Cloud Architecture for Hosting Local Ltda.

Table of Contents

1. Introduction

The objective of this project was to design a Cloud Architecture for Hosting Local Ltda., an Airbnb-like platform. The platform focuses on short-term rentals and serves as a connection between local hosts and travelers worldwide.

2. Problem statement

Hosting Local Ltda. on-premises infrastructure poses significant limitations on processing capacity due to the company's rapid growth and peak usage periods, leading to system failures and decreased performance. Additionally, networking issues compromise data availability, impacting revenue as customers may switch to competitors' platforms. Moreover, limited data access is affecting operational efficiency and customer trust in the company.

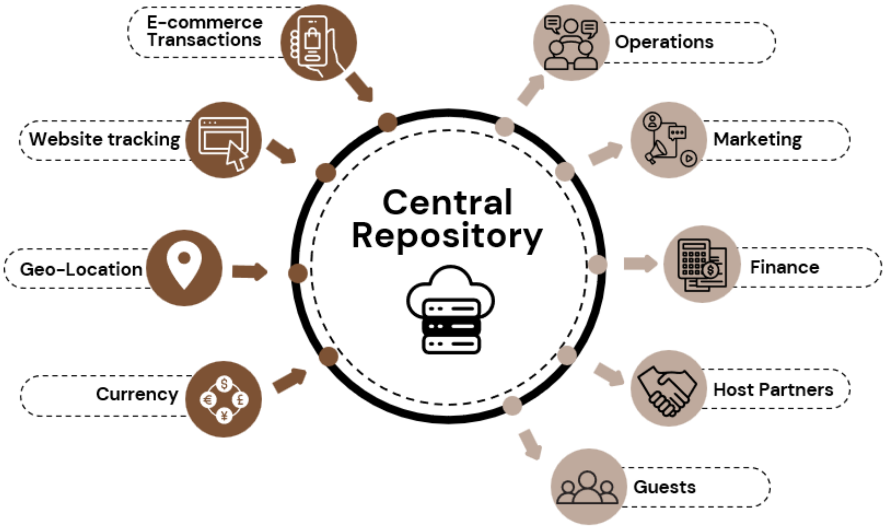
3. Objectives

The objectives of Hosting Local Ltda. to migrate to the cloud are:

1. Leverage cloud-based solutions to solve performance issues during peak periods and mitigate the risk of system failures.
2. Improve resource allocation through scalable cloud resources instead of investing resources in on-premises solutions.
3. Maximize revenue through improved data availability, reducing potential revenue loss, and optimizing operational efficiency.
4. Enhance customer trust and satisfaction by providing a seamless user experience, reducing disruptions and performance issues.

4. Architecture Vision

Leveraging the cloud enables better resource allocation by scaling up and down according to demand. Data from e-commerce transactions, website tracking, geolocation, and currency will be ingested, processed, cleaned, and prepared for consumption by end users. The architecture needs to ensure the availability of data for users, including operations, marketing, finance, host partners, and guests. The figure below illustrates the vision for this architecture, depicting the data sources and end users involved.



4.1. Data Sources

E-commerce Transactions: include booking records, cancellations, real-time availability calendars, user and host registration, reviews, and ratings. These data types are crucial for providing a seamless user experience. They enable users to check availability calendars, make bookings and cancellations, and update their accounts without delays.

Website Tracking: captures user interactions with the website such as which pages they visit, how much time they spend on each page, their navigation patterns, and purchase conversion metrics. This data is important to implement marketing strategies based on user activity and preference. Additionally, website tracking enables improving website performance based on user experience.

Currency: facilitates pricing updates on the website. Exchange rates can fluctuate significantly within a month. It is important to keep it updated so users can visualize the price based on the actual exchange rate. Real-time updating of exchange rates might lead to customer dissatisfaction and confusion if the prices change while they are browsing. Having a stable price throughout the day is more customer-friendly and simplifies financial analysis.

Geolocation: provides insights into user locations and regional preferences. Understanding regional preferences and trends allows businesses to target ads and promotions effectively. Real-time processing is not essential for this task as it would increase computing costs unnecessarily.

