# Paris 2024 Olympics Data Analysis – End-to-End Azure Analytics Solution

## Project Overview

This report describes a comprehensive data platform project built around the **Paris 2024 Olympic Games** dataset. The project demonstrates how to orchestrate an end-to-end data pipeline using **Microsoft Azure** services to ingest, process, and visualize Olympic data in a scalable, automated manner. Rather than a personal account, the narrative is presented in a professional third-person style, focusing on the project's structure, technical implementation, and results. Key goals of the project included:

- **Seamless Cross-Service Integration:** Utilizing Azure Data Factory, Data Lake Storage Gen2, Databricks, Synapse Analytics, and Power BI in one unified pipeline.
- **Data Pipeline Orchestration & Automation:** Automating data ingestion, transformations, and loading processes with minimal manual intervention.
- **Resource Planning & Cross-Region Deployment:** Demonstrating deployment of services across regions (e.g., Data Factory in Southeast Asia and Databricks in Central Canada) to illustrate global scalability and planning for performance.
- **Insightful Analytics & Visualization:** Delivering interactive dashboards in Power BI that present key Olympic metrics for technical reviewers and business stakeholders.

The solution processes **synthetic and real datasets** related to the Paris Olympics. Approximately 90% of the data was synthetically generated (using tools like ChatGPT to create realistic dummy data for athletes, coaches, etc.), while the remainder incorporated real Olympic data sourced from Kaggle and GitHub for authenticity. This combination allowed the project to simulate the Paris 2024 Olympic data scenario with rich detail, even ahead of the actual event.

## Architecture and Tools

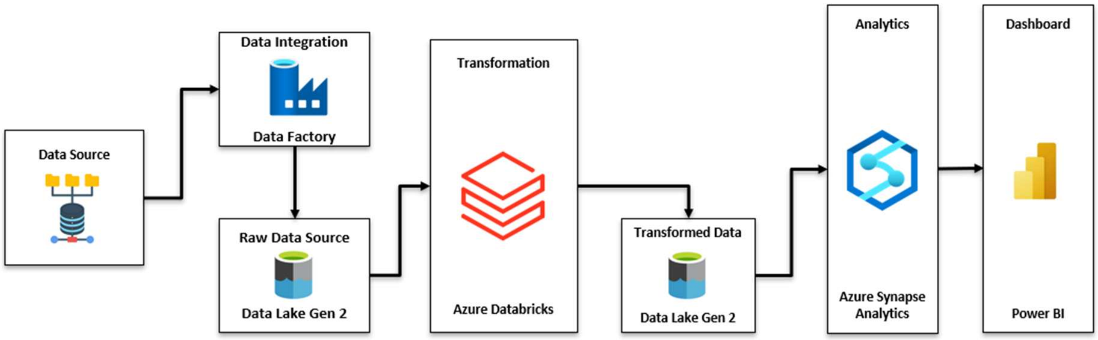


*Figure: End-to-end architecture of the Paris Olympics data pipeline, illustrating data flow from source to final dashboard.*

The solution follows a classic **lakehouse architecture**, with distinct data ingestion, storage, processing, analytics, and visualization phases. The figure above outlines the pipeline structure and the Azure services used at each stage. The key components and technologies in this architecture include:

- **Azure Data Factory (Data Integration):** A cloud-based data integration service orchestrating and automating data movement from source systems into the data platform.
- **Azure Data Lake Storage Gen2 (Raw & Curated Storage):** A scalable data lake that stores raw input data and transforms curated datasets. Data Lake Gen2 provides Hadoop-compatible storage, which is ideal for big data files.
- **Azure Databricks (Data Transformation):** An Apache Spark™ analytics platform (hosted on Azure) used for processing and transforming data at scale using Python and SQL. Databricks notebooks handle data cleaning, integration, and preparation.
- **Azure Synapse Analytics (Analytics & Aggregation):** Azure's unified analytics service is used here as a warehousing layer to aggregate and query the processed data. Synapse provides an SQL engine to combine datasets and prepare them for reporting, bridging the data lake and BI layer.
- **Power BI (Dashboard & Visualization):** Microsoft's business intelligence tool creates interactive dashboards and visualizations from the curated Olympic data, enabling end users to glean insights through charts and maps.

