Cloud Computing Essentials with Azure and AWS

# **3.1 Core Azure Services for Data Analytics**

## Storage Accounts

* **Storage Types**:
  + **Blob Storage**: Stores unstructured data like text and binaries.
  + **File Storage**: Shared storage for legacy apps using SMB protocol.
  + **Queue Storage**: Manages messages across distributed systems.
  + **Table Storage**: NoSQL key-attribute store, highly available.
  + Data is automatically replicated to ensure durability and high availability. Options include:
    - **LRS (Locally Redundant Storage)**: Basic protection within a single data center. Low cost for non-critical scenarios.
    - **GRS (Geo-Redundant Storage)**: Backup with failover to a secondary region. For backup scenarios.
    - **ZRS (Zone-Redundant Storage)**: Protection against data center failures. For high availability.
    - **GZRS (Geo-Zone-Redundant Storage)**: Combines GRS and ZRS for critical data protection – optimal protection selection
  + **Scalability**: Auto-scales for high demand and large workloads.
* **Azure Storage Includes**:
  + Azure Data Lake Storage Gen2
  + Azure Files
  + Azure Queues
  + Azure Tables
* **Integration**:
  + Works seamlessly with Azure services for analytics, computing, and networking.
* **Storage Browser**:
  + Graphical interface for managing storage accounts and their components like:
  + **Blob Containers:**
    - **Purpose**: Used for storing unstructured data like images, videos, and documents.
    - **Key Features**:
      * **Hierarchy**: Organizes blobs within folders (virtual directories).
      * **Access Tiers**: Hot (frequent access), Cool (infrequent access), and Archive (long-term storage).
      * **Access Control**: Supports Shared Access Signatures (SAS) and Azure role-based access control (RBAC).
      * **Use Cases**: Backups, streaming media, log storage, and big data analytics.
  + **File Shares**:
    - **Purpose**: Provides shared storage accessible via SMB protocol, ideal for lift-and-shift migration of legacy applications.
    - **Key Features**:
      * **Access**: Can be mounted to Windows, macOS, and Linux systems.
      * **Scalability**: Automatic scaling based on file size and number of files.
      * **Integration**: Works with Azure Backup for file-level recovery and Azure File Sync for hybrid storage.
      * **Use Cases**: File sharing, centralized storage for applications, and on-premise/cloud hybrid solutions.
  + **Key Differences**:
    - **Blob Containers**: Focused on unstructured data and large-scale analytics.
    - **File Shares**: Designed for traditional file storage with network file-sharing capabilities.
  + **Tools in Storage Browser**:
    - **Upload/Download**: Simplified options for moving data.
    - **Container/File Management**: Create, delete, and edit containers or file shares.
    - **Monitoring**: Insights into usage, performance, and access logs.

## SQL Databases

* **Azure SQL Database**:
  + Fully managed platform as a service (PaaS) database engine that handles most database management functions without user involvement.
* **Key features and use cases of Azure SQL Database**:
  + **Fully Managed**: Automatic backups, patching, monitoring, and performance tuning.
  + **Scalable**: Auto-scale resources up/down; pay only for what you use.
  + **High Availability**: 99.99% uptime SLA with multi-location data replication.

## Azure Data Factory Introduction

* **Azure Data Factory**: Cloud-based service for creating, scheduling, and orchestrating data workflows.
* **Use Case**: Ingest, prepare, and transform data from multiple sources into centralized stores for analysis.
* **Best For**: Managing complex data flows across on-premises and cloud environments.
* **What is Azure Data Factory?**
  + Cloud-based data integration service.
  + Helps create, schedule, and manage workflows for data movement and transformation.
  + Works with both on-premises and cloud data sources.
* **Key Features**
  + **Data Ingestion**: Connects to diverse data sources like SQL, Blob Storage, REST APIs, and more.
  + **Data Transformation**: Supports ETL/ELT processes with data flow pipelines.
  + **Orchestration**: Schedules and automates complex workflows.
  + **Integration**: Works seamlessly with other Azure services like Synapse Analytics, Databricks, and Storage.
  + **Monitoring**: Provides built-in activity tracking and alerts for pipeline performance.
* **What It’s Good For**
  + **Data Consolidation**: Moves data from multiple sources to centralized storage.
  + **Hybrid Scenarios**: Handles on-premises and cloud data integration.
  + **Large Workloads**: Processes big data efficiently with scale-out capability.
  + **Flexible Triggers**: Allows event-based, schedule-based, or manual pipeline triggers.
* **Supported Data Sources**
  + **Azure services**: Blob Storage, Data Lake, SQL Database, Synapse Analytics.
  + **Third-party sources**: Amazon S3, Google Cloud Storage, REST APIs.
  + **On-premises**: SQL Server, file systems, Oracle, etc.
* **Advantages**
  + **No-code Data Flows**: Drag-and-drop tools for building pipelines.
  + **Cost-effective**: Pay-as-you-go pricing model.
  + **Scalable**: Automatically scales resources to handle large data volumes.
  + **Secure**: Built-in data encryption and managed identity for access control.
* **Common Use Cases**
  + Migrating on-premises data to the cloud.
  + Data preparation for analytics or reporting.
  + Building and managing data pipelines for machine learning workflows.
  + Event-driven workflows like processing IoT data.

