

Data-Factory-Udemy-Course-Quiz

1. Which of the following is NOT a core component of Azure Data Factory?

- Options
 - Pipeline
 - LinkedService
 - Dataset
 - SQL Database
- Ans: SQL Database
- Azure Data Factory (ADF) consists of three main components: Pipelines: Logical containers for grouping activities. Linked Services: Connection information to external data sources. Datasets: Represent data structures within the Linked Services. A SQL Database is an external resource and not a direct core component of ADF, though it can be used within Linked Services to connect to or pull/push data.

2. What is the primary purpose of a Linked Service in Azure Data Factory?

- Options
 - Defining Structure of Data
 - Connecting ADF to external Data sources
 - Executing Transformations on data
 - Storing data within ADF
- Ans: Connecting ADF to external Data sources
- A Linked Service in Azure Data Factory serves as a connection string or interface to external data stores or computing services. It does not contain the actual data but holds the credentials, endpoints, and connection settings. A Dataset defines the structure and metadata of the data but is tied to the Linked Service. Transformations and data flow execution are handled within pipelines and activities.

3. Which of the following is the primary purpose of Azure Blob Storage?

- Options
 - Storing relational data in tables
 - Storing large amounts of unstructured data
 - Running compute-heavy processes
 - Hosting virtual machines
- Ans: Storing Large amounts of unstructured data
- Azure Blob Storage is designed for storing large amounts of unstructured data, such as text, images, video files, logs, or backups. It is not used for relational data, compute tasks, or VM hosting.

4. Which of the following is NOT a valid storage tier in Azure Blob Storage?

- Options
 - Hot
 - Cool
 - Cold
 - Archive
- Ans: Cold

5. What is the primary purpose of a Resource Group in Azure?

- Options
 - To manage and organize Azure resources as a logical container
 - To monitor network traffic between Azure services
 - To physically store all Azure resources
 - To provide role-based access control at the storage level
- Ans: To manage and organize Azure resources as a logical container

6. What is the primary use of variables in an Azure Data Factory pipeline?

- Options
 - To store and pass values between pipeline runs
 - To define external connections for data movement
 - To define dataset structures for transformation activities

- To monitor and log pipeline execution time
- Ans: To store and pass values between pipeline runs
- In ADF pipelines, variables are used to store temporary values during pipeline execution. They can be used to pass values between activities and loops.

7. Which activity in Azure Data Factory is used to update or assign values to variables?

- Options
 - Lookup Activity
 - Set Variable Activity
 - ForEach Activity
 - Execute Pipeline Activity
- Ans: Set Variable Activity

8. What is the main purpose of using parameters in Azure Data Factory?

- Options
 - To dynamically pass values into pipelines, datasets, or linked services
 - To permanently store execution results
 - To replace variables within pipeline execution
 - To create new storage accounts during pipeline runs
- Ans: To dynamically pass values into pipelines, datasets, or linked services
- Parameters are defined at the pipeline level, dataset level, or linked service level. You assign values to parameters when triggering a pipeline or calling a pipeline from another pipeline.

9. How do parameters differ from variables in Azure Data Factory?

- Options
 - Parameters can change values during pipeline execution, while variables cannot.
 - Parameters are assigned values at runtime, while variables can change within a pipeline run.
 - Parameters are used for system tasks, while variables are for user-defined tasks.
 - Parameters and variables are interchangeable in ADF.

- Ans: Parameters are assigned values at runtime, while variables can change within a pipeline run.
- Parameters are static during execution and are assigned values at the start of the pipeline run. Variables, on the other hand, can be updated or modified dynamically during the pipeline execution.

10. Which of the following is NOT a valid type of Integration Runtime in Azure Data Factory?

- Options
 - Azure Integration Runtime
 - Self-Hosted Integration Runtime
 - Hybrid Integration Runtime
 - Azure-SSIS Integration Runtime
- Ans: Hybrid Integration Runtime
- Azure Data Factory offers three types of Integration Runtimes:
 - Azure Integration Runtime: Managed by Azure, for cloud data movement and transformation.
 - Self-Hosted Integration Runtime: Used for on-premises data movement and processing.
 - Azure-SSIS Integration Runtime: Used for lifting and shifting SSIS packages to Azure.
 - There is no "Hybrid Integration Runtime".

11. Which of the following is a primary use case for a Self-Hosted Integration Runtime?

- Options
 - Running SSIS packages in Azure
 - Connecting to on-premises data sources
 - Moving data between Azure Blob Storage and Azure SQL Database
 - Executing machine learning models in Azure Synapse
- Ans: Connecting to on-premises data sources
- The Self-Hosted Integration Runtime (Self-Hosted IR) is used for: On-premises to cloud data movement (e.g., SQL Server to Azure Data Lake).