One notable aspect of the design is the **cross-region deployment** of services. In this project, the Azure Data Factory instance resides in Southeast Asia, while the Databricks workspace (and the primary Data Lake) are hosted in Central Canada. This distributed setup demonstrates Azure's flexibility in connecting services across regions. During planning, considerations were made for data latency and throughput between areas, and the pipeline was configured to ensure that data transfers and processing remained efficient despite the geographic separation. This approach can mirror real-world scenarios where data sources or compliance requirements necessitate multi-region architectures.

Overall, the architecture emphasizes modularity and scalability. Each service handles a specific stage of the pipeline, and they integrate seamlessly: Data Factory triggers and orchestrates processes across storage, Spark processing in Databricks, and loading to Synapse, culminating in visualization with Power BI. In the sections below, each pipeline phase is described in detail.

### Data Ingestion with Azure Data Factory

The project's first phase is **data ingestion**, handled by **Azure Data Factory (ADF)**. Data Factory pipelines were designed to automatically retrieve and load the source data (both synthetic and real) into the raw storage area of the data lake. The source data included multiple related datasets covering different aspects of the Olympics – for example, information on athletes, coaches, event entries by gender, medal tallies, and team data by country.

Azure Data Factory acts as the central **orchestrator** for these inputs. Pipelines in ADF are configured with a series of *Copy Data* activities, each tasked with moving one dataset from the source (in this case, files hosted on GitHub or generated and uploaded from ChatGPT outputs) into Azure Data Lake Storage. These Copy Data activities run sequentially to populate the Raw

Data zone with all required files. The orchestration ensures that datasets are ingested in the correct order and that any dependencies are respected (for instance, if the "Teams" data depends on the "Athletes" data, the pipeline can ensure athletes are loaded first).

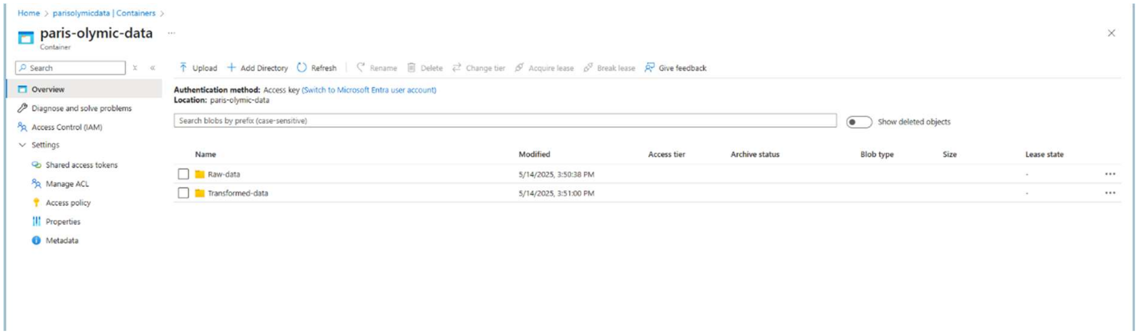
*Figure: Azure Data Factory pipeline orchestrating data ingestion for multiple Olympic datasets. Each block represents a Copy Data step for a specific dataset (e.g., Athletes, Coaches, Entries by Gender, Medals, Teams).*

The ADF screenshot above shows a pipeline with a series of **steps for copying data**. Each step successfully transfers a different Olympic dataset into the raw storage container (as indicated by the green checkmarks). This design showcases robust pipeline orchestration: ADF manages the sequence, monitors the success/failure of each step, and can be scheduled or triggered as needed (for example, on a schedule or demand). Parameters and reusable datasets were configured in Data Factory for flexibility, enabling the pipeline to be easily adjusted for new data sources or updated files in the future.

To ensure reliability and automation, the Data Factory pipeline was set up with **scheduled triggers** (for example, a daily run or a one-time trigger when new data is available). Logging and alerting in ADF helped identify and resolve any ingestion issues quickly. This means the ingestion process can run without manual intervention, a key aspect of *automation* in the project's data engineering practices.

