M01 Introduction to Big Data PROJECT

By

Rajasekhar - 1317710.
Priyatham - 1318815.
Sai Milind- 1320979.
Bangari Arjun- 1321198

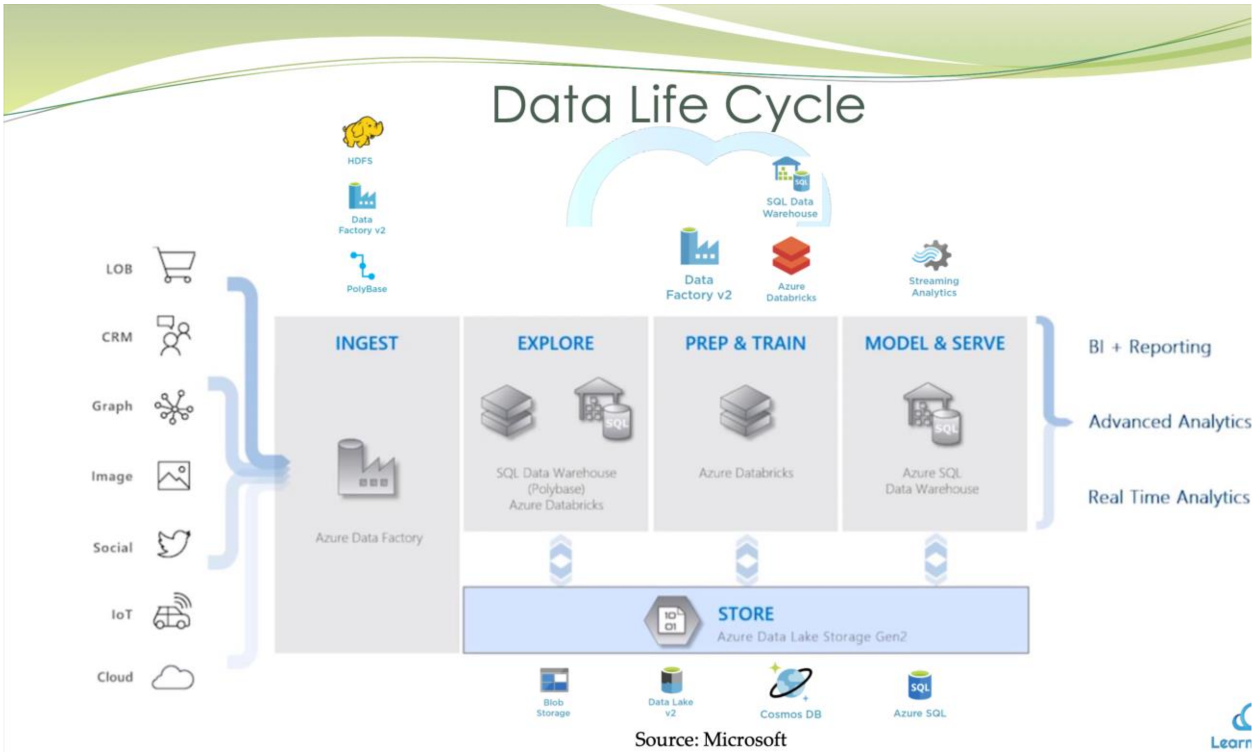
Data Engineering Life Cycle:

Input Data set to be considered from the sources:

https://www.kaggle.com/datasets/chaudharyanshul/airline-reviews

Data Engineering Life Cycle:

1. **Data Acquisition**: Gather data from diverse sources like databases, APIs, files, and streaming sources.
2. **Data Storage:** Store the acquired data in suitable repositories like data lakes, warehouses, or databases.
3. **Data Processing**: Clean, transform, and enrich the stored data to ensure its quality and usability.
4. **Data Integration:** Combine disparate datasets, resolving conflicts, to create a unified view for analysis.
5. **Data Analysis & Delivery:** Analyze processed data to extract insights and deliver actionable information for decision-making or product enhancements.



Source: Microsoft

To get started, here's a breakdown of steps you might consider for each of the core components:

1) Data Storage - Data Lake vs. Data Warehouse:

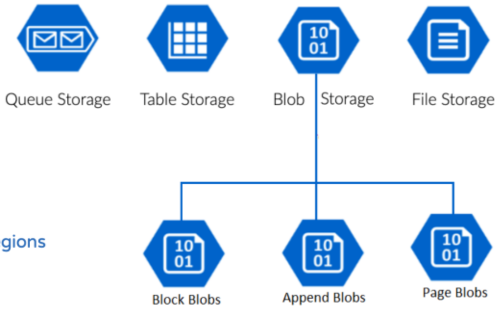
For your project, choosing between a Data Lake and a Data Warehouse in Azure depends on the nature of your data and its use cases.

1.1) Data Lake: Ideal for storing raw, unstructured, or semi-structured data. Use Azure Data Lake Storage Gen2, which combines the capabilities of a Data Lake with the security and scalability of Azure Blob Storage.

1.2) Data Warehouse (Azure Synapse Analytics): Suitable for structured data and analytics. It offers integration with various data sources and powerful querying capabilities. Choose based on your dataset's structure and the type of analysis you plan to perform.

