CLOUD ARCHITECTURE

-Maulin Vyas

Table of Contents

1.Abstract

Cloud architecture adoption has moved from a trend to a strategic need in today's corporate environment. The demand for scalability, agility, and data-driven decision-making is what is driving this paradigm change. Cloud architecture offers disruptive capabilities to organisations and is provided by industry heavyweights such as Microsoft Azure, Google Cloud Platform (GCP), and Amazon Web Services (AWS). The necessity of adopting cloud architecture in the current business climate is examined in this article, with an emphasis on the advantages it offers, including increased scalability, cost-effectiveness, global reach, and security.

The article explores the core steps that go into building a reliable cloud architecture, such as locating data sources, pipeline tactics, DevOps integration, and failure management techniques. It specifically looks at the case of using the Azure ecosystem to build a cloud architecture for a retail company. The retail company uses both physical locations and an online platform to conduct business, and it has aspirations to expand internationally. This underscores the value of cloud architecture in supporting expansion and worldwide reach.

The data intake techniques, Lakehouse architectural design, and pipeline methodologies covering data ingestion, curation, and aggregation are important elements of the suggested cloud architecture. To accommodate a variety of data sources, the design makes use of Azure Data Factory for batch ingestion and Azure Event Hub for streaming ingestion. The design of Lakehouse integrates layers of raw, curated, and aggregated data to enable effective handling and utilisation of data.

The essay also highlights the importance of DevOps principles and continuous integration/continuous deployment (CI/CD) approaches in optimising software development and deployment procedures in cloud architectures. It emphasises how cloud-native solutions can revolutionise enterprises by fostering creativity, agility, and a competitive edge.

In summary, adopting cloud architecture is a strategic necessity for companies looking to succeed in a cutthroat market, not just a technical achievement. Organisations may leverage data and technology to promote success, encourage creativity, and achieve resilience in the digital age by adopting cloud architecture.

2. Introduction

In today's dynamic corporate environment, when data-driven decision-making, scalability, and agility are paramount, embracing cloud architecture has evolved from a fad to a strategic need. By providing a means for enterprises to use the transformational potential of cloud computing services offered by industry titans like Microsoft Azure, Google Cloud Platform (GCP), and Amazon Web Services (AWS), cloud architecture signifies a paradigm change from traditional computing paradigms.

Fundamentally, cloud architecture serves as a design guide for applications and systems that make use of cloud computing resources. In contrast to the traditional method, which uses locally installed software and hardware on physical servers, cloud architecture makes use of the scalability, flexibility, and dependability that come with cloud services. This change makes a strong case for companies of all sizes, from startups looking to create a future-proof foundation from the ground up to established firms looking to modernise their infrastructure, to embrace cloud computing concepts.

In this piece, we explore the strong arguments for why cloud architecture implementation is essential in today's corporate environment. We examine its several advantages, which include improved scalability, cost-effectiveness, global reach, and security. Additionally, we delineate the fundamental procedures entailed in constructing a resilient cloud architecture, ranging from the identification of sources and techniques for data intake to pipeline tactics, mechanisms for managing failures, and the integration of DevOps.

In the end, we emphasise that cloud architecture is a strategic necessity for businesses looking to prosper in a cutthroat and competitive market, not just a technical advancement. Businesses may achieve unprecedented levels of creativity, productivity, and resilience by integrating cloud architecture deliberately. This will enable them to handle the challenges of the digital age with assurance and flexibility.

We will be understanding how a cloud architecture is formed for a retail business using Azure ecosystem which is provided by Microsoft. It is assumed that the retail business runs its operations through brick-and-mortar stores and e-commerce platform. It is also assumed that the business aims to expand its operations overseas and the cloud architecture is constructed accommodating this vision of business.

3. Scenario: Cloud Architecture

The designing of systems and applications that leverage cloud computing resources is known as Cloud architecture. In contrast to traditional computing where, software and hardware are hosted locally on physical servers; the cloud architecture utilizes the scalability, flexibility, and reliability of cloud computing services. Some of the popular companies providing cloud services are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Now there is a common question of why to implement cloud computing concept in an ongoing business or why it is suggested to adapt a cloud computing concept for a new venture? Following are some factors that answer this question:

•Scalability: Traditional IT infrastructure often struggles to scale resources in response to fluctuating demand. Cloud architecture provides the ability to scale resources up or down quickly and efficiently, allowing organizations to handle varying workloads without over-provisioning or underutilizing resources.

•Flexibility: Cloud architecture enables organizations to adapt to changing business requirements and technological advancements. It provides a flexible framework for deploying and managing applications, allowing for rapid development, deployment, and updates.