## Azure Data Factory – Blob Storage to SQL Databases

* **Overview**
  + Transfers data from Azure Blob Storage to SQL Databases using pipelines.
  + Commonly used for ETL/ELT workflows to prepare data for analysis or reporting.
* **Workflow Steps**
  + **Create a Linked Service**:
    - Define connections to Blob Storage and SQL Database.
    - Linked services act as data source/destination configurations.
  + **Set Up Data Sets**:
    - Define the Blob Storage dataset (e.g., CSV, JSON, Parquet files).
    - Define the SQL Database dataset (e.g., a specific table or schema).
  + **Build a Pipeline**:
    - Use Copy Activity to move data between Blob Storage and SQL.
    - Optionally, use Data Flow for transformations (e.g., data cleansing, filtering).
  + **Configure Pipeline Triggers**:
    - **Schedule Trigger**: Runs pipelines at regular intervals.
    - **Event-Based Trigger**: Starts pipeline when new data is uploaded to Blob Storage.
  + **Run and Monitor**:
    - Execute the pipeline and monitor using Azure Data Factory's monitoring dashboard.
* **Key Features**
  + **Data Transformation**: Use mapping data flows to clean and modify data during transfer.
  + **Error Handling**: Built-in retry policies and logging for debugging failed runs.
  + **Scalability**: Automatically scales resources based on data size.
* **Benefits**
  + Simplifies data migration from Blob Storage to SQL without coding.
  + Automates data movement, reducing manual intervention.
  + Secure integration with Azure services using Managed Identity.
  + Supports structured, semi-structured, and unstructured data formats.
* **Common Use Cases**
  + Loading raw files (e.g., CSV) from Blob Storage into SQL for reporting.
  + Consolidating multiple data sources into a single SQL Database.
  + Preparing data for machine learning or BI tools like Power BI.
  + Automating periodic updates to SQL tables from Blob Storage.
* **Quick Tips**
  + Use Self-hosted Integration Runtime for on-prem SQL databases.
  + Optimize performance by enabling parallel copy in the Copy Activity.
  + Leverage Data Preview in mapping data flows to validate transformations.
* Azure Data Factory is an efficient tool for automating data integration between Blob Storage and SQL Databases, handling both large-scale and real-time data workflows seamlessly.

## Transforming Data in Azure Data Factory

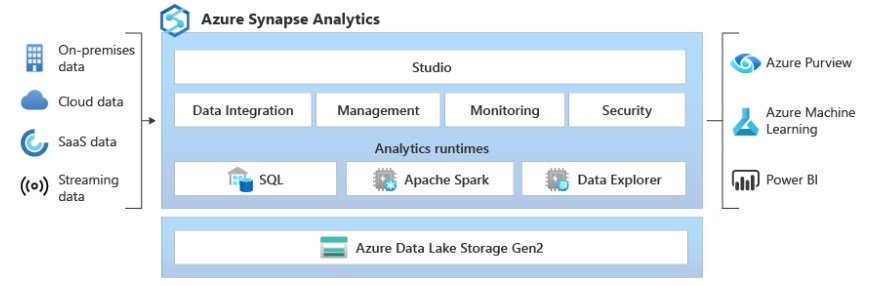
* **Overview**
  + Transformation involves modifying, cleaning, and shaping raw data to make it suitable for analysis or storage.
  + Azure Data Factory uses Mapping Data Flows and Custom Activities for data transformation.
* **Key Features for Data Transformation**
  + **Mapping Data Flows**:
    - No-code, visual interface for building transformations.
    - Supports operations like filtering, joining, aggregating, and sorting data.
    - Optimized for big data with Spark-based execution.
  + **Wrangling Data Flows**:
    - Allows data preparation using Power Query interface.
    - Ideal for simple transformations with interactive previews.
  + **Custom Activities**:
    - Use Azure Databricks or HDInsight for advanced custom transformations.
    - Write custom Python, Scala, or Spark scripts for flexibility.
  + **Inline Data Transformations**:
    - Perform basic transformations during Copy Activity (e.g., column mapping, data filtering).
* **Common Transformations**
  + **Filter Data**: Remove rows that don't meet criteria.
  + **Aggregate**: Summarize data (e.g., calculate totals, averages).
  + **Join Datasets**: Combine data from multiple sources based on keys.
  + **Derived Columns**: Create new columns using expressions.
  + **Schema Mapping**: Align source schema with destination schema.
* **Best Practices**
  + Test transformations using Data Preview.
  + Optimize performance by using partitioning in data flows.
  + Minimize transformation complexity by breaking them into smaller pipelines.

