.

### **🔑 Key Highlights:**

✅ Automated **data ingestion** from APIs using **Azure Data Factory**.  
✅ Organized data in a **Medallion Architecture** (Bronze, Silver, Gold).  
✅ Leveraged **Databricks (PySpark)** for scalable data transformations.  
✅ Utilized **Delta Lake** for data versioning, time travel, and performance.  
✅ Built **Power BI dashboards** for actionable insights.

### **⚙️ Tools and Technologies Used:**

💻 **Azure Data Factory**💻 **Databricks (PySpark)**💻 **Azure Data Lake Storage Gen2**💻 **Delta Lake**💻 **Parquet Format**💻 **Power BI**

### **🏆 Key Learnings:**

1️⃣ Built parameterized **dynamic pipelines** to fetch API data for an entire year in one automated workflow.  
2️⃣ Mastered **Delta Lake features** like time travel and transaction logs for big data management.  
3️⃣ Gained hands-on experience with **Medallion Architecture**, ensuring structured data processing.

### **🚀 Next Steps:**

🔄 Extend the pipeline for real-time data ingestion.  
📊 Enhance dashboards with predictive analytics.

I’m proud of this achievement and excited about what’s next in my data engineering journey.

👉 Check out my GitHub repository for this project: *(Insert GitHub link)*Let me know your thoughts or suggestions!

#DataEngineering #Azure #Databricks #PowerBI #DeltaLake #MedallionArchitecture #BigData

Feel free to personalize it further or let me know if you'd like additional refinements!

4o

##### **You said:**

How and what to add on github if i want to share this project to anyone by using github

###### **ChatGPT said:**

ChatGPT

To effectively share your NYC Taxi Data Engineering project on GitHub, follow these steps to create a professional and comprehensive repository:

### **1. Create a New Repository**

1. Go to [GitHub](https://github.com/).
2. Click **New** (top-right corner) to create a new repository.
3. Name your repository (e.g., NYC-Taxi-Data-Engineering).
4. Add a brief description (e.g., "End-to-end data engineering project for NYC Taxi dataset using Azure, Databricks, and Delta Lake").
5. Set the repository to **Public** if you want to share it widely.
6. Add a README file and a .gitignore file for Python or general use.

### **2. Organize Repository Structure**

Create a well-organized folder structure. Here's an example structure:

plaintext

Copy code

NYC-Taxi-Data-Engineering/

│

├── README.md # Project overview

├── assets/ # Any images or visuals (e.g., architecture diagrams)

├── data/ # Sample data files (non-sensitive)

│ ├── lookup/ # Lookup files (e.g., zone.csv)

│ └── output/ # Example output data (small-sized)

├── notebooks/ # Databricks notebooks or PySpark scripts

├── pipeline/ # ADF pipeline export files (JSON or ARM templates)

├── scripts/ # Supporting Python or Bash scripts

├── requirements.txt # Python dependencies, if applicable

└── LICENSE # License for your project

### **3. What to Include**

#### **a. README File**

The README file is the first thing visitors see. Include:

1. **Project Overview**:
   * Brief description of the project goals and tools used.
2. **Technologies**:
   * List of tools and technologies (e.g., Azure Data Factory, Databricks, Delta Lake).
3. **Architecture Diagram**:
   * Add an image of the Medallion Architecture or data flow (upload in the assets/ folder).
4. **Steps to Reproduce**:
   * Clear instructions for users to replicate your project.
5. **Usage Examples**:
   * Show outputs (e.g., screenshots of Power BI dashboards).
6. **GitHub Link**:
   * Add a link back to your LinkedIn post.

#### **b. Data Samples**

* Include **small sample files** from the dataset (e.g., one Parquet file and one CSV lookup file).
* Avoid uploading sensitive or large files to GitHub.

#### **c. Code and Scripts**

1. **Databricks Notebooks**:
   * Export your PySpark notebooks as .dbc or .ipynb files.
   * Add comments to explain key transformation steps.
2. **ADF Pipelines**:
   * Export your Azure Data Factory pipelines as ARM templates or JSON files.
     + Go to ADF Studio > Manage > Export ARM Template.
3. **Helper Scripts**:
   * Add Python scripts for any additional steps (e.g., preprocessing).
   * Include any shell commands or automation scripts.

#### **d. Documentation**

1. **Setup Instructions**:
   * Guide users on setting up Azure services (Data Factory, Data Lake, etc.).
   * Include commands for creating containers, uploading files, etc.
2. **Pipeline Details**:
   * Explain each pipeline step (e.g., dynamic ingestion, parameterization).
3. **Transformation Logic**:
   * Document how data is cleaned, enriched, and aggregated in Databricks.

#### **e. Visuals**

* Add diagrams for architecture and data flow (in assets/ folder).
* Example:
  + A Medallion Architecture diagram showing Bronze → Silver → Gold layers.
  + Sample Power BI dashboard screenshots.

#### **f. License**

* Add a license file (e.g., MIT License) to specify how others can use your code.

### **4. Example README Template**

markdown

Copy code

# NYC Taxi Data Engineering Project

## Overview

This project demonstrates an end-to-end data engineering solution for processing NYC Taxi data using modern tools like Azure Data Factory, Databricks, and Delta Lake.

## Architecture

![Medallion Architecture](./assets/medallion\_architecture.png)

## Tools and Technologies

- \*\*Azure Data Factory\*\*: For dynamic pipelines.

- \*\*Azure Data Lake Storage Gen2\*\*: For data storage.

- \*\*Databricks (PySpark)\*\*: For data transformation.

- \*\*Delta Lake\*\*: For time travel and versioning.

- \*\*Power BI\*\*: For visualizing results.

## Steps to Reproduce

1. Set up Azure Free Account.

2. Configure Azure Data Lake and Data Factory.

3. Import the provided ADF pipeline template and scripts.

4. Set up Databricks and run the notebooks for transformations.

5. Visualize the Gold layer in Power BI.

## Repository Contents

- `assets/`: Architecture diagrams and visuals.

- `data/`: Sample Parquet and CSV files.

- `notebooks/`: Databricks notebooks for data transformations.

- `pipeline/`: Exported JSON/ARM templates for ADF pipelines.

- `scripts/`: Python scripts for automation.

## Sample Dashboard

![Power BI Dashboard](./assets/powerbi\_dashboard.png)

## License

This project is licensed under the MIT License.

### **5. Push to GitHub**

Initialize a local Git repository:  
bash  
Copy code  
git init

git add .

git commit -m "Initial commit for NYC Taxi Data Engineering Project"

git branch -M main

git remote add origin https://github.com/<your-username>/<repo-name>.git

git push -u origin main

### **6. Share the Repository**

* Add the repository link to your LinkedIn post.
* Include a brief description in the repository about why the project is valuable.

Let me know if you need any help setting this up!