•Cost-Efficiency: Cloud architecture offers cost advantages compared to traditional on-premises infrastructure. It eliminates the need for upfront capital expenditures on hardware and allows organizations to pay only for the resources they consume, leading to cost savings and better cost predictability.

•Global Reach: Cloud architecture enables organizations to reach global audiences by leveraging the geographic distribution of cloud providers' data centers. This allows for improved latency, resilience, and compliance with data sovereignty regulations.

•Resilience and High Availability: Cloud architecture designs systems for resilience and high availability by leveraging redundant infrastructure and distributed architectures. This ensures that applications remain accessible even in the event of hardware failures or disasters.

•Security: Cloud architecture addresses security concerns by providing built-in security features and compliance certifications. Cloud providers invest heavily in security measures to protect data, applications, and infrastructure from unauthorized access, data breaches, and other security threats.

•Agility: Cloud architecture enables organizations to innovate and deliver value to customers more quickly. It reduces the time and effort required to provision and manage infrastructure, allowing teams to focus on developing and deploying new features and services.

4. Steps to build Cloud Architecture

The process of building a Cloud Architecture consists of multiple steps and those steps are as follows:

1. Source Identification
2. Methods of data ingestion
3. Designing Lakehouse Architecture
4. Pipeline Strategy
5. Pipeline Failure Strategy
6. DevOps/ (CI/CD)

4.1 Source Identification

Considering a business which is operating via both brick-and-mortar stores and e-commerce platforms following are some sources of data which are identified.

1. Sales Database:

A sales database typically stores information related to transactions, customers, products, and sales activities. It may include data such as sales orders, invoices, payments, customer details, product descriptions, and pricing information.

2. E-commerce Transactions:

E-commerce transactions refer to the online purchase activities conducted on an e-commerce platform. This data includes information about the products purchased, transaction amounts, payment methods, shipping details, and customer demographics.

3. Currency Conversion API:

Considering the business is offering online purchase from USA or has a vision to expand its operations in USA, a currency conversion API is a web service that allows applications to convert one currency to another based on current exchange rates. This data source provides real-time or historical exchange rate information for various currency pairs.

4. Website Tracking Logs:

Website tracking logs capture detailed information about user interactions with a website or online application. This data includes page views, clicks, navigation paths, referral sources, user agents, and other user engagement metrics.

5. Geo-location Data:

Geo-location data refers to information about the geographic location of users or devices. This data can include latitude, longitude, IP addresses, postal codes, and location-based services (LBS) data.

4.2 Methods of data ingestion

There are typically two ways to ingest data from various resources into a Lakehouse and they are Batch Ingestion and Streaming Ingestion.

Batch Ingestion: Batch ingestion is a data processing method used to collect and load large volumes of data into a system or storage repository in discrete batches or chunks. Batch ingestion involves collecting data over a period and processing it in predefined intervals, such as hourly, daily, or weekly. In Azure ecosystem, Azure Data Factory is the tool which is commonly used for batch ingestion.

Streaming Ingestion: Streaming ingestion is a data processing method used to collect, process, and analyze data in real-time as it is generated or received from various sources. Unlike batch ingestion, streaming ingestion handles data continuously and immediately as it becomes available, without waiting for predefined batch windows. In Azure ecosystem, Azure Event Hub is the tool which is commonly used for Streaming ingestion.

For the current scenario, the data can be ingested in batches using Azure Data Factory for data sources sales database, website tracking, currency conversion, and geo-location data. The suggested time interval of this batch ingestion is of 8 hours. Whereas the e-commerce data can be ingested by streaming ingestion technique using Event Hub.

4.3 Designing Lakehouse Architecture

A data Lakehouse is a modern data architecture that combines elements of both data lakes and data warehouses to address the limitations and challenges of traditional data management approaches. It integrates the flexibility and scalability of data lakes with the structured querying and performance optimization capabilities of data warehouses, providing a unified platform for storing, managing, and analyzing diverse data types at scale.

Layers or zones of Lakehouse architecture:

•Raw zone/ Bronze layer: In a data Lakehouse architecture, the "raw zone" refers to a section or layer within the unified storage where raw, unprocessed data is stored in its original format without any transformations or schema enforcement. The raw zone serves as the initial landing area for ingested data before it undergoes further processing and refinement.

•Curation zone/ Silver layer: In a data Lakehouse architecture, the "curation zone" refers to a section or layer within the unified storage where data is processed, refined, and curated to improve its quality, consistency, and usability for analytics and consumption. The curation zone serves as an intermediate layer between the raw zone and the aggregated zone.