4.2. End Users

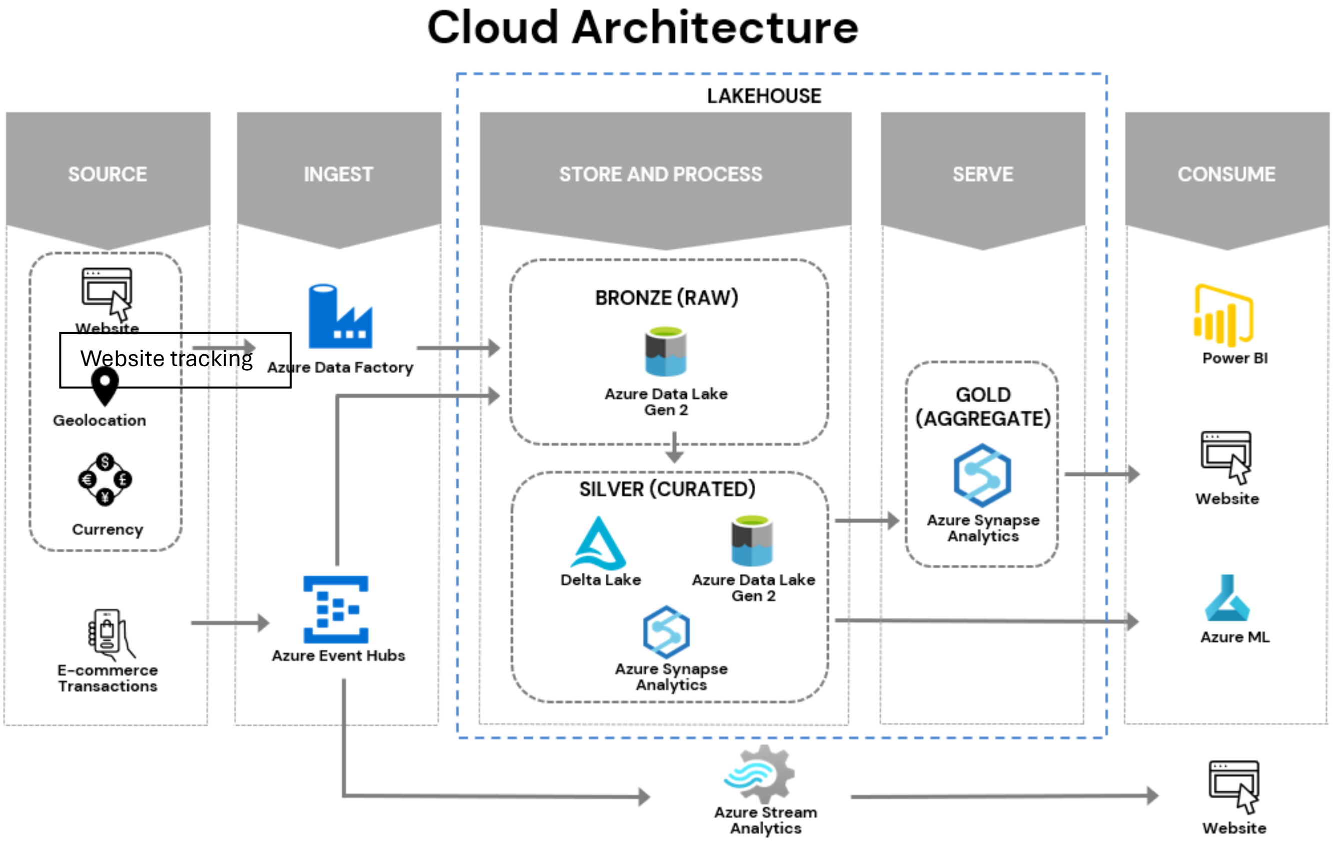
End users will utilize the data for various purposes, such as:

- **Operations:** manage website functionalities, analyze customer service performance, and identify fraudulent hosts.
- **Marketing:** targeted marketing campaigns, analyze user preferences, provide personalized recommendations, and suggest optimized pricing strategies for guests.
- **Finance:** financial analysis, revenue forecast, and managing payouts to hosts.

- Host Partners: review guest feedback, track bookings and manage property listing.
- **Guests:** browse listings, make bookings, manage reservations, and provide reviews.

5. Cloud Architecture

The architecture below illustrates the data flow within the cloud, from the source to the consume stage.



