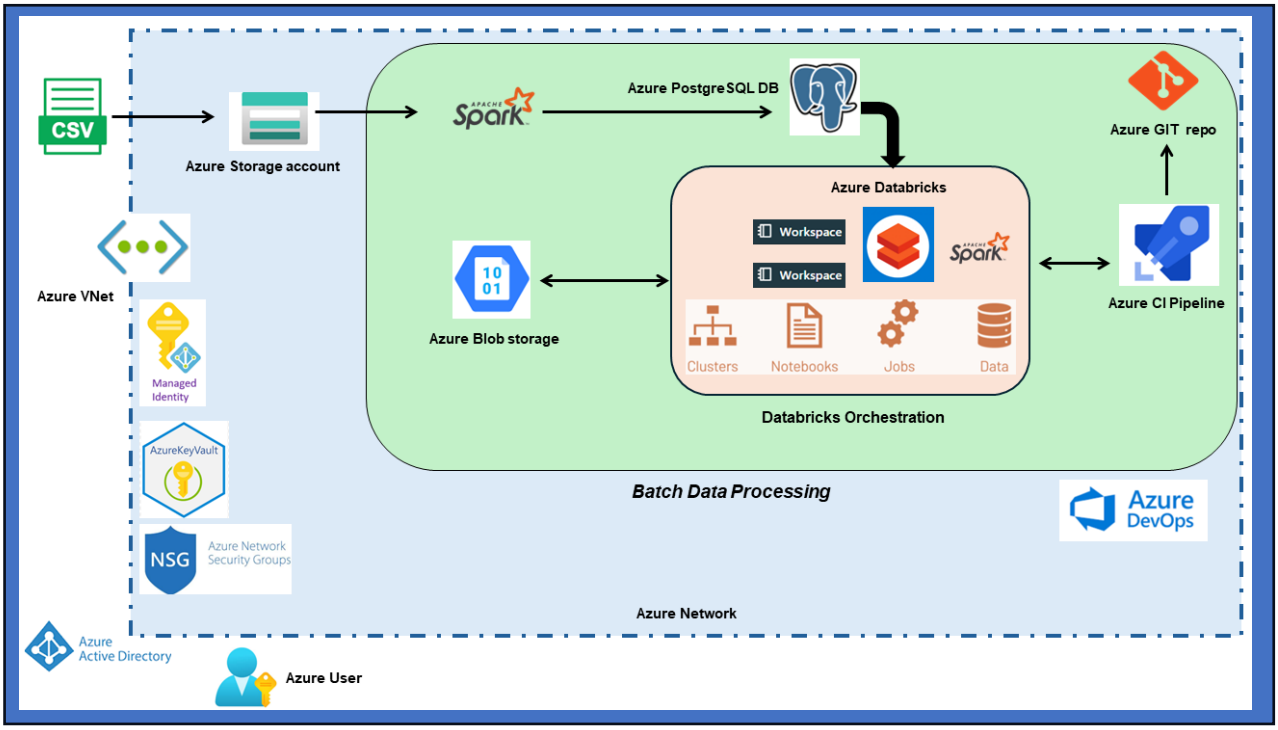
Project: Data Engineering - Development Phase

Build a batch-processing-based data architecture for a data-intensive application

Abstract:

The goal of this project is to use Azure services to design and build a reliable batch-processing data architecture that will be used as the backend of a machine learning application that processes enormous amounts of data. Creating a scalable, dependable, and maintainable system that can effectively process substantial amounts of data, store it, conduct the necessary preprocessing, and aggregate the data for use in machine learning applications is the main goal. The primary goal of this project is to construct the required data infrastructure by utilizing Azure DevOps, Azure Databricks, Azure Key Vault, Azure Blob Storage, Azure PostgreSQL, and Azure Network security. At the conceptual stage, we will focus on integrating canonical software components and frameworks and adapting common data engineering principles to create a state-of-the-art architecture for data processing. The design will prioritize batch scheduling of data processing tasks to coincide with the frontend application's quarterly execution cycle, which oversees producing new iterations of the machine learning model. Wilder, B. (2012).