•Aggregation zone/ Gold layer: In a data Lakehouse architecture, the "aggregation zone" refers to a section or layer within the unified storage where precomputed aggregations and summaries of data are stored. The aggregation zone, which is the last zone of all, serves as a repository for aggregated data that has been processed and summarized to accelerate query performance and optimize analytics workflows.

4.4 Pipeline Strategy

A data pipeline is a series of processes and tools used to ingest, process, transform, and analyze data as it moves from its source to its destination. It facilitates the flow of data from diverse sources to data storage, processing, and analytics systems, enabling organizations to derive insights and make data-driven decisions.

We will be using three types of pipelines in this cloud architecture which are:

•Data Ingestion pipeline: A data ingestion pipeline is a specific type of data pipeline that focuses on the process of collecting and ingesting data from various sources into a centralized storage or processing system. It involves extracting data from source systems, transforming it into a standardized format (if necessary), and loading it into a destination system for further processing, analysis, or storage.

•Data Curation Pipeline: A data curation pipeline is a specific type of data pipeline focused on the process of refining, enriching, and preparing raw data for analysis and consumption. It involves a series of steps to improve the quality, consistency, usability of data, normalising, removing null values, removing bad characters, and standardizing data before it is used for analytical or operational purposes.

•Data Aggregation Pipelines: Data aggregation pipelines are a specific type of data pipeline designed to consolidate and summarize data from multiple sources into aggregated datasets. These pipelines are essential for generating high-level insights from large volumes of data and optimizing query performance for analytical use cases.

Parent Child Pipeline:

In workflow orchestration and dependency management, parent-child pipelines—also called hierarchical pipelines—are a part of larger data processing or ETL (Extract, Transform, Load) workflows. This architecture arranges and manages several smaller pipelines, called child pipelines, under the supervision of a parent pipeline, which is a higher-level pipeline.

Parent Pipeline: One or more child pipelines are executed by the parent pipeline, which is the overall process that coordinates and controls their execution. It outlines the dependencies, coordination, and general order of execution for the child pipelines. Typically, the parent pipeline manages the entire workflow's scheduling, monitoring, error handling, and logging.

Child Pipelines: These are discrete activities or workflows that carry out certain data transformation, processing, or analysis operations. Every child pipeline is intended to complete a particular activity or collection of tasks and constitutes a portion of the total workflow. Child pipelines can be shared across several parent pipelines or processes since they can be made modular and reusable.

4.5 Pipeline Failure Strategy

Any circumstance or occurrence that prevents a pipeline system from operating normally is referred to as a pipeline failure. This term is commonly used in relation to software development, data processing, and workflow automation. In these situations, a pipeline is a set of linked steps or activities intended to carry out a certain operation, including assembling code, processing data, or distributing software.

Following are some of the strategies that can be implement in case of a pipeline failure:

•Timeout: The term "timeout" in the context of a pipeline designates the maximum duration allocated for the completion of a task or set of actions inside the pipeline. Timeout methods are frequently used to stop jobs from continuing endlessly, particularly when there's a chance a process could become stuck or run into an issue that keeps it from finishing.

•Retry: When a pipeline encounters an error or failure, "retry" describes the procedure whereby it automatically tries to run a failed task again. Retry methods are frequently used in pipelines to boost resilience and raise the possibility of a task being completed successfully, particularly in the event of momentary problems or failures. Setting the value to 3 attempts of retry is a good practice.

•Alert mechanism: The alert features of a pipeline failure are the systems designed to alert pertinent parties in the event of a pipeline failure. These notifications aid in prompt issue awareness, facilitating quick action and resolution. Common alert channels include email, SMS, instant messaging platforms (such as Slack or Microsoft Teams), and dedicated alerting systems (such as PagerDuty or OpsGenie).

4.6 DevOps/ (CI/CD)

Software developers employ a set of procedures and principles called continuous integration/continuous deployment, or CI/CD, to automate the building, testing, and releasing of code changes. It entails the automated distribution or deployment of code modifications into production settings (Continuous Deployment) and the integration of those changes into a shared repository (Continuous Integration). When CI/CD is used in relation to pipelines, it usually means that these practices are being applied inside a software development pipeline. When CI/CD is included into a pipeline, it streamlines the processes of developing, testing, and releasing software, enabling the timely and dependable delivery of applications. This is an explanation of CI/CD within a pipeline:

Continuous Integration (CI)

•Several times a day or even more frequently, CI entails integrating code changes from several developers into a shared repository.

•Every integration sets off automated build procedures, which compile, test, and confirm the accuracy of the code.