**Raw Data Storage in Azure Data Lake Gen2**

Once ingested, all data lands in **Azure Data Lake Storage Gen2**, the **raw data repository**. The project's data lake is organized into separate containers (or folders) to keep raw data separate from processed data. In this case, a container (**Paris-Olympic-data**) holds two main directories: one for **Data** and another for **transformed data**. The raw zone contains the unmodified source files as ingested by Data Factory, preserving their original form (CSV files as obtained from Kaggle or generated synthetically). This allows the team always to reference the original data if needed and supports the reproducibility of transformations.



**Authentication method:** Access key (Switch to Microsoft Entra user account)
**Location:** paris-olymic-data

| Name | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|---|---|---|---|---|---|---|
| Raw-data | 5/14/2025, 3:50:38 PM | | | | | - |
| Transformed-data | 5/14/2025, 3:51:00 PM | | | | | - |

*Figure: Azure Data Lake Gen2 storage container showing the structured folders for raw data and transformed data (curated) for the Paris Olympics project.*

Storing raw data in the lake has multiple advantages. It provides a centralized, scalable repository that can handle large volumes (the athlete and event files contain thousands of records, for example). Azure Data Lake Gen2 is built on blob storage, meaning it can store semi-structured files efficiently and cost-effectively. For this project, the raw data included files such as **Athletes.csv**, **Coaches.csv**, **EntriesByGender.csv**, **Medals.csv**, and **Teams.csv** (naming conventions for illustration). These files were either sourced from public datasets (e.g., Kaggle's

Paris 2024 Olympics dataset for medals and entries) or synthetically generated for the project. Using a data lake, the project ensures that as new data becomes available (e.g., updated results or more participants' data closer to the Olympics), those files can be dropped into the raw folder and automatically picked up for processing in the next pipeline run.

Security and governance were also considered at this stage. Azure Data Lake Gen2 integrates with Azure's role-based access control (RBAC), so access to raw data was restricted only to the necessary services (Data Factory and Databricks) and personnel. This kind of planning highlights the project's focus on not just moving data but doing so in a controlled and secure manner – an important aspect of data management in enterprise scenarios.

**Data Transformation with Azure Databricks (Apache Spark)**

Raw data alone is not immediately useful for analytics – it often requires cleaning, integration, and aggregation. In this project, the transformation phase is carried out by **Azure Databricks**, which provides a managed Apache Spark environment for big data processing. Using Databricks notebooks, the team implemented data transformation logic in **PySpark (Python with Spark)** and SQL.

After the Data Factory finishes ingesting the raw data lake, a Databricks workflow (or an ADF-triggered Databricks notebook activity) takes over to process those files. The transformations included steps such as:

- Removing or correcting invalid entries (e.g., ensuring athlete and coach records have proper IDs and country codes),
- Standardizing formats (for example, country names or sports disciplines to have consistent naming),
- Integrating datasets (joining athletes with their coaches or teams, combining medal counts with country information, etc.) and
- Deriving new metrics (such as calculating the total number of participants by country or the male-to-female ratio in each sport).

The Databricks environment was configured in the Central Canada region, meaning the Spark cluster is geographically closer to the data storage (also presumed in Canada for low latency access). This is a conscious resource planning decision: keeping the computer near the data to optimize performance. The Spark cluster size and auto-scaling settings were planned based on data volume – for example, using a modest cluster that can scale out if the data grows (Spark's distributed computing ensures that even large datasets like thousands of Olympic records can be processed efficiently in parallel).

Throughout the transformation phase, an emphasis was placed on **automation and reproducibility**. Notebooks were parameterized so the same code could run in development or production with different file paths or dates. If integrated via Azure Data Factory, the pipeline could trigger the Databricks job automatically after data ingestion, ensuring a **smooth cross-service integration** – Data Factory passes control to Databricks to perform the heavy data processing. Once complete, control can pass back to continue the workflow (e.g., notifying Synapse or moving the processed data).