Figure 1: Batch data processing Architecture



Source: Own representation.

1. Project development overview:

1.1 Tools used for this entire project:

- Azure DevOps
- Azure Repos
- Azure Pipelines
- Azure Portal
- Azure Database for PostgreSQL flexible server
- Azure Databricks Service
- Resource Group
- Network Watcher
- Azure Blob Storage account
- Network security group
- Azure Key Vault

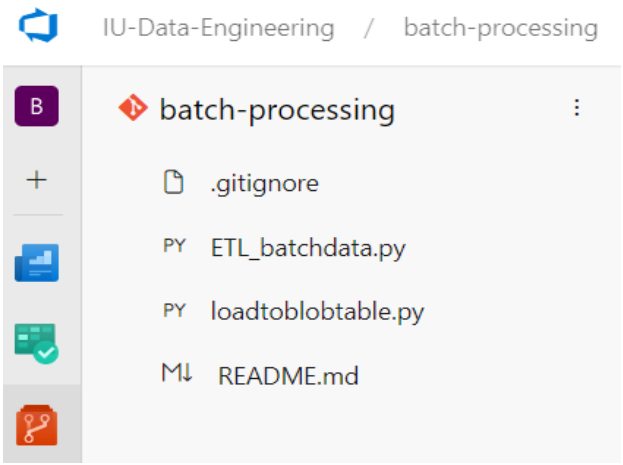
1.2 Subscriptions required for the project

- Azure subscription
- Azure DevOps account
- GitHub account

1.3 GitHub account:

Here, Azure Repos will be used to store and process the code, for evaluation, this same code is deployed on GitHub as it's accessible, here is the link: https://github.com/DadaNanjesha/batch-processing.

Figure 2: Project Structure

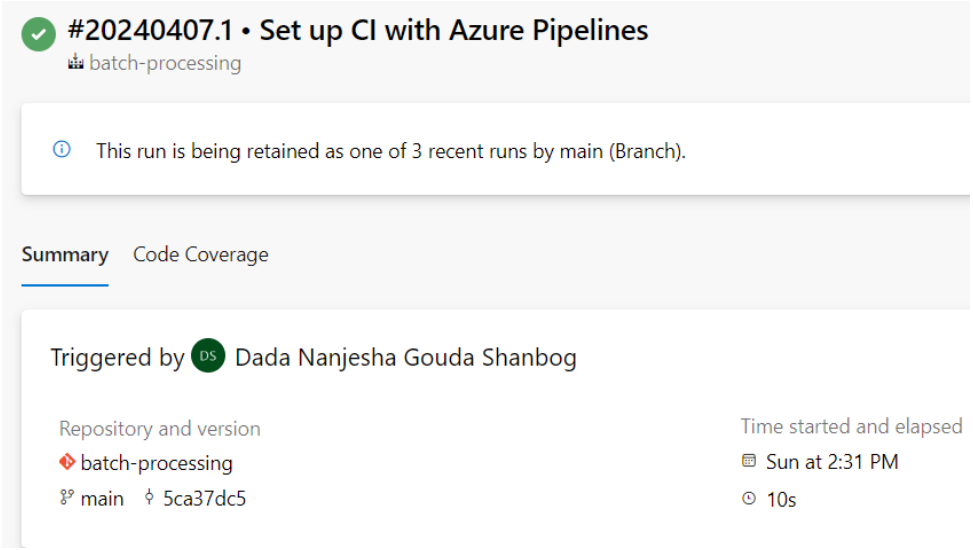


Source: Own representation from the project deployed.