•Early in the development cycle, continuous integration (CI) helps find conflicts, integration problems, or defects, facilitating quicker feedback and resolution.

Continuous Deployment (CD):

•After successful testing and verification, code changes are automatically deployed to production environments. This approach advances Continuous Delivery.

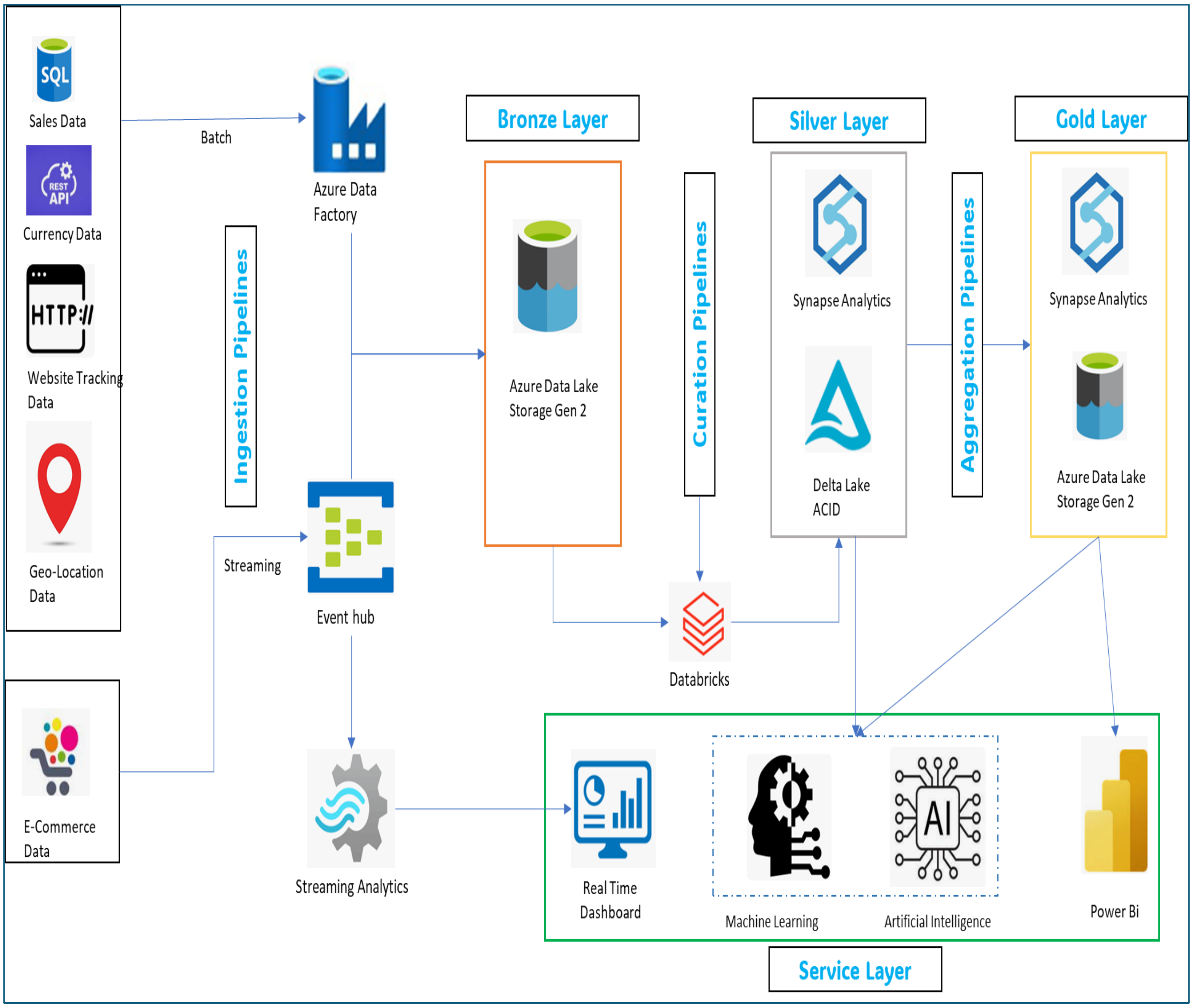
•Every code update that passes the automated testing is automatically deployed into production using continuous deployment, eliminating the need for human intervention.

•To guarantee the stability and dependability of production environments, Continuous Deployment necessitates a high degree of trust in the automated testing and deployment processes.