After processing, the cleaned and transformed data was written back to the **transformed data** area in the Azure Data Lake. This curated zone contains data that is analytics-ready: for instance, a refined **Athletes table**, a **Medals summary table**, etc., possibly stored in an organized folder structure or partitioned by categories (like a delta table or parquet files for efficient querying). By separating the transformed data from raw data, the project follows the best practices of a medallion architecture (with raw, curated, and possibly aggregated layers). The transformed data now serves as the input for the analytical phase in Synapse.

### Data Aggregation and Analytics with Azure Synapse

With cleaned data available, the next phase involves aggregating and preparing it for analysis using **Azure Synapse Analytics**. Synapse was chosen as the analytical layer or **data warehouse** for this project. There are a couple of ways Synapse can be used here: one approach is to utilize **Synapse's serverless SQL pools** to create external tables over the curated data in the lake, and another is to load the transformed data into a **dedicated SQL pool (data warehouse)** for high-performance querying. In this project, the team opted for a streamlined approach by leveraging Synapse serverless SQL to directly query the parquet/CSV files in the **Transformed** zone of the data lake. This saved time when loading data while enabling powerful SQL queries for reporting.

Views or aggregated tables were created within Synapse Analytics to support the Power BI reports. For example:

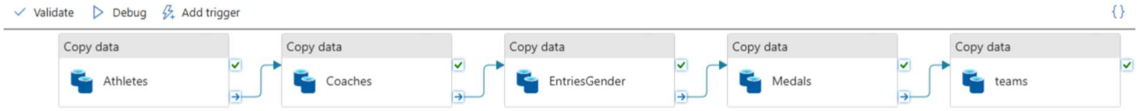
- A **Medal Tally** aggregation was built, summarizing total gold, silver, and bronze medals by country.
- A **Participation** summary was prepared, counting athletes by country, sport, and gender.
- A **Teams** summary table lists the number of teams each country has in each sport.
- Other dimensional data (such as a lookup for sports disciplines or countries) were made available to facilitate slicing the data in various ways.

By using Synapse, the project can take advantage of distributed query processing and the integration with Azure Data Lake. Synapse's analytics engine can read the transformed files quickly and join across different datasets (e.g., linking athletes to their country's medal count or merging team data with participation data). The result is that all business logic for reporting is encapsulated in SQL views or queries that Power BI will use. This approach highlights **cross-service integration** again: Azure Synapse reads data prepared by Databricks in the lake, demonstrating how Azure services can interoperate (Databricks writes data, and Synapse reads it without lots of data movement).

It is worth noting that resource planning was considered here as well. If using a dedicated SQL pool, one would size the data warehouse appropriately for the expected data volume and query complexity. In this serverless approach, the cost is based on the data scanned, so queries were optimized to scan only necessary data (partition pruning, projecting only needed columns, etc.). These optimizations ensure that the analytical layer is both *performant and cost-efficient* – an important project management consideration.

### Visualization and Reporting with Power BI

The final stage of the pipeline is delivering insights through **Power BI dashboards**. With curated and aggregated data available via Synapse (or directly from the lake through Synapse's endpoint), Power BI can connect and fetch the data to create interactive visualizations. The project's Power BI report was designed to be intuitive for business users while providing rich technical analysis detail. It includes multiple pages or sections focusing on different Olympic statistics.



**Key visualizations in the dashboard include:**

- **Participation by Gender and Sport:** A bar chart showing the count of participants (athletes) in each sport, broken down by gender. This highlights the gender distribution across different Olympic disciplines.
- **Total Medals by Country:** A world map visualization shading each country by the number of medals won, giving a global view of which countries are leading in the medal tally.
- **Distribution of Medals:** A horizontal bar chart summarizing the total counts of Gold, Silver, and Bronze awarded. This quickly shows the overall number of events won by medal type.
- **Number of Teams by Country:** A stacked column chart showing how many teams each country has sent, with segments colour-coded by sport. This helps illustrate the breadth of participation by country across various sports.