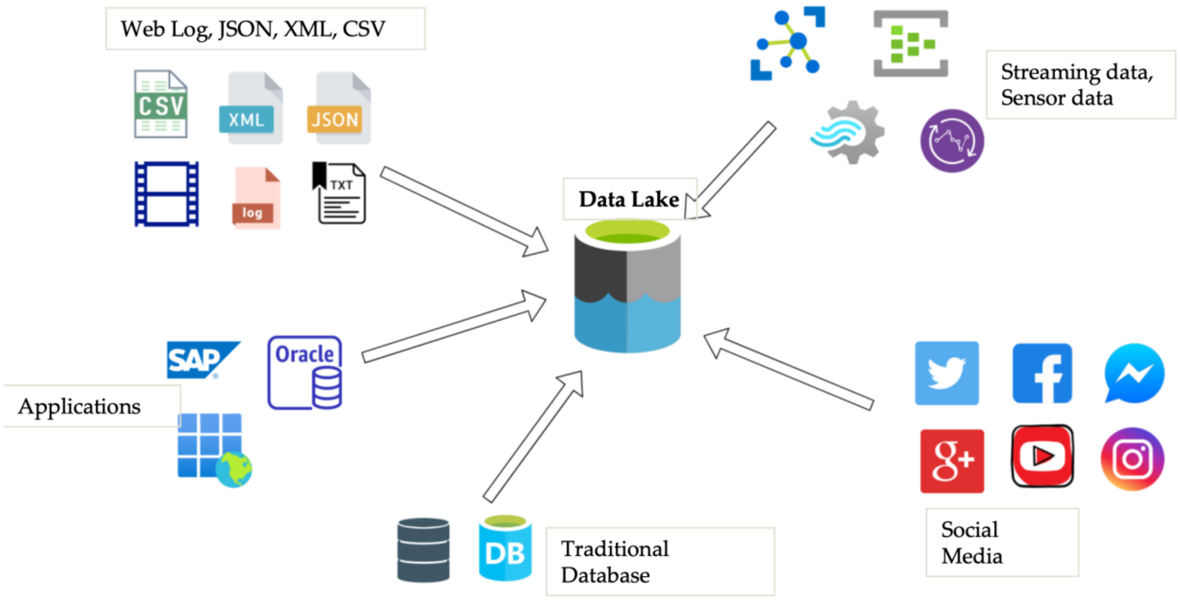
Azure Storage Service

- Diff types of data and requirements
  - Relational, non-relational/No-SQL, datasheets, images, videos, backups
  - Storage, access, security, availability, latency, processing, backup
- Diff types of Data Service
  - Azure Blobs: Text and binary data
  - Azure Files: Managed file shares (SMB Protocol)
  - Azure Queues: Messaging
  - Azure Tables: NoSQL store



- Features
  - Durable and highly available – redundancy across datacenters or regions
  - Secure – all data encrypted by default
  - Scalable – massively scalable
  - Managed - Azure handles hardware maintenance, updates, and critical issues for you.
  - Accessible - accessible from anywhere in the world over HTTP or HTTPS.
    - Clients libraries are available in all languages
    - Support scripting in PowerShell or Azure CLI

Data Lake Sources



What is Data Lake?

"If you think of a DataMart as a store of bottled water – clean and packaged and structured for easy consumption – the data lake is a large body of water in a more natural state. The contents of the data lake stream in from a source to fill the lake, and various users of the lake can come to examine, dive in, or take samples."

Data Warehouse

Data Lake

In a Nutshell,

Selection Rationale: Consider using Azure Data Lake Storage Gen2 for a Data Lake approach. It's ideal for handling large volumes of diverse data. The hierarchical namespace and integration with Azure services make it suitable for big data analytics. Defend this choice based on the flexibility it offers in managing unstructured and semi-structured data.

2) Connecting to Azure Services:

Utilize Azure services like Azure Data Factory or Azure Databricks to connect your chosen storage solution to the cloud. These services allow for seamless integration, management, and processing of large datasets.

Azure Services: Utilize Azure Data Factory to orchestrate data movement and transformation tasks between your Azure Data Lake Storage Gen2 and other Azure services.
Integration: Explain the process of setting up Azure Data Factory pipelines to ingest, process, and store data from various sources into your Data Lake.

Azure Databricks: Set up an Azure Databricks workspace to run Apache Spark jobs.
Configuration: Detail the steps to configure and connect Azure Databricks to your Azure Data Lake Storage Gen2.
Spark Application: Develop and execute a Spark application using Azure Databricks, showcasing data processing, transformations, or analytics.

2.1) Azure Databricks:

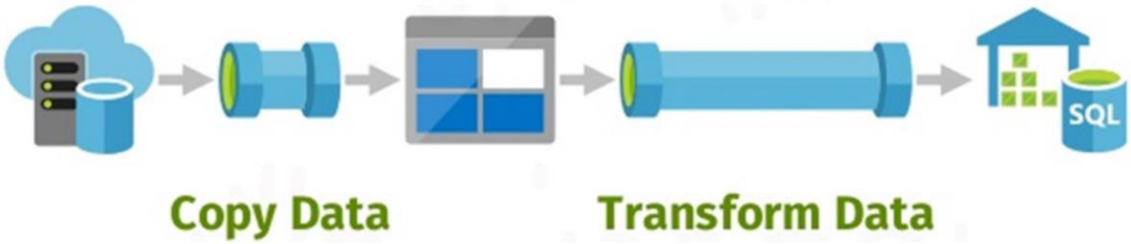
Description: Azure Databricks is an Apache Spark-based analytics platform optimized for Azure. It provides a collaborative environment for big data processing, machine learning, and data engineering tasks.

Features:
Unified Analytics Platform: Enables collaboration between data engineers, data scientists, and analysts in a single workspace.
Scalability: Offers scalable clusters for running Spark jobs, allowing adjustments in cluster size based on workload requirements.
Integration: Seamlessly integrates with other Azure services like Azure Data Lake Storage Gen2, Azure SQL Data Warehouse, Azure Cosmos DB, etc.
Usage:
Execute distributed Spark jobs for data processing, ETL (Extract, Transform, Load), machine learning, and real-time analytics.

2.2) Azure Data Factory:

Description: Azure Data Factory is a cloud-based data integration service that allows creating, scheduling, and managing data pipelines for moving and transforming data.
Features:
Data Orchestration: Orchestrates and automates data movement and transformation activities.
Connectivity: Connects to various data sources and destinations, including on-premises and cloud-based services.
Monitoring and Management: Provides monitoring dashboards and management tools for pipeline monitoring and optimization.
Usage:
Ingests data from diverse sources into Azure Data Lake Storage, Azure SQL Database, Azure Synapse Analytics, etc.
Orchestrates data movement and transformation tasks between different Azure services.

What can you do in Azure Data Factory?