DevOps: it is a similar practice to CI/CD where the goal of the DevOps set of procedures, cultural tenets, and technical resources is to enhance coordination and dialogue between teams working on software development (Dev) and IT operations (Ops). It aims to automate repetitive chores, expedite the software delivery process, and promote an innovative and continuous improvement culture within businesses. The integration of development, testing, deployment, and operations processes is emphasised by DevOps in order to produce high-quality software quickly and effectively. Integrating procedures, equipment, and cultural values that support cooperation, automation, and continuous improvement throughout the software development lifecycle is necessary to implement DevOps principles into a pipeline.

5. Cloud Architecture

The cloud architecture presented below is for retail business, considering its operations in both brick-and-mortar stores and e-commerce platforms. It is also assumed that the business aims at expanding its business overseas via e-commerce website and API. The cloud architecture is built within Azure ecosystem, which is the cloud service provided by Microsoft.



Cloud Architecture Diagram

As shown in the diagram, there are five data sources i.e. sales data, currency data, web-site tracking data, geo-location data, and e-commerce data. The data flow is starting from data source and ending at data consumption at service layer. The first step is to ingest data from various sources mentioned earlier into a storage space. The ingestion technique used is batch ingestion and streaming ingestion. Azure data factory is preferred option for batch ingestion and Event Hub is the preferred option for streaming ingestion. Ingestion pipelines are used here to ingest data from data sources to bronze layer of Lakehouse. There is no transformation of data taking place in this step.

For streaming data there are two approaches displayed in the architecture, one is using the real time data directly coming from event hub to display real time dashboards via streaming analytics and second one is to store the data that is being collected from event hub. Traditionally streaming data was not stored as it was only used for dashboarding purposes but in recent years with the development of ML and AI, predictive modeling capabilities have been developed by working on the stream data that is stored.

The second step is to transform the stored data in bronze layer by normalizing, removing null values, bad characters, and standardizing data into schemas into silver layer of Lakehouse. The Silver layer consists of synapse analytics where transformation takes place with help of curation pipelines and transforming data using Databricks. Implementing Delta Lake storage is a key feature of silver layer. The main reason for this is to accommodate ACID transactions. ACID stands for Atomicity, Consistency, Isolation, and Durability.

When combined, these characteristics guarantee the consistent and reliable processing of database transactions, which lays the groundwork for data consistency and integrity in database systems. Applications where data integrity is crucial, such banking systems, e-commerce platforms, and other mission-critical applications, must adhere to ACID standards.

The third step is to get data from silver layer and aggregate that into more refined form of data and store it in gold layer of Lakehouse. Aggregation pipelines are used here to aggregate data from silver layer and make it available for service layer ahead. Here, again synapse analytics will be used to do aggregation functions and data lake storage gen 2 will be used for storing the data simultaneously.

In the last step there is the service layer which has multiple consumers like power bi users, data marts, Machine learning, and Artificial Intelligence. A data mart is an organised subset of a data warehouse that is intended to serve a particular department or functional area inside a company. It is intended to manage and maintain data pertinent to a certain user group’s need, like a business unit or departmental team. Data marts will be given access only to the gold layer of Lakehouse. Whereas the ML and AI teams will be given access to both gold and silver layer, new algorithms require designing of new aggregate pipelines hence access to silver layer is important.

6. Conclusion

To sum up, implementing cloud architecture becomes an obvious must for contemporary companies looking to maximise scalability, streamline processes, and gain access to data-driven insights. Organisations can reap numerous benefits by utilising the built-in advantages of cloud computing services provided by companies such as Microsoft Azure, Google Cloud Platform (GCP), and Amazon Web Services (AWS).

The importance of cloud architecture becomes clear when one looks at how business requirements and technology are changing over time. In comparison to traditional on-premises infrastructure, its scalability, flexibility, and dependability enable enterprises to effectively handle varying workloads, react to changing requirements, and realise cost efficiencies.

Additionally, cloud architecture helps businesses to reach a wider audience internationally, guarantee high availability and resilience, handle security issues, and promote innovation and agility in value delivery. Cloud-native solutions' capacity to automate procedures and dynamically modify resources makes it easier to build, deploy, and update quickly, which eventually gives businesses a competitive edge in the fast-paced market of today.

Cloud architecture must be strategically implemented, which requires careful preparation and execution of several important phases, including pipeline strategies, failure management mechanisms, source identification, data ingestion techniques, Lakehouse architectural design, and DevOps integration. Every element is essential to building a strong base for data processing, administration, and consumption, which results in insights that are useful for making decisions and expanding the organisation.

Essentially, cloud architecture is a revolutionary way to utilising data and technology to drive organisational success, rather than merely a change in technology. Adopting cloud architecture is essential for creativity, efficiency, and resilience in the dynamic world of modern business, especially as companies continue to embrace digital transformation.