## Pipeline Orchestration in Azure Data Factory

* **Overview**
  + Orchestration manages and automates workflows by chaining multiple activities in a pipeline.
  + Supports dependencies, triggers, and error handling for seamless operation.
* **Key Features for Pipeline Orchestration**
  + **Activities**:
    - **Data Movement**: Copy Activity for transferring data.
    - **Transformation**: Mapping Data Flow, Databricks Activity.
    - **Control Flow**: If Condition, ForEach Loop, Wait, Execute Pipeline.
  + **Triggers**:
    - **Schedule Trigger**: Runs pipelines at specific times.
    - **Event-Based Trigger**: Activates on events like Blob creation.
    - **Manual Trigger**: Starts pipeline on demand.
  + **Dependencies**:
    - Define activity execution order with Success, Failure, or Completion dependencies.
  + **Error Handling**:
    - Configure retry policies and use Fail/Continue settings for fault tolerance.
    - Route failed activities to alternative workflows using On-Failure dependencies.
* **Best Practices**
  + Use parameterized pipelines for reusability across datasets and workflows.
  + Enable logging and monitoring to track pipeline performance.
  + Implement error handling to ensure smooth operation during failures.
  + Break complex pipelines into smaller modular pipelines for better management.

## Azure Synapse Analytics Introduction

* **What is Azure Synapse Analytics?**
  + A unified analytics platform combining big data and data warehousing.
  + Allows seamless integration of data ingestion, preparation, management, and analytics.
* **Key Features**
  + **Integrated Workspace**:
    - Combines SQL, Spark, and Data Explorer for analytics.
    - Single interface for querying, managing, and analyzing data.
  + **Data Warehousing**:
    - Scalable and high-performance SQL-based data warehouse.
    - Supports massive parallel processing (MPP) for large datasets.
  + **Big Data Integration**:
    - Native support for Apache Spark for processing big data.
    - Works with structured, semi-structured, and unstructured data.
  + **Data Pipelines**:
    - Built-in orchestration for data workflows.
    - Similar to Azure Data Factory, with additional Synapse-specific capabilities.
  + **Serverless and Dedicated Pools**:
    - **Serverless**: Pay-per-query model for on-demand SQL processing.
    - **Dedicated**: Pre-allocated resources for consistent, high-performance workloads.
  + **Integration with Azure Services**:
    - Works with Power BI, Azure Machine Learning, and Azure Data Lake.
    - Supports connectors for external data sources like Amazon S3, REST APIs, etc.
* **Core Use Cases**
  + **Data Warehousing**: Store and manage large volumes of structured data for analytics.
  + **Real-Time Analytics**: Analyze live streaming data for quick insights.
  + **Big Data Processing**: Ingest, transform, and analyze large-scale datasets.
  + **BI Integration**: Create interactive reports and dashboards using Power BI.
* **Advantages**
  + Unified platform for data engineering, data science, and BI.
  + Scalable and flexible with serverless and dedicated options.
  + Reduces complexity by integrating data pipelines, warehousing, and analytics.
  + High performance for complex queries with minimal latency.
* **Best Practices**
  + Use serverless SQL pools for ad-hoc querying.
  + Implement dedicated SQL pools for consistent, high-throughput workloads.
  + Optimize big data workloads with Spark pools and partitioning.
  + Integrate with Azure Synapse Link for real-time data access from operational systems.
* Azure Synapse Analytics simplifies end-to-end analytics by combining data integration, warehousing, and big data processing in a single platform, making it a robust solution for modern data needs.