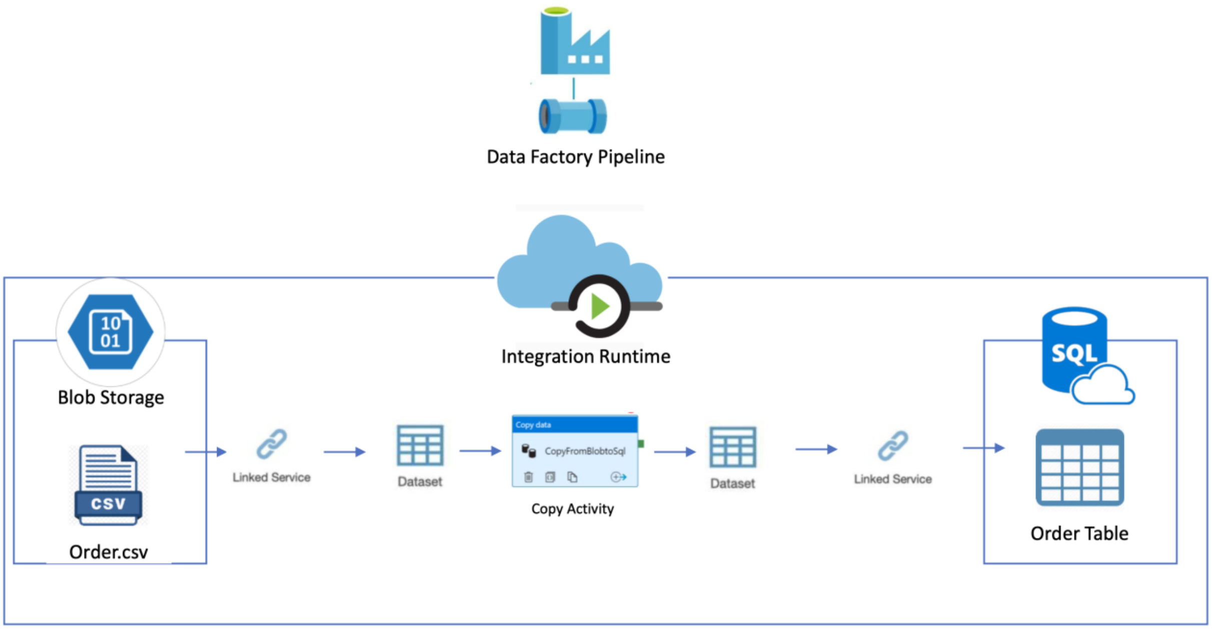


- **Copy Data**

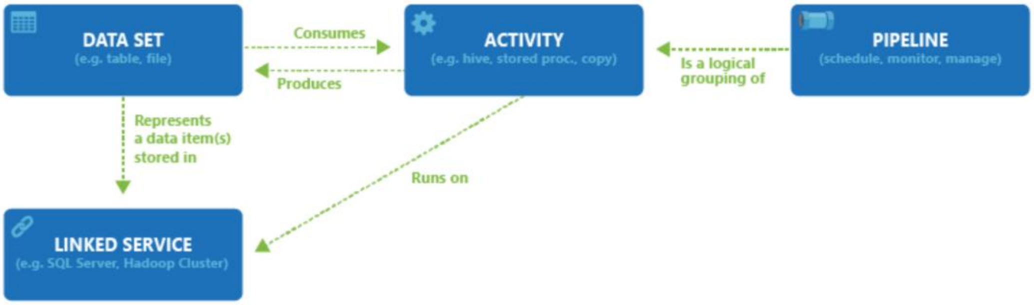
  More than 80 connectors to different services are available

- **Transform Data**

  Using newly added Data Flow, now Data Factory is complete cloud based ETL tool.



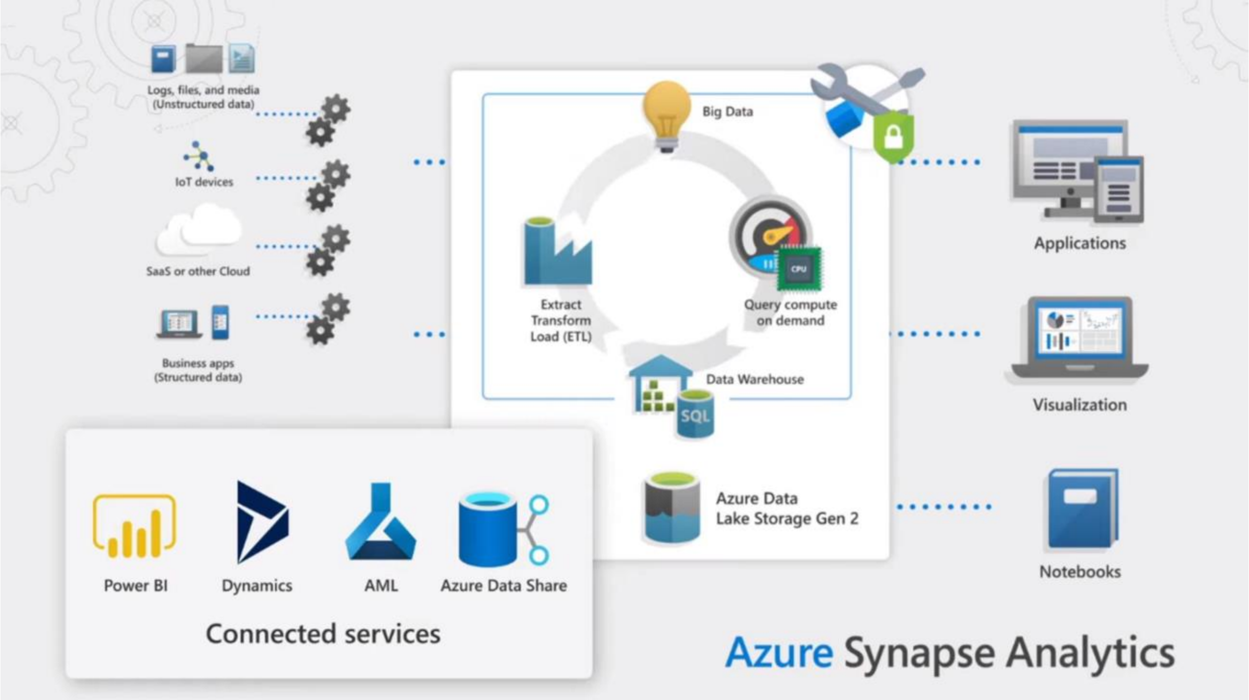
ADF Components


DATA SET
(e.g. table, file)
Consumes
Produces
ACTIVITY
(e.g. hive, stored proc., copy)
Is a logical grouping of
PIPELINE
(schedule, monitor, manage)
Represents a data item(s) stored in
Runs on
LINKED SERVICE
(e.g. SQL Server, Hadoop Cluster)

2.3) Azure Synapse Analytics (formerly Azure SQL Data Warehouse):

Description: Azure Synapse Analytics is an analytics service that brings together enterprise data warehousing and big data analytics. It enables querying and analyzing large volumes of data.