1.4 Pipelines

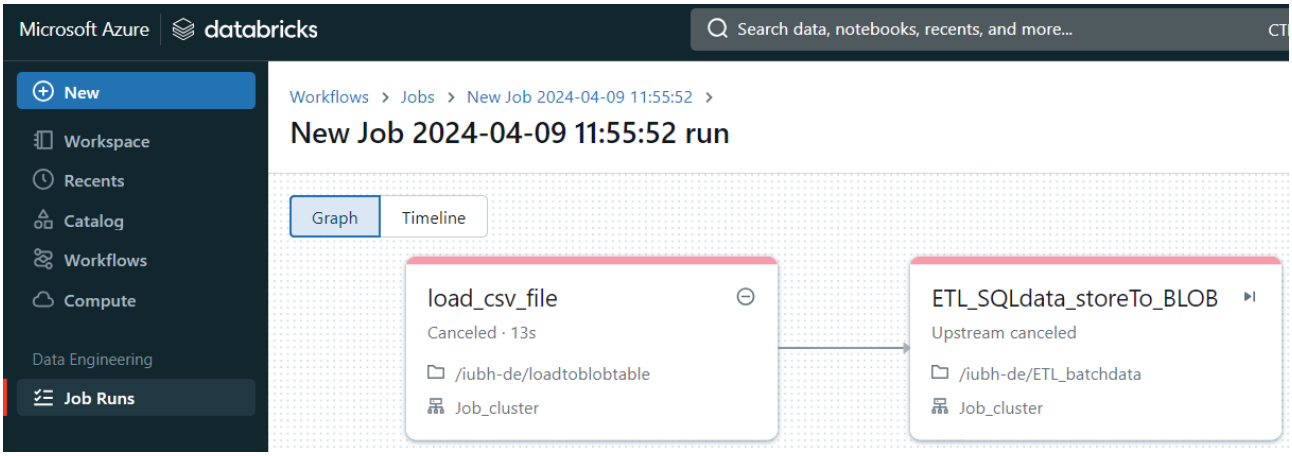
Here, Azure Pipelines is used for continuous integration with the code and services and Azure Databricks workflow is used to synchronize the data with the trigger.

Figure 3: Azure Pipelines



Source: Own representation from the project deployed.

Figure 4: Azure Databricks jobs

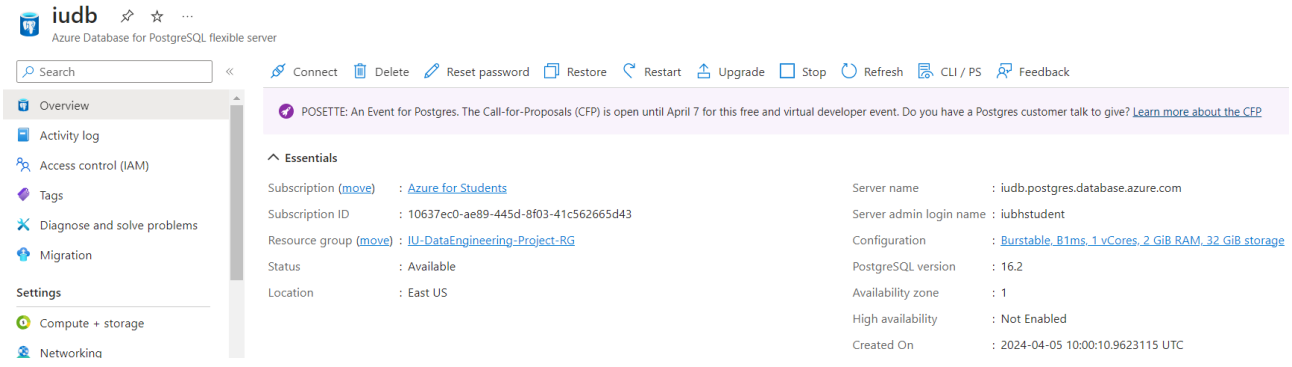


Source: Own representation from the project deployed.

1.5 Databases:

For this project, Azure Blob storage and Azure PostgreSQL are used to store the data. Azure blob storage is used to store the CSV data (dataset from Kaggle) and Parquet data (output of batch processing).