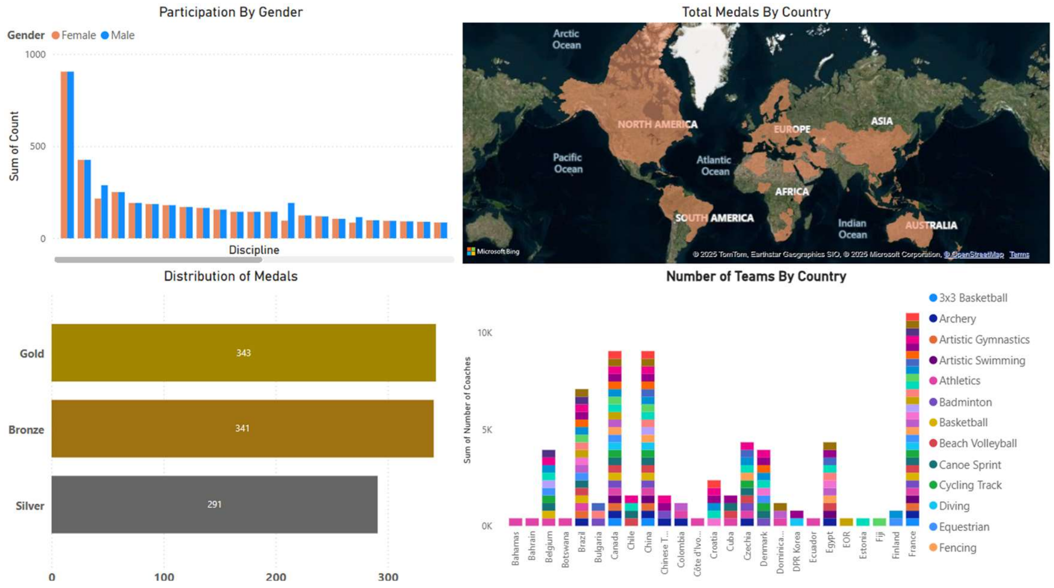


*Figure: Sample Power BI dashboard visuals from the project, showing key Olympic metrics. For example, the chart on the top-left breaks down athlete participation by gender for each sport, while the map (top-right) highlights the total medals by country. The bottom visuals include medal counts and team counts by country.*

The Power BI screenshot above showcases a snippet of the dashboard. In the top-left, the **Participation by Gender** chart reveals, for instance, that Athletics has one of the highest participants, with a roughly equal gender split, whereas smaller disciplines show varying male/female participation counts. The **Total Medals by Country** map on the top-right provides a quick geographic insight – countries like the USA, China, or host nation France might be prominently highlighted if they top the medal counts. The **Distribution of Medal's** bar chart (bottom-left) confirms how many events resulted in Gold, Silver, and Bronze medals – here, Gold medals might total 343, indicating 343 events or categories yielded a gold medal (with similar counts for Bronze and Silver). Finally, the **Number of Teams by Country** chart (bottom-right) is an overview of team representation: each country's bar is broken into coloured segments for each sport, showing not only how many teams they have in total but also how those teams are spread across sports (for example, the chart might show that the United States has a large number of teams across basketball, volleyball, athletics, etc., whereas a country like Botswana might have teams in fewer sports).

From a technical perspective, Power BI was connected to the Synapse Analytics SQL endpoint. This allowed using **Azure Active Directory single sign-on** and secure connectivity to the data. The data model in Power BI was built with relationships reflecting the Olympic data domain (athletes, teams, and medals as related tables). Measures and calculations (such as total medals or percentage of female athletes in a country's delegation) were created in Power BI's DAX language to enable interactive analysis. The report was then published to the Power BI service, where it can be shared with stakeholders. Scheduled refreshes were set up (for example, daily refresh) so that if new data is ingested and processed through the pipeline (e.g., as the Olympics progresses, daily results could update the medals table), the Power BI dashboard would automatically update, reflecting the latest information. This end-to-end automation — from Data Factory ingestion to Power BI refresh — exemplifies the project's success in delivering a live data solution.