5.1. Batch Ingestion

Data ingestion is conducted through two different methods and distinct services. Azure Data Factory handles batch ingestions of website tracking, geolocation, and currency every 24 hours, while Event Hubs manage data streaming for e-commerce. Notice that Event hubs also save the streaming data in the Bronze Layer for further analysis.

Bronze Layer: Raw data is stored in Azure Data Lake Storage Gen 2. The raw website tracking, currency exchange rates, geolocation, and e-commerce data are stored here in their original format without any transformation. This layer acts as a landing zone for all incoming data. Only reading is allowed at this stage.

Silver Layer: Curated and processed data are stored in this layer and business rules are established. Data Lake Storage Gen 2 is used to store unstructured data such as customer reviews and website tracking, intermediate datasets, or derived tables, while Synapse Analytics and Delta Lake are used for structured data such as currency and e-commerce transactions, respectively.

The data from the Bronze layer is first validated and cleaned to ensure data quality and accuracy. This process may involve checking for missing values, correcting data formats, and removing duplicates or anomalies. Additionally, Azure Synapse Analytics can be used to combine datasets for advanced analysis and modeling. For example, exchange rates can be utilized to analyze e-commerce transactions, and geolocation data can be combined with website tracking data for further user analysis.

In the silver layer, Delta Lake is employed for its schema evolution capabilities, allowing schema modification without disrupting data pipelines or existing data.

Gold Layer: The curated data in the Silver layer serves as the foundation for generating aggregated insights in the Gold layer. Within the Gold layer, Azure Synapse Analytics is utilized as a primary data repository, storing structured and aggregated data sets that will serve as the foundation for various analytical purposes.

This layer contains data marts which are specialized data repositories for specific business departments or teams. Data marts may be created for departments such as marketing, finance, and operations, each containing aggregated data sets relevant to their respective objectives.

Consume: The Consumption phase is the final step in the data transformation journey where raw data is transformed into actionable insights in the decision-making process.

Power BI connects with the Gold layer to access aggregated data sets stored in Azure Synapse Analytics. Power BI can create dashboards to show revenue trends and KPI performance, track goals and identify areas for improvement in a user-friendly way.

Azure Machine Learning utilizes the Silver layer for cognitive analytics because the silver layer maintains a level of detail and granularity that is necessary for training machine learning models. The aggregation of data in the Gold layer is good for looking at overall trends and making reports but it might lose important details that were in the original data. Machine Learning models can predict customer behavior and automate recommendations to customers based on previous searches, or based on historical website tracking data, it can also suggest optimized pricing strategies for hosts based on similar listings in the same region.

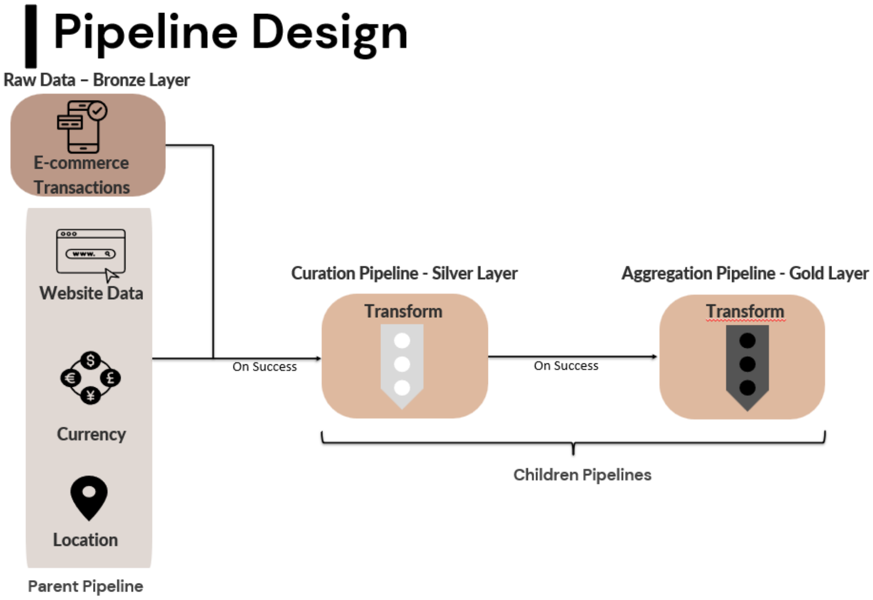
The Hosting Local Ltda. website can connect with the Gold Layer and Power BI, and Azure Machine Learning to provide valuable insights for hosts and users, such as analytics on booking trends, occupancy rates, suggestions on rates to optimize host revenue, and suggestions for users based on their preferences.

