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

   There are a multitude of reasons why you should use Azure Key Vault when working in an Azure environment. One reason is the ability to create and import your keys within minutes. Another reason is because it allows for task automation with SSL/TLS certificates.

   Some alternatives to using Azure Key Vault are SAS tokens.

   With Azure Key Vault you are able to manage and monitor how keys are being used. One of the downsides of using Azure Key Vault are pricing and key transaction limits.

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

   Loop functionality in Azure Data Factory is achieved by using the ‘ForEach’ activity/selection within the pipeline activities pane. This loop functionality is similar to using a for loop within a programming language and can be used to iterate over files.

3. What are expressions in Azure Data Factory? How are they helpful when designing a data pipeline (please explain with an example)?

   Expressions allow for the passing of external values outside of an ETL workflow. They also have the ability to call various functions. With the use of expressions you can pass input and output file paths to your ETL pipeline.

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

Parametrizing a dataset in Azure Data Factory has many pros and cons. By turning a given dataset into a variable or parameter we can now easily use it within another data workflow. Essentially, saving us valuable time and allowing for reproducibility. Although, one of the trade offs here could be readability. As a new engineer, I might not be familiar with this concept.

5. 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?

   Avro format

   Binary format

   Delimited text format

   Excel format

   JSON format

   ORC format

   Parquet format

   XML format

   A parquet file should be used over an ORC file when using data processing tools like Spark.

   If we are performing intensive writing operations we should use AVRO over parquet due to its row based format.