Features:
Massively Parallel Processing (MPP): Distributes processing across multiple nodes for fast querying.
Integration: Integrates with various data storage options like Azure Data Lake Storage, Azure Blob Storage, etc.
Built-in Analytics: Offers built-in capabilities for data exploration, machine learning, and business intelligence.
Usage:
Perform analytics on large datasets by querying and analyzing data stored in Azure Data Lake Storage or other Azure storage solutions.

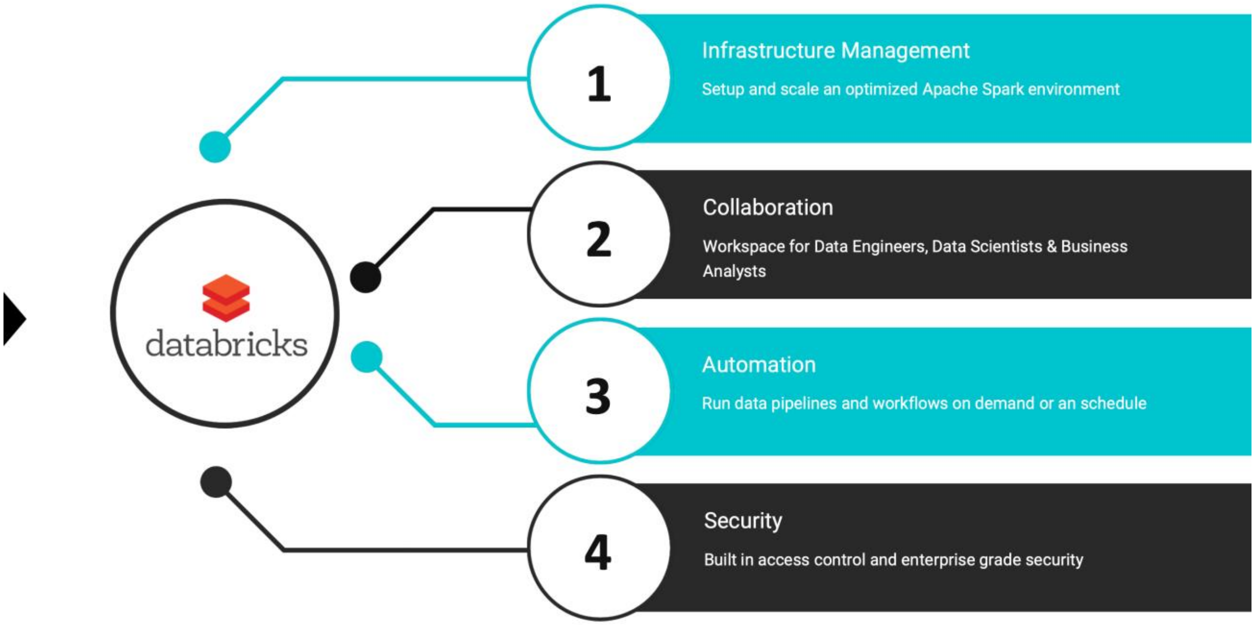


2.4) Azure HDInsight:

Description: Azure HDInsight is a fully managed cloud service for open-source analytics. It supports various open-source frameworks like Hadoop, Spark, Hive, HBase, etc.

Features:
Open-Source Support: Allows running open-source big data frameworks on Azure infrastructure.
Integration: Integrates with Azure storage and other Azure services for data processing and analytics.
Scalability and Flexibility: Offers scalability and flexibility in configuring clusters for specific analytics workloads.
Usage:
Run distributed analytics workloads using Hadoop, Spark, or other open-source frameworks.
These distributed cloud services in Azure provide a range of capabilities for handling big data workloads, from data ingestion to processing, analytics, and machine learning. The choice of service(s) depends on specific requirements, data types, and the nature of analytics tasks needed for your project.

3) Running Spark Application in Azure:



1 Infrastructure Management

Setup and scale an optimized Apache Spark environment

2 Collaboration

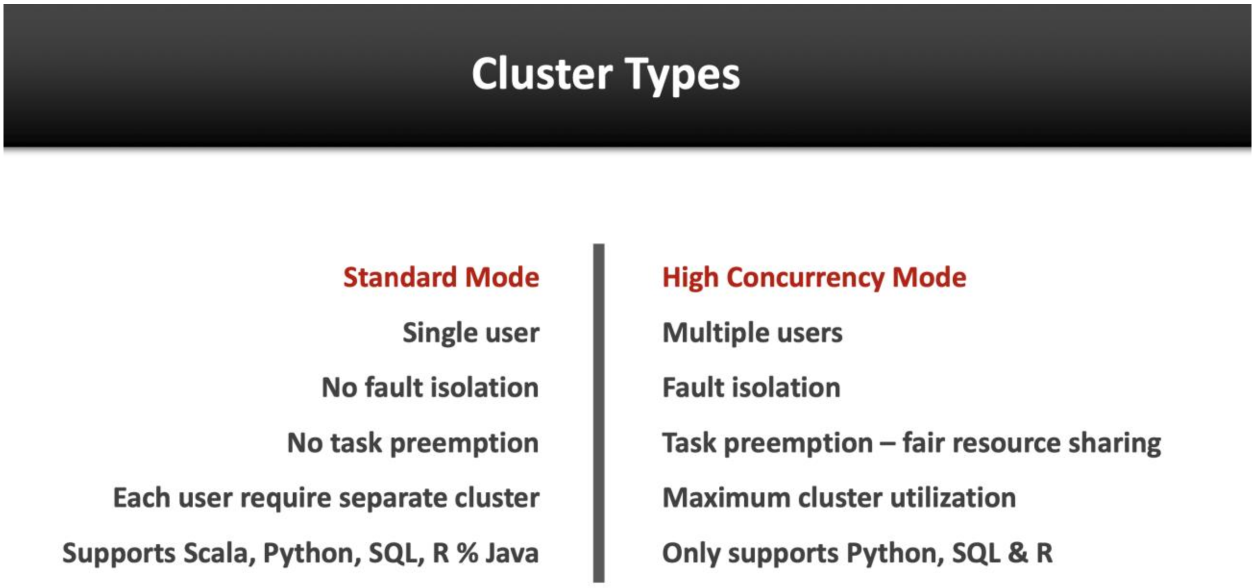
Workspace for Data Engineers, Data Scientists & Business Analysts