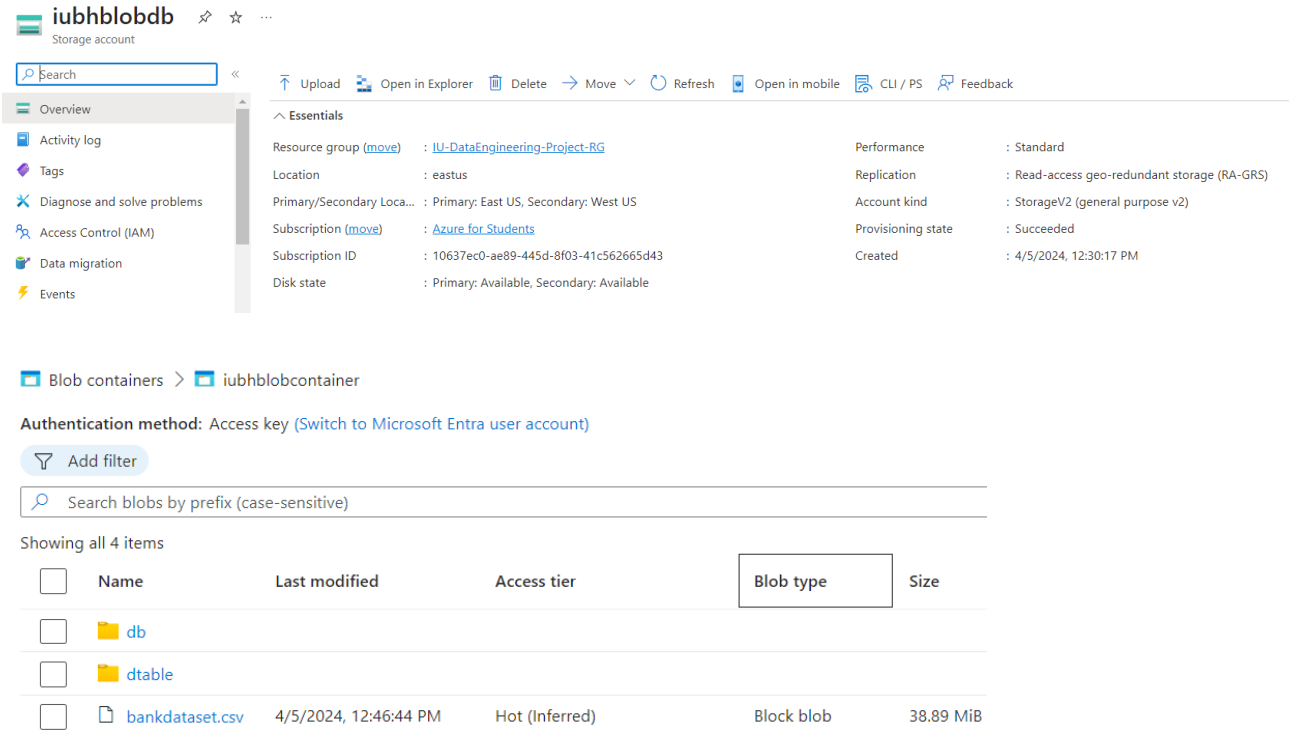
Figure 5: Azure PostgreSQL



Name ↑	Character set	Collation	Schema type
azure_maintenance	UTF8	en_US.utf8	System
azure_sys	UTF8	en_US.utf8	System
postgres	UTF8	en_US.utf8	User
iu_batchprocessed_db	UTF8	en_US.utf8	User

Source: Own representation from the project deployed.

Figure 6: Azure Blob Storage account



Source: Own representation from the project deployed.

1.6 Azure Resources

Initially, to streamline the process, an Azure active directory account is created, here with "*Azure for Students*" subscription. A resource group is created with the name *IU-DataEngineering-Project-RG.* Within this resource group services which are required are created as shown in the below image.

