AZURE Data Factory Hands-On Mini Project

1. Why should one use Azure Key Vault when working in the Azure environment? What are the pros and cons? What are the alternatives?

**Azure Key Vault is a** centralizing storage of application secrets in Azure Key Vault allows you to control their distribution. Key Vault greatly reduces the chances that secrets may be accidentally leaked.

Pros:

1. Security of storing sensitive information in credentials store which only the ADF service or Administrators can read from
2. If Credentials need to be rotated ADF Linked Service will not need to be modified
3. When we migrate the ADF pipeline from Dev to Test to production no change is required

Cons:

1. Backups are not able to be imported into Key Vaults in another region.

Alternatives:

* Use third party software like AWS Key Management service, LastPass, CyberArk, KeePass

1. How do you achieve loop functionality within a Azure Data Factory pipeline? Why would you need to use this functionality in a data pipeline?

ADF foreach and until activities are designed to handle loop functionality within an Azure data Factory pipeline.

Data integration flows often involve execution of the same tasks on many similar objects. A typical example could be - copying multiple files from one folder into another or copying multiple tables from one database into another.

1. What are expressions in Azure Data Factory? How are they helpful when designing a data pipeline? Please explain with an example.

Expressions are evaluated at runtime. They are very helpful when we are passing the external values into pipeline or when we need to perform repetitive tasks.

Example: we have to run the same pipeline for multiple dates to download a file from a source. We can create an expression that gets the value from a lookup activity that loads the list of dates from a file.

1. What are the pros and cons of parametrizing a dataset’s activity in Azure Data Factory?

Parameters provide a capability to pass external values into pipelines, datasets, linked services, and data flows. ADF does not allow data parameters so dates need to be passed as a string in format “yyyy0MM-dd” format.

1. What are the different supported file formats and compression codecs in Azure Data Factory? When will you use a Parquet file over an ORC file? Why would you choose an AVRO file format over a Parquet file format?

Different supported file formats and compression codecs in Azure Data Factory are:

1. Avro
2. Binary
3. Delimited text
4. Excel
5. JSON
6. ORC
7. Parquet
8. XML

Parquet and ORC both store data in columns and are great for reading data, making queries easier and faster by compressing data and retrieving data from specified columns rather than the whole table. Parquet is better optimized for use with Apache spark. [[1]](#footnote-1)

For schema evolution support or the capability of the file structure to change over time, the winner is Avro since it uses JSON in a unique manner to describe the data, while using binary format to reduce storage size. Avro has row-based data storage and excels at writing data. [[2]](#footnote-2)

1. http://www.differencebetween.net/technology/difference-between-orc-and-parquet/

   2https://bryteflow.com/how-to-choose-between-parquet-orc-and-avro/ [↑](#footnote-ref-1)
2. \* <http://www.differencebetween.net/technology/difference-between-orc-and-parquet/>

   \*\* https://bryteflow.com/how-to-choose-between-parquet-orc-and-avro/ [↑](#footnote-ref-2)