3 Automation

Run data pipelines and workflows on demand or an schedule

4 Security

Built in access control and enterprise grade security



Cluster Types

Standard Mode	High Concurrency Mode
Single user	Multiple users
No fault isolation	Fault isolation
No task preemption	Task preemption – fair resource sharing
Each user require separate cluster	Maximum cluster utilization
Supports Scala, Python, SQL, R % Java	Only supports Python, SQL & R

Azure Databricks provides a collaborative Apache Spark-based analytics platform. You can create a Spark cluster within Databricks, load your data, and perform distributed computing tasks.

Running a Spark application over distributed services in Azure typically involves setting up and utilizing Azure Databricks. Here's a detailed list of steps involved in running a Spark application using Azure Databricks in an Azure environment:

Setting Up Azure Databricks:

1. Create an Azure Databricks Workspace:

2. Log in to the Azure portal.

3. Search for and select "Azure Databricks" in the Marketplace.

4. Create a new Databricks workspace, configuring the necessary settings such as subscription, resource group, and workspace name.

5. Access Azure Databricks Workspace: Once the workspace is provisioned, access it through the Azure portal.

6. Create a Cluster: Inside the Azure Databricks workspace, create a new cluster specifying the Spark version, instance types, and configurations.

7. Connecting Azure Data Lake Storage Gen2 to Databricks: Link Azure Data Lake Storage Gen2:

8. Access the Azure Databricks workspace.

9.Navigate to "Data" -> "Add Data" -> "Data Lake Storage Gen2."

10.Configure the connection settings, providing the required credentials and access control details.

11.Running Spark Application:

12.Create a Notebook:

13.Inside the Databricks workspace, create a new notebook for your Spark application.

14.Load and Process Data:

15.Use Spark APIs (in Python, Scala, or SQL) within the notebook to load data from Azure Data Lake Storage Gen2 into Spark DataFrames.

16.Perform necessary transformations, data cleaning, or analytical operations using Spark functions.

17.Execute Spark Jobs:

18. Write and execute code within the notebook to run Spark jobs. This can involve data aggregations, machine learning algorithms, or any analysis relevant to your project.

19. Monitor and Optimize:

20. Monitor the Spark job's progress, performance, and resource utilization within the Azure Databricks environment.

21. Optimize Spark configurations if needed for better performance.

22. Save Results or Output:

23. Store Processed Data:

24. Save the processed or analyzed data back to Azure Data Lake Storage Gen2 or any other Azure data service for further use or reporting.

25. Documentation and Reporting:

26. Detailed Documentation:

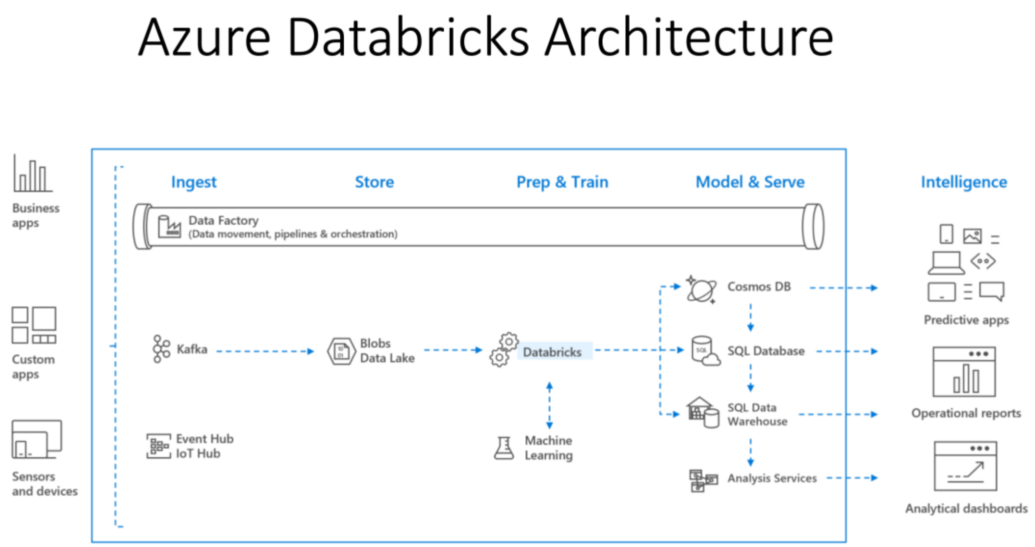
27.Document each step taken in the Azure Databricks notebook, providing comments and explanations for clarity.

28. Reporting:

29. Include details about the Spark application's purpose, dataset used, data processing steps, and the significance of the results in your final report.

30. Following these steps meticulously will allow you to effectively set up and run a Spark application using Azure Databricks over distributed services in Azure, leveraging the

power of Spark for big data processing and analytics.