## Azure Synapse Analytics - Data and Develop

* Azure Synapse Analytics provides a unified platform for data integration and development, enabling seamless collaboration among data engineers, data scientists, and analysts.
* **Data Integration**:
  + **Data Hub**: Centralized access to various data sources, including Azure Data Lake Storage, Azure Cosmos DB, and external databases.
  + **Linked Services**: Establish connections to external data sources for streamlined data movement and transformation.
  + **Data Exploration**: Utilize SQL on-demand and Spark pools to query and analyze data without the need for data movement.
* **Development Environment**:
  + **Synapse Studio**: Integrated workspace offering a collaborative interface for building and managing data pipelines, SQL scripts, Spark jobs, and notebooks.
  + **Notebooks**: Support for multiple languages (e.g., Python, Scala, SQL) to develop and execute code interactively.
  + **Code Management**: Integration with Git repositories for version control and collaborative development.
* **Data Processing**:
  + **SQL Pools**: Provisioned and serverless options for scalable data warehousing and querying.
  + **Spark Pools**: Apache Spark integration for big data processing and machine learning workloads.
  + **Data Flows**: Visual data transformation tools for ETL processes without coding.
* **Collaboration and Monitoring**:
  + **Shared Workspaces**: Facilitate teamwork with shared projects, resources, and artifacts.
  + **Monitoring**: Track pipeline runs, data flows, and resource utilization through built-in dashboards.
* **Key Capabilities**
  + **Data Hub**: Centralized access to Azure Data Lake, Cosmos DB, and external databases.
  + **Synapse Studio**: Unified workspace for data pipeline creation, SQL, Spark jobs, and notebooks.
  + **SQL Pools**: On-demand and provisioned SQL options for querying and warehousing.
  + **Spark Integration**: Apache Spark for big data processing and machine learning.
  + **Notebooks**: Interactive coding in Python, Scala, or SQL for real-time development.
* **Use Case**
  + Seamlessly integrate, process, and develop data workflows with built-in tools for analytics and collaboration.

## Azure Synapse Analytics - Automation

* Automation in Azure Synapse Analytics streamlines deployment, monitoring, and management of data workflows, ensuring efficiency and consistency.
* **Continuous Integration and Continuous Deployment (CI/CD)**:
  + **Source Control Integration**: Connect Synapse Studio with Git repositories (e.g., Azure DevOps, GitHub) for version control.
  + **Workspace Deployment**: Use Azure Resource Manager (ARM) templates to automate the deployment of Synapse workspaces and artifacts across environments.
  + **Release Pipelines**: Set up Azure DevOps release pipelines to manage and automate the deployment process.
* **Pipeline Automation**:
  + **Triggers**: Configure schedule-based, event-based, or manual triggers to initiate pipelines automatically.
  + **Parameterization**: Use parameters to create flexible and reusable pipelines adaptable to different scenarios.
  + **Error Handling**: Implement retry policies and conditional logic to manage failures and ensure robust workflows.
* **Monitoring and Alerts**:
  + **Activity Monitoring**: Track the status and performance of pipelines, data flows, and other activities.
  + **Alerts**: Set up alerts to notify stakeholders of failures, successes, or performance issues.
  + **Logging**: Maintain detailed logs for auditing and troubleshooting purposes.
* By leveraging these automation capabilities, organizations can achieve faster development cycles, maintain consistency across deployments, and reduce manual intervention.
* **Key Features**
  + **CI/CD Integration**: Version control with Git and automated deployment via Azure DevOps.
  + **Pipeline Triggers**: Schedule-based, event-driven, or manual workflows.
  + **Error Management**: Built-in retry policies, conditional logic, and error notifications.
  + **Monitoring**: Real-time dashboards for pipeline activity and system health.
* **Use Case**
  + Automate and monitor data workflows, ensuring reliability and efficiency in deployment.

## Azure Databricks

* Azure Databricks is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud, designed to accelerate data engineering and data science workflows.
* **Key Features**:
  + **Collaborative Notebooks**: Interactive notebooks supporting multiple languages (e.g., Python, Scala, SQL) for collaborative development.
  + **Managed Clusters**: Automated cluster management with auto-scaling and auto-termination to optimize
* **Key Features**
  + **Collaborative Notebooks**: Supports Python, Scala, SQL, and R for real-time teamwork.
  + **Managed Clusters**: Auto-scaling and termination for cost efficiency.
  + **Integration**: Works with Azure Data Lake, Synapse, and ML tools.
  + **Streaming**: Real-time data ingestion and processing.
* **Use Case**
  + Simplifies large-scale data engineering, advanced analytics, and machine learning development.

## TL;DR

* **Azure Data Factory**: Focused on moving and orchestrating data pipelines (ETL/ELT).
* **Azure Synapse Analytics**: Combines data warehousing and big data analytics for BI and reporting.
* **Azure Databricks**: Optimized for big data engineering, machine learning, and AI workflows.
* ASA is business-analytics-first; Databricks is data-science-first