12. Which of the following is a valid JSON structure in Azure Data Factory?

Options

Option 1

```
{
  "name": "Pipeline1",
  "activities": [
    {"name": "Activity1", "type": "Copy"}
  ]
}
```

Option 2

```
{
  name: "Pipeline1",
  activities: [
    {name: "Activity1", type: "Copy"}
  ]
}
```

Option 3

```
[
  "name": "Pipeline1",
  "activities": {"name": "Activity1"}
]
```

Option 4

```
{
  "name": "Pipeline1",
```

```
"activities": {"name": "Activity1"}  
}
```

Answer: Option 1

13. What is the purpose of the JSON structure in an Azure Data Factory pipeline definition?

- Options
 - To define external Linked Services in YAML format
 - To specify the metadata, activities, and configurations of the pipeline
 - To store data directly within the pipeline
 - To create Azure Blob Storage accounts automatically
- Ans: To specify the metadata, activities, and configurations of the pipeline
- In Azure Data Factory, pipelines are defined using a JSON structure. The JSON specifies:
 - Metadata: Name and properties of the pipeline.
 - Activities: Actions performed (e.g., Copy, ForEach, ExecutePipeline).
 - Configuration: Parameters, variables, and settings for activities.

14. ____ Activities help to execute SQL Queries in ADF

- Options
 - Execute pipeline
 - Lookup
 - Stored Procedure
- Ans: Stored Procedure

15. Data in data factory can be monitored through ____.

- Options
 - Azure Monitor
 - Data explorer
 - Pipeline Runs
- Ans: Pipeline Runs

16. In ADF, a ____ Activity allows you to wait for a specific time before proceeding

- Options
 - Wait
 - ForEach
 - If Condition
- Ans: Wait

17. In ADF, ____ Activities handle exceptions and retry logic

- Ans: Retry Policies

18. In ADF, ____ Enables the integration of on-premises data sources

- Options
 - Data bricks
 - Integration Runtime
 - Datasets
- Ans: Integration Runtime

19. Azure Data factory's ____ helps in managing multiple environments for data movement.

- Ans: Integration runtime

20. What are the Activities in Azure Data factory

1. Copy

- **What it does:** Copies data from one place to another.
- **Example:** Move data from an Excel file in OneDrive to a SQL database.

2. Append Variable

- **What it does:** Adds a value to an existing list variable.
- **Example:** Add each file name to a list as you loop through files.

3. Delete

- **What it does:** Deletes files or folders.
- **Example:** Remove old files from a storage location after processing.

4. Execute Pipeline

- **What it does:** Runs another pipeline from within your current pipeline.
- **Example:** Run a cleanup pipeline after your main data load pipeline.

5. Fail

- **What it does:** Stops the pipeline and marks it as failed.
- **Example:** If a file is missing, use this to stop everything and alert someone.

6. Get Metadata

- **What it does:** Retrieves information about a file or folder.
- **Example:** Check the size or last modified date of a file.

7. Lookup

- **What it does:** Reads a small amount of data (like a config or a single row).
- **Example:** Get a list of file names from a control table.

8. Set Variable

- **What it does:** Assigns a value to a variable.
- **Example:** Set a variable to today's date.

9. Wait

- **What it does:** Pauses the pipeline for a set amount of time.
- **Example:** Wait 5 minutes before checking if a file has arrived.

10. Switch

- **What it does:** Like a multiple-choice decision.
- **Example:** If the file type is CSV, do one thing; if it's JSON, do another.

11. Filter

- **What it does:** Filters a list based on a condition.

- **Example:** From a list of files, keep only the ones modified today.

12. ForEach

- **What it does:** Loops through a list and performs actions for each item.
- **Example:** Process each file in a folder one by one.

13. Until

- **What it does:** Repeats actions until a condition is true.
- **Example:** Keep checking for a file until it appears.

14. If Condition

- **What it does:** Runs different actions based on a true/false condition.
- **Example:** If a file exists, process it; otherwise, send an alert.



21. What are the Transformations in Data Flows

Transformations are like **data editing tools**. They change or shape your data.

1. Conditional Split

- **What it does:** Splits data into different paths based on conditions.
- **Example:** Send high-value orders to one path, low-value to another.

2. Exists

- **What it does:** Checks if matching data exists in another dataset.
- **Example:** Keep only customers who have placed an order.

3. Union

- **What it does:** Combines rows from multiple datasets.
- **Example:** Combine sales data from different regions.

4. Join

- **What it does:** Merges data from two datasets based on a common column.
- **Example:** Join customer info with their orders.

5. Aggregate

- **What it does:** Summarizes data (like sum, average, count).
- **Example:** Total sales per region.

6. Surrogate Key

- **What it does:** Adds a unique ID to each row.
- **Example:** Add a new ID column to a dataset that doesn't have one.

7. Select

- **What it does:** Chooses which columns to keep or rename.
- **Example:** Keep only name and email columns.

8. Lookup

- **What it does:** Adds data from another dataset
- **Example:** Add product names to a sales dataset using product IDs.

9. Derived Column

- **What it does:** Creates new columns using expressions or formulas.
- **Example:** Add a column that calculates tax from price.

10. Pivot

- **What it does:** Turns rows into columns.
- **Example:** Turn monthly sales rows into columns for each month.

11. Unpivot

- **What it does:** Turns columns into rows.
- **Example:** Convert monthly sales columns into rows for each month.

12. Rank

- **What it does:** Assigns a rank to rows based on a value.
- **Example:** Rank products by sales.

13. Window

- **What it does:** Performs calculations across a set of rows.
- **Example:** Calculate running totals or moving averages.

14. Flatten

- **What it does:** Expands nested data (like JSON arrays) into rows.
- **Example:** Turn a list of items in a single row into multiple rows.

15. Filter

- **What it does:** Keeps only rows that meet a condition.
- **Example:** Keep only orders over \$100.