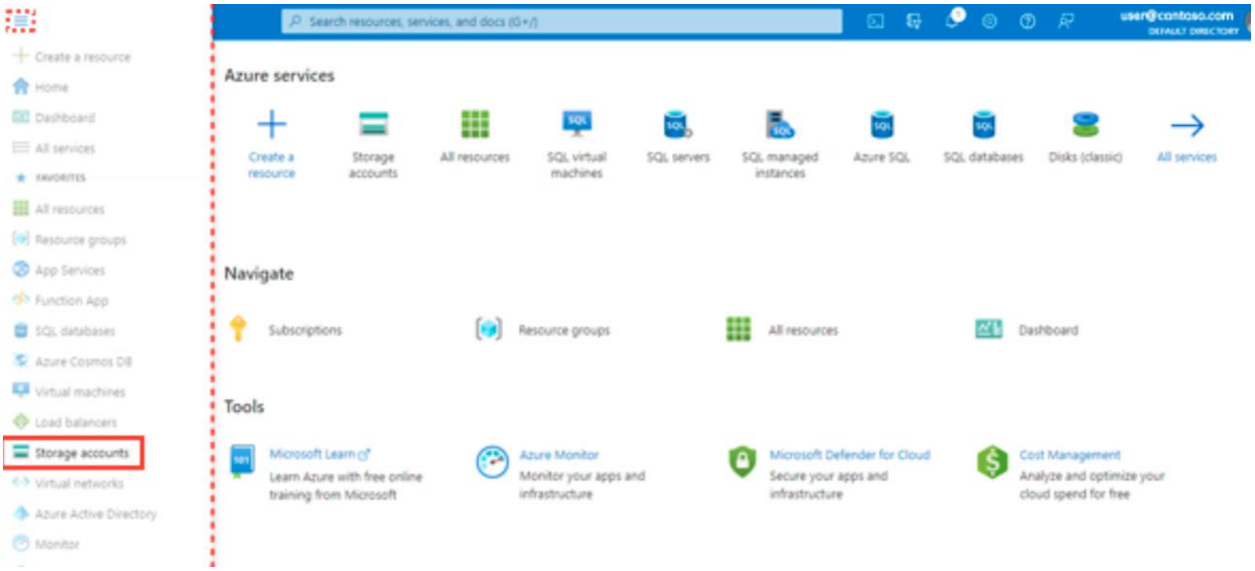


END – END Implementation Flow By Taking Screenshots in AZURE:

END – END DATA ENGINEERING FLOW:

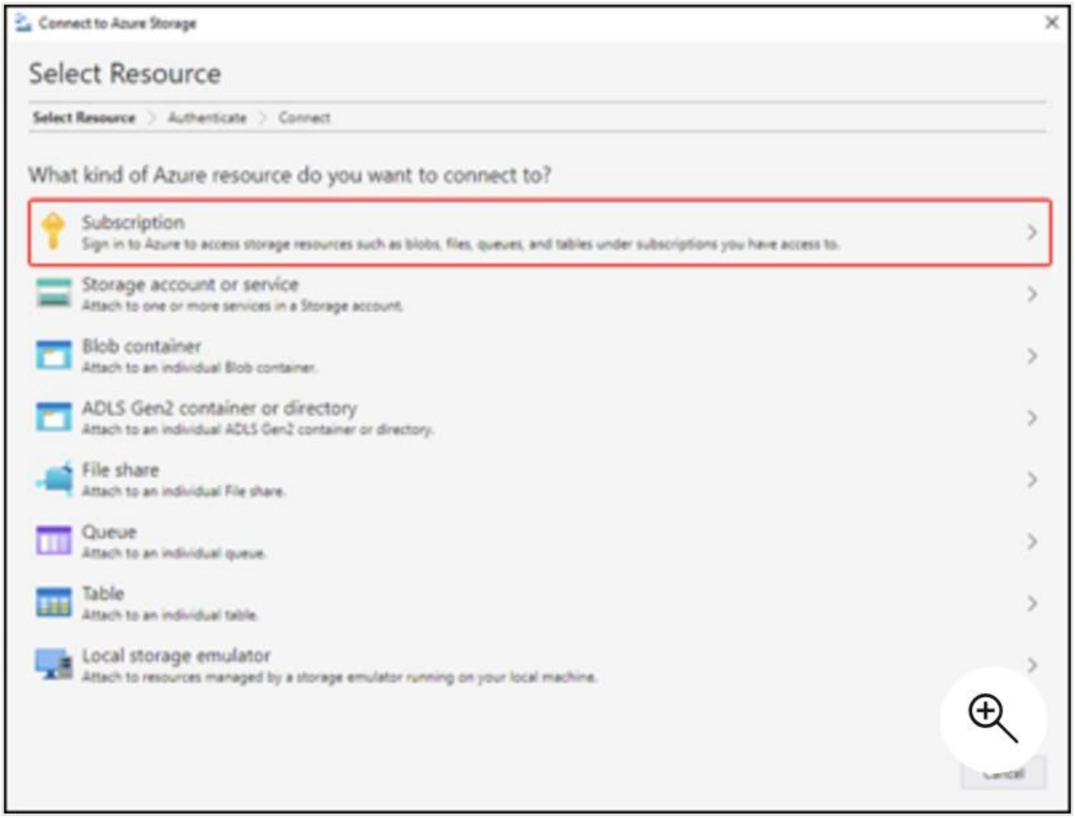
Azure Data Engineering project typically involves several steps from setting up storage to deploying data pipelines. Here's a comprehensive breakdown:

1. Storage Setup - Azure Data Lake Storage Gen2



Setting up Azure Data Lake Storage Gen2:

