1. What is Azure Data Factory (ADF), and what are its key components?
2. Explain the key differences between Azure Data Factory and SSIS (SQL Server Integration Services).
3. What is the purpose of linked services in Azure Data Factory? How do you define a linked service?
4. Can you explain the concept of pipelines in Azure Data Factory?
5. What are activities in Azure Data Factory? Provide examples of different types of activities.
6. How do you monitor and troubleshoot Azure Data Factory pipelines and activities?
7. What are data sets in Azure Data Factory? How are they defined?
8. How do you handle errors and retries in Azure Data Factory pipelines?
9. How can you transform data within Azure Data Factory? Provide examples of data transformation activities.
10. How do you schedule and orchestrate data movement and processing in Azure Data Factory?
11. Can you explain the concept of triggers in Azure Data Factory?
12. What is the purpose of integration runtimes in Azure Data Factory? How do they enable data movement?
13. How do you integrate Azure Data Factory with other Azure services, such as Azure Databricks or Azure Synapse Analytics?
14. Explain the concept of data flow in Azure Data Factory and its benefits.
15. What security measures can you implement in Azure Data Factory to protect sensitive data?
16. How do you handle incremental data loading in Azure Data Factory pipelines?
17. Can you provide an example of using Azure Data Factory to extract data from an on-premises SQL Server database and load it into Azure Blob storage?
18. Have you worked with any data integration patterns or frameworks in Azure Data Factory? Explain your experience.
19. How would you handle a scenario where you need to process large volumes of data in Azure Data Factory? Discuss potential optimization techniques.
20. What are the best practices for performance tuning and optimizing data movement in Azure Data Factory?

Answers:

1. What is Azure Data Factory (ADF), and what are its key components?

A – ADF is a code free ETL tool on the cloud. With ADF, we can create, manage and schedule data pipelines that can ingest data from multiple sources, perform data transformations and load into necessary data stores.

Key components – Pipelines, Linked Services, Activities (transformation), Datasets

2. E**xplain the key differences between Azure Data Factory and SSIS (SQL Server Integration Services).**

A – ADF is cloud based data integration service, orchestrates data pipelines and ETL workflows. SSIS (SQL server integration service) is on-premise ETL tool for on-premise applications.

1. SSIS can't handle bigdata loads, ADF has AzureHDInsight – a Hadoop service to process bigdata.
2. SSIS can only handle batch processing, ADF can handle Batch and Stream processing
3. SSIS only structured data, ADF handles structured, semi-structured, and unstructured data.
4. SSIS is hardware restricted. ADF is Azure managed, so highly available and scalable.
5. SSIS need custom integration with Azure services. ADF is more integrated with Azure services and enables end-to-end data pipelines on Azure.
6. ADF is designed for modern data integration scenarios supporting cloud and hybrid environments, SSIS has on-premise data integration.

3. **What is the purpose of linked services in Azure Data Factory? How do you define a linked service?**
   A – Linked Services are connection configurations between ADF and other services. ADF establishes a logical connection between different data sources and destinations using Linked Services. Linked Servies are defined in JSON format, and stores details like credentials, URLs, and authentication methods.

4. **Can you explain the concept of pipelines in Azure Data Factory?**
   A – Pipelines are logical grouping of activities that define data movement and transformations. Each pipeline consists of activities that include ingesting data, transforming and storing data. These pipelines are modular structured, so that they are reusable, scalable and easily maintained.

5. **What are activities in Azure Data Factory? Provide examples of different types of activities**
   A – Activities are building blocks of data pipelines. Each pipeline consists of activities. Each activity represent the operation that is executed on data. Activities include data movement and data transformation.
   Data movement activities include Copy, Move, and Delete, allowing data to be transferred between data stores.
   Data Transformation activities include query-like operations – select, groupby, aggregation, derived column.
   Other activities include Execute Pipeline, Lookup, Web Activity, HDInsight Hive Activity, and more, providing a wide range of operations to support diverse data integration scenarios.

6. **How do you monitor and troubleshoot Azure Data Factory pipelines and activities?**
   A – ADF provides monitoring capabilities through Azure Monitor and the Data Factory portal.

   For troubleshooting, you can use Debug mode to step through pipeline activities, view data previews, and inspect data transformations.

   Azure Data Factory also offers logging options, allowing you to track data movement and transformation progress, as well as any errors that occur during pipeline execution.

7. **What are data sets in Azure Data Factory? How are they defined?**
   A – Datasets in ADF represent data structures that define the data to be processed in activities.
   They tell the metadata about data such as schema, partition scheme, file format, etc.
   Datasets act as pointers to the data present in data stores.
   Datasets are defined in JSON format, mention connection details, data structure and required transf settings.

8. **How do you handle errors and retries in Azure Data Factory pipelines?**
   A – ADF provides built-in error handling and retry mechanisms.

   For each activity, you can configure retry policies, specifying the number of retries and retry intervals in case of failures.