Figure 7: Azure Resources

Name ↑	Type	Location	Resource Group	Subscription
iudb	Azure Database for PostgreSQL flexible server	East US	IU-DataEngineering-Project-RG	Azure for Students
iubh_databricks	Azure Databricks Service	East US	IU-DataEngineering-Project-RG	Azure for Students
workers-sg	Network security group	East US	databricks-rg-iubh_databricks-...	Azure for Students
NetworkWatcherRG	Resource group	East US	NetworkWatcherRG	Azure for Students
NetworkWatcher_eastus	Network Watcher	East US	NetworkWatcherRG	Azure for Students
iubhblobdb	Storage account		IU-DataEngineering-Project-RG	Azure for Students
IU-DataEngineering-Project-RG	Resource group	East US	IU-DataEngineering-Project-RG	Azure for Students
dbmanagedidentity	Managed Identity	East US	databricks-rg-iubh_databricks-...	Azure for Students
IU-key	Key vault	East US	IU-DataEngineering-Project-RG	Azure for Students
Azure for Students	Subscription			Azure for Students

Source: Own representation from the project deployed.

1.7 Data Processing

Here Azure Databricks cluster is used which offers an unparalleled ETL (extract, transform, load) experience by fusing the power of Apache Spark with Delta Lake and proprietary tools, which are used to induce the data to the storage. This can compose ETL logic using Python, Scala, and SQL, and then with a few clicks, orchestrate scheduled job deployment. We are using Databricks Runtime Version 15.0 (includes Apache Spark 3.5.0, Scala 2.12) with the node type Standard_DS3_v2, 14 GB memory 4 cores processor.

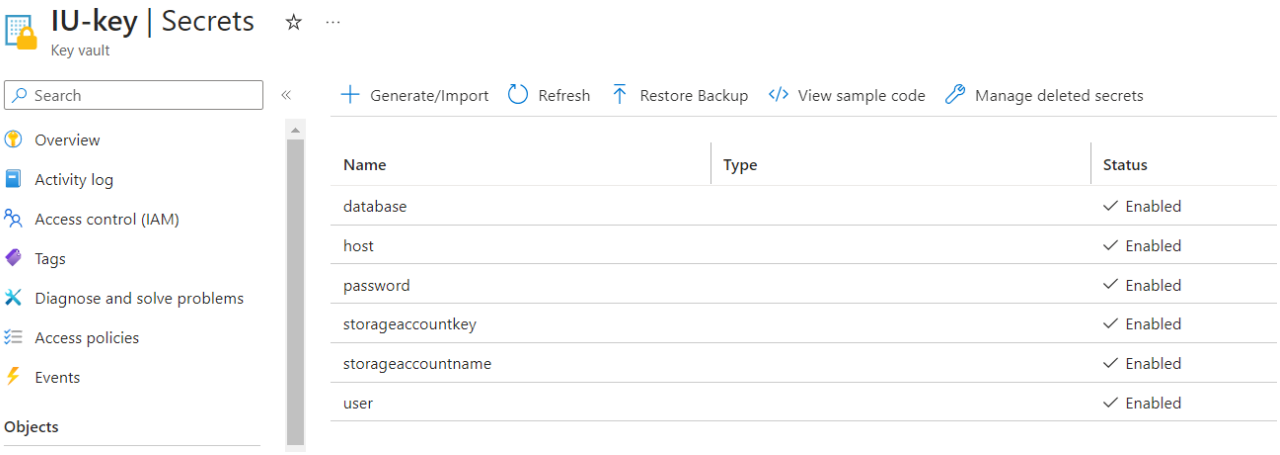
Figure 8: Azure Databricks cluster

Source: Own representation from the project deployed.

1.8 Data privacy

Network traffic between Azure resources in an Azure virtual network can be filtered using an Azure network security group. A network security group contains security rules that allow or prohibit network traffic passing different types of Azure resources. Identity access management (IAM) provides governance as Cloud administrators configure and integrate coarse access control permissions for Unity Catalog and to secure personal data and sensitive information Azure Vault is used as shown below.