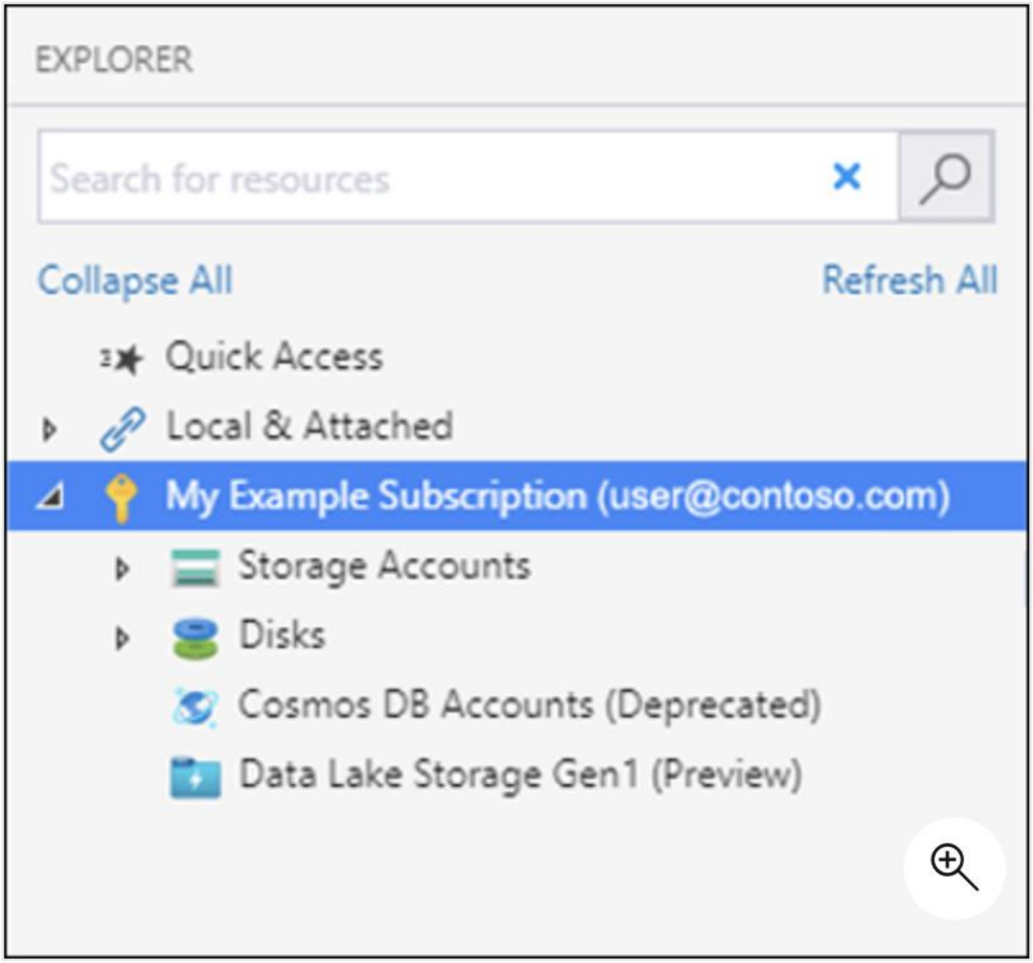
In the **Select Resource** panel, select **Subscription**.



Select Resource

What kind of Azure resource do you want to connect to?

- Subscription
  Sign in to Azure to access storage resources such as blobs, files, queues, and tables under subscriptions you have access to.
- Storage account or service
  Attach to one or more services in a Storage account.
- Blob container
  Attach to an individual Blob container.
- ADLS Gen2 container or directory
  Attach to an individual ADLS Gen2 container or directory.
- File share
  Attach to an individual File share.
- Queue
  Attach to an individual queue.
- Table
  Attach to an individual table.
- Local storage emulator
  Attach to resources managed by a storage emulator running on your local machine.


EXPLORER
Search for resources
Collapse All
Refresh All
Quick Access
Local & Attached
My Example Subscription (user@contoso.com)
Storage Accounts
Disks
Cosmos DB Accounts (Deprecated)
Data Lake Storage Gen1 (Preview)

1.1) Create an Azure Storage Account:
Log in to Azure portal and create a new Storage Account.
Choose the storage account type as "StorageV2 (general purpose v2)."
Enable hierarchical namespace to make it Gen2.

1.2) Set Access Control and Permissions:
Define access controls and permissions for data security.
Set up Shared Access Signatures (SAS) or Azure Active Directory (AAD) authentication for access control.

2. Data Ingestion - Azure Data Factory:

Using Azure Data Factory for Ingestion:

2.1) Create an Azure Data Factory (ADF) instance:
Navigate to the Azure portal and create an Azure Data Factory instance.

Configure linked services to connect Data Factory with Azure Data Lake Storage Gen2 and other data sources like databases, SaaS applications, etc.

2.2) Define Pipelines:
Create pipelines in ADF to ingest data from various sources into Azure Data Lake Storage Gen2.
Use Copy Data activities to move data between sources and storage.
Use Data Transform activities to perform certain transformations and actions.

3. Data Processing - Azure Databricks:

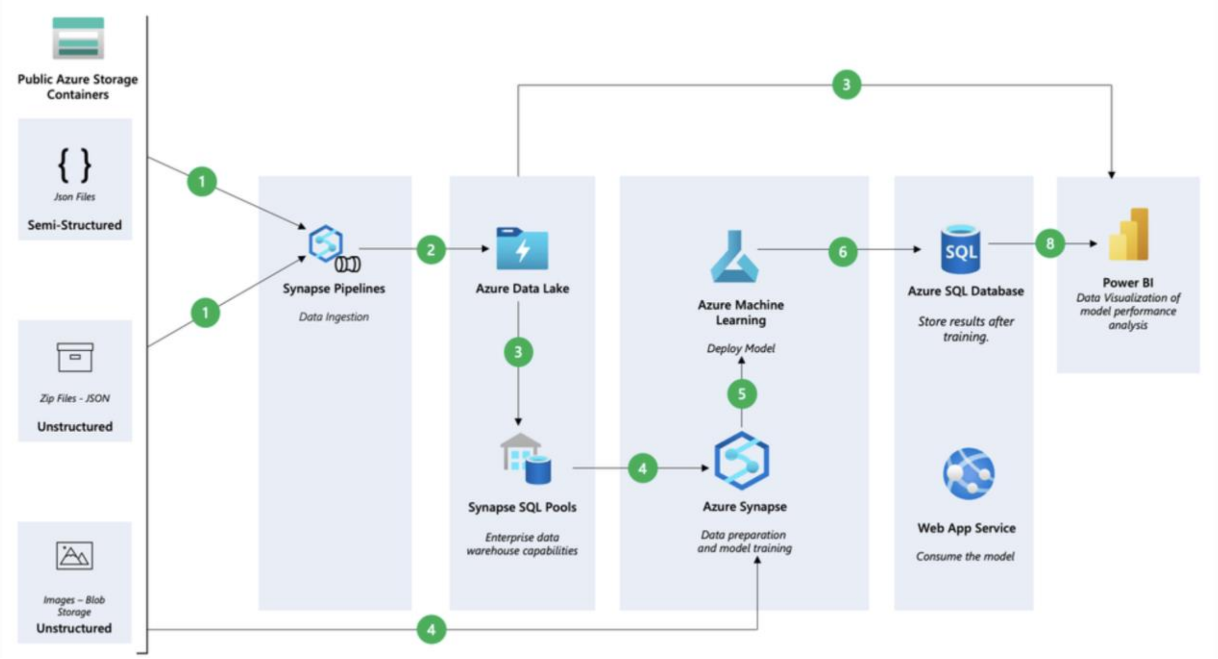
Leveraging Azure Databricks for Processing:

Set Up Azure Databricks Workspace:
Create an Azure Databricks workspace in the Azure portal.
Integrate with Azure Data Lake Storage Gen2:
Configure Databricks to connect with Azure Data Lake Storage Gen2.
Develop Data Processing Logic:
Create notebooks in Databricks to write code for data processing using Spark.
Perform data transformations, cleanups, aggregations, or machine learning tasks as needed.