**Project Highlights and Learnings**

This Paris Olympics data project underscores several important technical and project management skills:

- **End-to-End Orchestration:** The project successfully orchestrated a complex workflow across multiple services. Azure Data Factory acted as the central scheduler and controller, ensuring that each step (ingest, process, load, etc.) happened in the correct sequence and without manual intervention. The ability to chain tasks (like triggering a Databricks notebook after data copy or notifying Synapse of new data) showcases proficiency in building *automated data pipelines*.
- **Cross-Service Integration:** Integrating diverse Azure services was a key challenge and achievement. The pipeline demonstrates how Data Factory, Storage, Databricks, Synapse, and Power BI can work together. For instance, credentials and access had to be managed so Databricks could read/write to Data Lake, Synapse could query the lake, and Power BI could query Synapse – a web of connections configured securely and efficiently. This reflects an understanding of each service's APIs and connectivity (for example, using service principals or managed identities for authentication instead of personal credentials).

- **Resource Planning and Cross-Region Deployment:** Deciding to deploy Data Factory in a different region (Southeast Asia) from the data processing (Canada) was a deliberate choice to simulate a real-world scenario where data and orchestrator might not be co-located. The project involved planning for possible network latency and ensuring that large data transfers were still performant. It also required monitoring Azure costs (as cross-region data egress can incur charges) and adjusting the design as needed. Additionally, sizing the Databricks cluster and configuring Synapse's usage was part of resource planning – balancing cost with performance so that the solution remains fast and cost-effective.
- **Data Handling and Quality:** Using a mix of synthetic and real data requires careful management to maintain data quality. Synthetic data (90% generated via ChatGPT and other tools) was used to fill gaps where real data was unavailable (for example, generating plausible athlete and coach records). The project verified that this synthetic data conformed to expected formats and distributions. Real data from Kaggle (the remaining ~10%) provided a baseline of authenticity (e.g., real medal counts or actual list of sports/events). Combining these sources taught valuable lessons in data validation – ensuring that the merged dataset was coherent and realistic. Techniques like data profiling, verification of referential integrity (e.g., every team referenced a valid country and sport), and handling missing values were all part of the transformation notebooks.
- **Scalability and Future-Proofing:** The implemented pipeline is scalable. If tomorrow the dataset grows (say, more detailed Olympics data or data from previous Olympics for historical comparison), the infrastructure can handle it with minimal changes. Azure Databricks can scale to larger clusters, Synapse can be scaled or switched to a dedicated pool for heavier querying, and the data lake can store petabytes effortlessly. The modular design (each phase independent but connected through well-defined data contracts like file formats and table schemas) means that updates or maintenance in one area (e.g., switching data sources or adding a new data transformation) do not break the entire pipeline. This is an important aspect of project management – designing for change and growth.

Throughout the project, extensive testing was performed at each stage: pipeline debug runs in Data Factory, notebook tests in Databricks on sample data, query validations in Synapse, and verification of dashboard numbers against the source data. This ensured that the data was accurate and reliable by the time it reached Power BI. Such diligence reflects a professional approach to data engineering, where data correctness and system reliability are paramount.

### Conclusion

In summary, the Paris 2024 Olympics Data Pipeline project delivered a fully functional, cloud-based data platform to ingest raw data, transform it into meaningful information, and present insights through interactive dashboards. The use of Azure's modern data stack – spanning data factory, big data processing, cloud storage, analytics, and BI – showcases a holistic skill set in data engineering and analytics. Moreover, the project highlights how thoughtful **project structure** and **planning** (from cross-region architecture decisions to automation and scheduling) can result in a solution that is not only technically sound but also aligned with business needs (timely insights, accuracy, and the ability to answer key questions about the data).

This report focuses on the solution's features and outcomes by emphasising a professional narrative rather than personal involvement. It illustrates the kind of end-to-end solution that a data team might build in an enterprise context, albeit applied to the context of Olympic Games data. The result is a portfolio piece that is informative to technical reviewers (who can appreciate the use of Spark, SQL, and cloud services at scale) and accessible to business-oriented readers (who can see the value in the insights produced, like medal tallies and participation metrics).

*Detailed implementation artefacts (such as code snippets, pipeline JSON definitions, and configuration details) have been documented separately for interested readers or potential collaborators. These can be made available upon request (for example, via the author's LinkedIn*.