Figure 9: Azure Key Vault



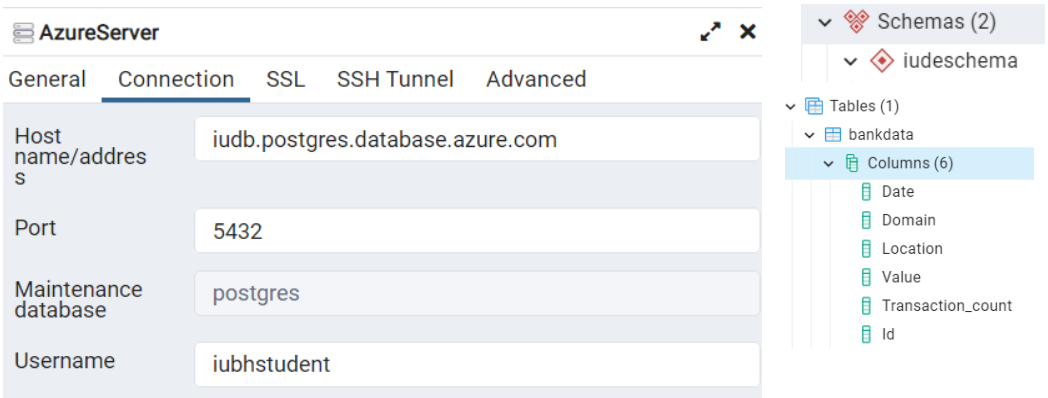
Source: Own representation from the project deployed.

2. Project development workflow and execution:

After creating the resources required for the project in the above stages, now here the execution part will be discussed.

First, the base schema and table are created in the Azure PostgreSQL server, which is required to store the data, fetched from the CSV file. This is processed by the Azure databricks cluster, where all the data cleaning and sorting part is done.

Figure 10: Azure PostgreSQL Database table.



Source: Own representation from the project deployed.

Secondly, Load the CSV dataset to the blob storage and then the Azure Databricks jobs will trigger the job to run the PySpark program which will transform, clean, and filter this CSV data into SQL data and then store it in the PostgreSQL database.

This SQL data is then consumed by Azure Databricks PySpark which will load this data to Azure Blob storage in the Parquet format which in later versions can be consumed for application consumption. This trigger to load the data is initiated every month and this can be changed based on requirements.

In these two phases, we discussed all the tools and technologies, monitoring status, the dataset that will be used and the process of execution. In the next phase, the results, structure of tables and output of the process will be added.

3. Risk and improvements:

As this is a cloud-based project thorough professional knowledge is required to set up the instances and create the required configurations. It's also important to check the cost management and analysis of the required technology for the execution. This may lead to a hefty amount in the later part if not taken care of, this may be the possible risk.

For the improvement part, there is always a space to improve and accelerate the execution and there are no models which are perfect. However, this project can be improvised with the additional services using microservices to trigger and load the data.

References

Wilder, B. (2012). *Cloud architecture patterns: using Microsoft azure*. " O'Reilly Media, Inc.".

Collier, M., & Shahan, R. (2015). *Microsoft azure essentials-fundamentals of azure*. Microsoft Press.

Dudley, R. J. (2010). *Microsoft Azure: Enterprise Application Development*. Packt Publishing Ltd.

Galiveeti, S., Tawalbeh, L. A., Tawalbeh, M., & El-Latif, A. A. A. (2021). Cybersecurity analysis: Investigating the data integrity and privacy in AWS and Azure cloud platforms. In *Artificial intelligence and blockchain for future cybersecurity applications* (pp. 329-360). Cham: Springer International Publishing.

Sharma, V., Nigam, V., & Sharma, A. K. (2020). Cognitive analysis of deploying web applications on microsoft windows azure and amazon web services in global scenario. *Materials Today: Proceedings*, *11*.