5.2. Streaming Ingestion

Returning to the ingestion of streaming data, Within the architecture, two distinct paths are established for managing streaming data. On the first path, as data is ingested, Azure Stream Analytics processes and analyzes the streaming data in real-time. Stream Analytics allows filtering, aggregating, and transforming the data which can be transferred directly to the website for real-time view. On the second path, data ingested via Event Hubs is directed to the bronze layer for storage. Once situated within the bronze layer, the data passes through the same processing layers as batch data.

6. Pipeline Design

The data pipeline architecture involves data flow from parent pipelines in the Bronze layer to child pipelines in the silver and gold layers. For example, a parent pipeline first ingests data from all the data sources into the Bronze layer, followed by a child pipeline that cleans and processes the data for the Silver layer. From the silver layer, another child pipeline is triggered for the aggregated data.



The figure above shows a dependency between each layer, which means that the next pipeline will only run if the previous pipeline succeeds. A sequence of activities is set and connections are established between them to orchestrate the flow of data in a cohesive sequence. Subsequent activities are dependent on previous activities to be executed. For example: if the data ingestion is completed successfully it triggers the curation pipeline.

An Event-based Trigger is activated after successful file ingestion into the bronze layer. This means that when files from website tracking, currency, and geolocation are successfully ingested into the bronze layer, it triggers the silver pipeline. The silver pipeline is primarily responsible for curating and refining data. Once the data undergoes successful processing and transformation within the silver layer, another event-based trigger starts the gold pipeline. The gold pipeline then takes over to aggregate and further refine the curated data, preparing it for consumption and visualization by end users.

7. Monitoring the Pipeline:

Azure Synapse Analytics will be used to monitor the pipeline process. All of the activities contained in a pipeline can be monitored. Monitoring involves tracking and analyzing various aspects of the pipeline's performance and execution at different stages. Examples of monitoring metrics include: data ingestion rates, number of successful and failed records, pipeline start and end times, latency, status (success/failure), data quality, and any error message due to failure.

Custom monitoring dashboards can be created within Azure Monitor by aggregating key metrics from different Azure services involved in the pipeline. These dashboards provide a better view of pipeline performance and facilitate optimization in the pipeline configurations, improving data quality and pipeline performance.

8. Addressing failures:

In case of failure, automatic actions need to be implemented promptly to address the issues efficiently.

Messages: Synapse Analytics can be configured to send emails or messages on Teams based on predefined conditions. In case of failure, deviation in performance, or even if the pipeline is successful a specific message can be triggered to alert the responsible team as needed.

Timeout: the duration for each activity to complete successfully under normal circumstances is estimated and the timeout is set to allow sufficient time for activities to complete their tasks, while also preventing activities from running for a long time and increasing computing costs. Activities execution times can be monitored to adjust timeout as needed.

Retry intervals: if a pipeline activity fails, the system will be configured to automatically retry the operation to ensure data continuity. The first retry interval will start with a shorter interval of 5 minutes. If the first retry isn’t successful, the system applies an incremental interval by doubling the first time to 10 minutes. Similarly, for the third retry, the time interval is doubled again to 20 minutes. The incremental time between each retry gives the system time to recover from temporary issues. The system will be set to attempt retries 3 times. After the third retry manual investigation is necessary to resolve the issue.

Rerun the entire Pipeline Automatically: if the ingestion phase fails due to Azure service outages or network connectivity issues, Azure Data Factory will be set to rerun the entire pipeline.

Rerun the Failed Activity Automatically: if the fail is not related to network or Azure services outages and not related to the ingestion process, only the failed activity will be rerun. A conditional execution to identify failed activities and trigger automated retries for the specific activity needs to be configured.

Terminate the pipeline automatically: based on monitoring activities in Azure Monitor, a trigger is activated if data quality is compromised, which automatically terminates the pipeline to avoid further propagation of corrupted data. The cause of the issue is investigated and corrective actions are initiated.