4. Data Analysis and Preparation - Azure Synapse Analytics:

Utilizing Azure Synapse Analytics:

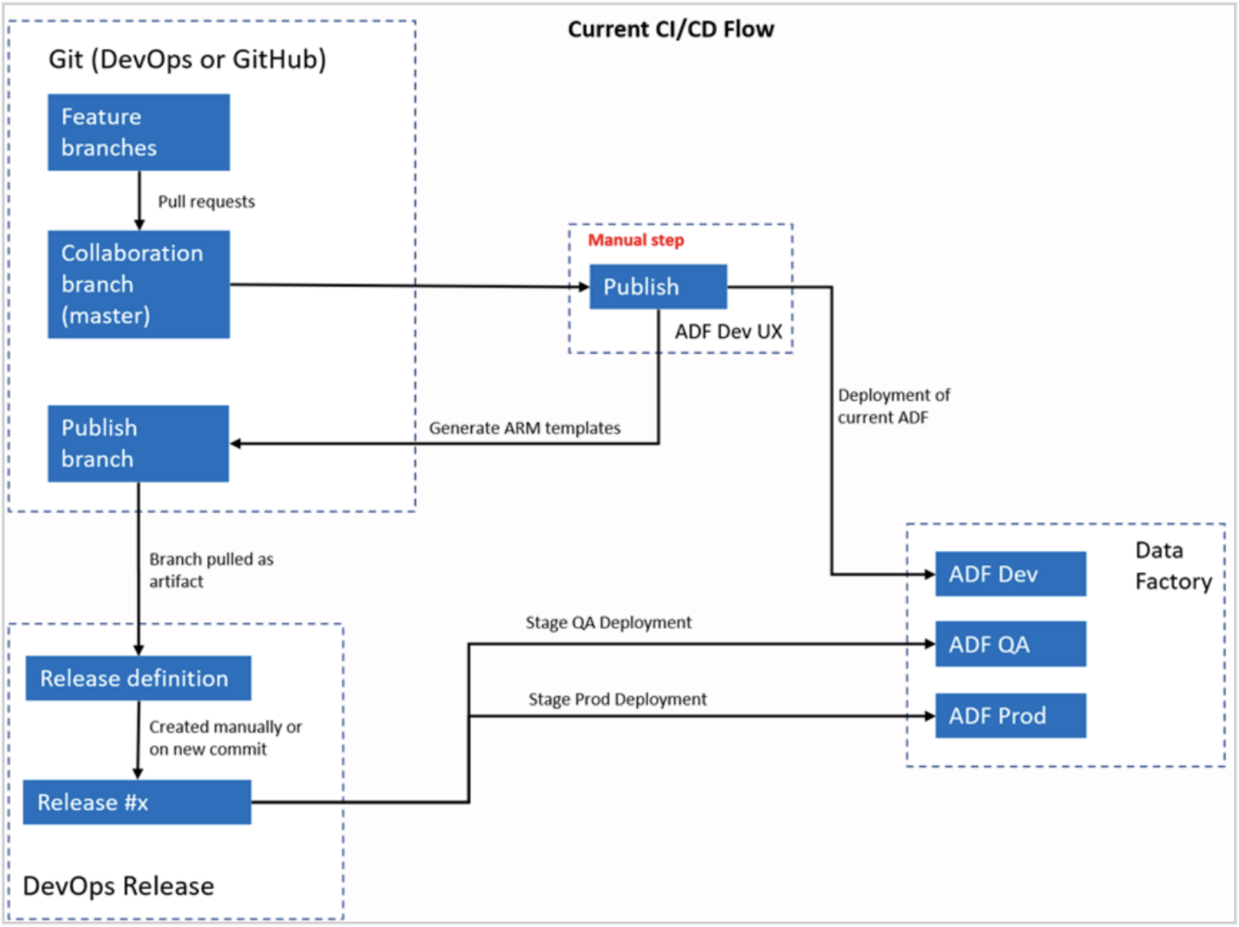
Create an Azure Synapse Analytics Workspace:
Set up an Azure Synapse Analytics Workspace in the Azure portal.
Data Exploration and Analysis:
Use Synapse Studio to explore and analyze data stored in Azure Data Lake Storage Gen2.
Write and execute T-SQL queries for data preparation and analysis.



5. Deploying Data Pipeline - Azure Data Factory

Deployment of Data Pipeline in Azure Data Factory:

Build Data Pipelines:
In Azure Data Factory, design and orchestrate data movement and transformation tasks.
Incorporate data processing logic developed in Azure Databricks and data analysis results from Azure Synapse Analytics.
Pipeline Deployment:
Deploy the created data pipelines in Azure Data Factory.
Schedule pipelines for regular execution or trigger them based on specific events.



6. Conclusion and Maintenance:

Documentation and Reporting:
Document each step and process followed in the project, detailing configurations, code snippets, and any troubleshooting encountered.
Create a final report outlining the project's objectives, methodology, results, and insights gained.

7. Monitor and Maintain:

Continuously monitor the pipelines' performance, data quality, and system health.
Iteratively optimize the pipelines for improved efficiency and performance.
This sequential approach covers setting up storage, ingesting data, processing it using Azure Databricks and Synapse Analytics, and deploying the final pipeline using Azure Data Factory in an Azure Data Engineering project. Each step contributes to the overall process of managing and analyzing data in the Azure environment.