   You can also set up error handling mechanisms to manage failed activity runs, like redirecting data to an error store or notifying stakeholders about the failures.
   In addition, you can use Azure Logic Apps or Azure Functions to implement more advanced error handling and notification workflows.

9. **How can you transform data within Azure Data Factory? Provide examples of data transformation activities.**
   A – Data transformation can be done using Data Flow. Provides a drag and drop UI to select activities to ingest data, transform and load data.
   Transformations include filter, pivot, groupby, aggregate, functions, derived column, changing data types, etc.

10. **How do you schedule and orchestrate data movement and processing in Azure Data Factory?**
    A – ADF allows to schedule and orchestrate pipelines using triggers. – Built-in and Custom
    Built-in triggers – schedule based – hourly, daily, monthly. Event based triggers.
    Custom – Can create custom triggers based on external events using Azure LogicApps or Azure Functions to start pipeline execution.

11. **Can you explain the concept of triggers in Azure Data Factory?**
A – Triggers are used to run a pipeline without manual intervention – automated execution of pipelines.
2 types – Built-in and custom triggers
Built-in -> Time based and event based (data arrived in datalake, data started streaming in event hub)
Custom -> Integration with Azure LogicApps and Functions to write custom logic to execute the pipeline.

12. **What is the purpose of integration runtimes in Azure Data Factory? How do they enable data movement?**
A – Integration runtime is the compute infrastructure used by ADF and Synapse to provide data movement and transformation (Data Flows).
3 types -> Azure hosted, Self-hosted and Azure – SSIS
Azure Integration Runtime is fully managed and is used for cloud-to-cloud or cloud-to-on-premises data movement.
Self-hosted Integration Runtime is installed on an on-premises server and is used for data movement between on-premises and cloud data stores.
Azure-SSIS Integration Runtime is used for running SSIS packages in Azure Data Factory.
Self-hosted IR > Azure hosted IR > Global Azure IR

13. **How do you integrate Azure Data Factory with other Azure services, such as Azure Databricks or Azure Synapse Analytics?**
A – ADF integrated with Databricks and Synapse using Native connectors and activities.
For example, you can use the Azure Databricks Notebook activity to run notebooks in Azure Databricks as part of a pipeline.
The Copy activity supports moving data to and from Azure Synapse Analytics (formerly Azure SQL Data Warehouse).

14. **Explain the concept of data flow in Azure Data Factory and its benefits.**
A – Data flow is a GUI based workspace to build data pipelines in ADF. Code-free approach, can handle PP massively parallel processing workloads.

15. **What security measures can you implement in Azure Data Factory to protect sensitive data?**
A – Use managed identities and role-based access control (RBAC) to control access to Azure Data Factory resources.
Implement data encryption at rest and in transit to protect sensitive data during storage and transmission.
Store sensitive information like connection strings or credentials securely using Azure Key Vault and use linked services with Key Vault references.
Utilize Azure Data Factory Managed Private Endpoints to enhance data security for data movement activities.

16. **How do you handle incremental data loading in Azure Data Factory pipelines?**
A – Use timestamp or watermark based techniques to identify last processed data and new data.
Design the data loading pipeline to identify and load only the new data.
Use filter conditions to select the new data based in timestamp.

17. C**an you provide an example of using Azure Data Factory to extract data from an on-premises SQL Server database and load it into Azure Blob storage?**
Create a self-hosted integration runtime on on-premise server and register it with ADF.
Create linked services in ADF connecting it with Blob storage and with on-premise SQL server.
Build and execute a copy pipeline that moved data from on-premise server to Blob storage.

18. **Have you worked with any data integration patterns or frameworks in Azure Data Factory? Explain your experience.**
Common data integration patterns in Azure Data Factory include Extract-Load-Transform (ELT), Incremental Load, Change Data Capture (CDC), and Data Hub.
    - ELT involves moving raw data to the destination and performing transformations on the destination side.
    - Incremental Load only processes new or changed data since the last execution.
    - CDC captures only the changes made to the source data since the last execution.
    - Data Hub pattern involves staging data in a central location for consistency and reuse.

19. **How would you handle a scenario where you need to process large volumes of data in Azure Data Factory? Discuss potential optimization techniques.**
    - Optimize data movement by using PolyBase or bulk loading techniques for large data sets.
    - Use parallel execution in Data Flows or mapping transformations in Copy activities to increase performance.
    - Implement partitioning techniques for large tables or files to achieve parallelism and optimize data processing.
    - Use the appropriate integration runtime to auto-scale on demand.

20. **What are the best practices for performance tuning and optimizing data movement in Azure Data Factory?**
    - Use partitioning to distribute data processing across multiple nodes.
    - Minimize data movement by processing data closer to its source or destination.
    - Use staging techniques to optimize data flow and transformation in Data Flows.
    - Monitor and analyze pipeline execution times to identify bottlenecks and optimize data integration workflows.
    - Utilize compression and binary format options in data stores to reduce data size during movement.