Grow Data Skills

BigQuery

Azure Blob Storage

Azure Blob Storage is Microsoft's cloud-based object storage solution for unstructured data. It is designed to store a massive amount of data, such as images, videos, documents, backups, and logs, providing scalable and secure storage with high availability.

Key Components

- **Storage Account:** A storage account is the top-level namespace for your Azure Storage services. It provides a unique namespace in Azure for your data and can be used to access Blob, Queue, Table, and File storage. ***Example***: images-container

- **Blob Containers:** A container is a logical grouping of blobs within a storage account. It provides an organization structure for the blobs, similar to folders in a file system. ***Example***: images-container

- **Blobs:** A blob is an individual piece of data or object stored in Azure Blob Storage. ***Example***: photo.jpg

  The path to access a blob in Azure Blob Storage follows a hierarchical structure:
  https://<storage_account_name>.blob.core.windows.net/<container_name>/<blob_name>

  Example - ***https://mystorageaccount.blob.core.windows.net/images-container/photo.jpg***

- **Security and Access:**
  - ***Access Tiers:*** Azure Blob Storage provides different access tiers to optimize costs based on how frequently data is accessed:
    - ***Hot***: Optimized for data that is accessed frequently.
    - ***Cool***: Lower cost, optimized for data that is infrequently accessed.
    - ***Archive***: Lowest cost, used for data that is rarely accessed and can tolerate retrieval latency.

Azure Blob Storage vs. Azure Data Lake Storage Gen2

Azure Blob Storage and Azure Data Lake Storage Gen2 (ADLS Gen2) are both Azure cloud storage solutions, but they serve different purposes and are optimized for different use cases. Below is a comparison of the two:

Purpose and Use Cases

- **Azure Blob Storage:**
  - ***General-Purpose Storage***: Designed for storing unstructured data such as documents, images, videos, backups, and logs.
  - ***Use Cases***: Media storage, backup and restore, web and mobile applications, and serving files directly to users.

- **Azure Data Lake Storage Gen2 (ADLS Gen2):**
  - ***Big Data Analytics:*** Built on top of Azure Blob Storage but optimized for large-scale analytics workloads.
  - ***Use Cases:*** Data lakes, big data analytics, and processing workloads involving massive datasets, particularly those using tools like Apache Hadoop, Spark, and Azure HDInsight.

Key Features

- **Azure Blob Storage:**
  - ***Storage Tiers***: Offers Hot, Cool, and Archive tiers for cost optimization based on access frequency.
  - ***Simplicity***: Easy to use for general-purpose storage with support for all blob types (Block, Append, Page).
  - ***Access***: Simple hierarchical namespace with basic directory-like structures.

- **Azure Data Lake Storage Gen2:**
  - ***Hierarchical Namespace:*** Provides a true hierarchical file system (folders and directories) on top of Blob Storage, enabling more efficient management of large datasets.
  - ***Hadoop Compatibility:*** Fully compatible with Hadoop Distributed File System (HDFS), allowing it to work seamlessly with big data processing tools.
  - ***Performance:*** Optimized for high-throughput analytics workloads.

Azure Functions

Grow Data Skills

Definition: Event-driven, serverless compute services that execute code in response to various triggers (e.g., HTTP requests, timers, or messages from other Azure services).

Trigger and Bindings:

- Triggers: Define how a function is invoked (e.g., HTTP, Timer, Queue, Blob).
- Bindings: Simplify the integration with other services by automatically handling input/output data (e.g., connecting to databases, storage).

Benefits:
- Serverless Architecture: No need to manage servers; focus solely on code.
- Scalability: Automatically scales out based on the number of incoming events.
- Cost-Efficiency: Pay only for the execution time of the function, making it ideal for variable workloads.
- Language Support: Supports multiple programming languages like C#, JavaScript, Python, Java, and PowerShell.

Azure Function App

Definition: A logical container that hosts one or more Azure Functions, providing an environment to manage, deploy, and scale these functions.

Key Responsibilities:

- Configuration Management: Centralized management of settings like environment variables, connection strings, and keys.
- Scaling and Load Balancing: Automatically scales based on demand, managing the underlying infrastructure transparently.
- Deployment Slots: Supports multiple deployment slots (e.g., staging, production) for testing and deployment.