INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Experimentação de Plataformas de Dados & AI na Cloud

JOÃO RICARDO CIDRA FIGUEIREDO
Junho de 2025

POLITÉCNICO
DO PORTO

ISEP INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

Experimenting with Data Platforms & AI in the Cloud

João Figueiredo

A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Information and Knowledge Systems

Advisor: Paulo Oliveira
Supervisor: Pedro Neves

Porto, June 29, 2025

Statement of Integrity

I hereby declare that I have conducted this academic work with integrity.

I have not plagiarised or applied any form of undue use of information or falsification of results along the process leading to its elaboration.

Therefore the work presented in this document is original and authored by me, having not previously been used for any other end. The exceptions are explicitly recognised in the section "Ethical considerations" of the first chapter. This section also states how AI tools were used and for what purpose.

I further declare that I have fully acknowledged the Code of Ethical Conduct of P.PORTO.

ISEP, Porto, June 29, 2025

Dedicatory

To procrastination — without you, this would have been done sooner, but way less interesting.

Abstract

Cloud-based data and AI platforms have become critical for organizations seeking scalable, real-time analytics and operational efficiency. This thesis investigates the capabilities of leading providers—AWS, Google Cloud, and Microsoft Azure—focusing on their suitability for large-scale data engineering and machine learning workloads in complex domains such as telecommunications.

Conducted within the context of Altice Labs, the research includes a hands-on implementation of a cloud-native data migration and analytics pipeline. The SIGO system was rearchitected using AWS services, replacing legacy on-premise processes with scalable, serverless components like AWS Glue, S3, Athena, and QuickSight. The migration delivered measurable improvements in ETL runtime, query latency, system reliability, and cost efficiency.

Beyond performance, this work also explores ethical dimensions of cloud adoption—emphasizing data governance, access control, and compliance in multi-tenant environments. The project demonstrates how a cloud-first approach can empower organizations to operationalize AI, democratize insights, and respond faster to evolving service needs.

By combining platform evaluation, practical deployment, and organizational impact, this thesis offers a comprehensive view of what it takes to modernize data infrastructure in the age of AI.

Keywords: Data, Cloud, Artificial Intelligence, AWS

Resumo

As plataformas de dados e IA baseadas na nuvem tornaram-se críticas para as organizações que procuram análises escaláveis e em tempo real e eficiência operacional. Esta tese investiga as capacidades dos principais fornecedores - AWS, Google Cloud e Microsoft Azure - centrando-se na sua adequação a cargas de trabalho de engenharia de dados em grande escala e de aprendizagem automática em domínios complexos como as telecomunicações.

Conduzida no contexto da Altice Labs, a investigação inclui uma implementação prática de um pipeline de migração e análise de dados nativos da nuvem. O sistema SIGO foi rearquitetado usando serviços AWS, substituindo processos legados no local por componentes escaláveis e sem servidor, como AWS Glue, S3, Athena e QuickSight. A migração proporcionou melhorias mensuráveis no tempo de execução do ETL, latência de consulta, fiabilidade do sistema e eficiência de custos.

Além do desempenho, este trabalho também explora as dimensões éticas da adoção da nuvem - enfatizando a governança de dados, o controle de acesso e a conformidade em ambientes multi-locatários. O projeto demonstra como uma abordagem que prioriza a nuvem pode capacitar as organizações a operacionalizar a IA, democratizar os insights e responder mais rapidamente às necessidades de serviço em evolução.

Ao combinar a avaliação da plataforma, a implantação prática e o impacto organizacional, esta dissertação oferece uma visão abrangente do que é necessário para modernizar a infraestrutura de dados na era da IA.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my academic supervisor, Professor Paulo Oliveira, for his invaluable guidance, insightful feedback, and continuous support throughout the development of this dissertation. His availability, encouragement, and thoughtful advice greatly improved the quality of my work and my experience during this academic journey.

I would also like to thank my friends, whose companionship and encouragement provided motivation and much-needed balance during challenging times. Finally, I am immensely grateful to my family for their unwavering love, patience, and support throughout my studies — this achievement would not have been possible without them.

Contents

List of Figures

List of Tables

List of Acronyms

ACM	Association for Computing Machinery.
AI	Artificial Intelligence.
API	Application Programming Interface.
AWS	Aamzon Web Services.
CCPA	California Consumer Privacy Act.
CI/CD	Continuous Integration / Continuous Deployment.
ETL	Extract, Transform, Load.
GCP	Google Cloud Platform.
GDPR	General Data Protection Regulation.
IAM	Identity Access Management.
IEEE	Institute of Electrical and Electronics Engineers.
ML	Machine Learning.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
SQL	Structured Query Language.
VPC	Virtual Private Cloud.
WBS	Work Breakdown Structure.

Chapter 1

Introduction

1.1 Context

Telecommunications is naturally a data-intensive industry, replete with operational and user data streaming from the networks. This ranges from network performance monitoring to personalized service delivery, including predictive maintenance. The volume and complexity of these datasets necessitate platforms that can process and analyze data in real time and integrate seamlessly with existing systems. Cloud-based data platforms are leading this charge in the fulfillment of these demands. To this end, scalability empowers organizations to have dynamic resource allocation against fluctuating workloads; advanced analytics opens up new doors toward innovation. Besides, they enable collaboration on data and allow access remotely, which is extremely important for work environments dispersed all over the globe nowadays.

The last couple of years have seen an explosion in the generation of data and, consequently, in the complexity of operations on data. How to efficiently manage, integrate, and analyze vast volumes of data coming from diverse sources has become very crucial for organizations across different industry verticals. This challenge is particularly acute in the telecommunications sector, where real-time decision-making, network optimization, and customer experience enhancement rely on timely and accurate data insights.

The key to this, in large part, has come in the form of data platforms and AI services that companies such as Google Cloud Platform, Amazon Web Services, and Microsoft Azure provide. The reasons include: the provision of scalable, secure, and cost-effective solutions; ingesting, storing, and analyzing big data sets in real time. Challenges do persist. Organizations are faced with challenges of complexity in data migration, integration of heterogeneous data sources, cost management, and compliance with strict data protection regulations.

1.2 Problem description

Cloud-based data platforms are part of Altice Labs' business strategy; it is only necessary to sustain business management-one of the most innovative leaders within the telecommunication business area. These realities push companies to know about and experience the technologies. If those can indeed manage data traffic that keeps on increasing, due to telecommunication networks, in a precise imperatively required judgment by Altice Labs.

However, the transition to these is not without issues. Integrations with cloud-based solutions on-premise always have inherent problems and require carefully laid-out planning for seamless operability. But most importantly, it needs consideration whether these systems

can cater effectively to the key needs of such a telecommunications networks, which do include low latency processing, good security, and various other sector regulations. Moreover, such platforms should have their cost-benefit ratio optimized to bring value into the organization in a measurable way.

Having these challenges handled through proper structuring will let Altice Labs unlock the full potential of cloud-based data platforms and increase its data management capabilities in order to keep its leading position in the telecommunications industry.

1.3 Objectives

The project aims to compare the functionalities of these platforms, such as (Google Cloud Platform (GCP)), Azure and Aamzon Web Services (AWS), testing their capabilities in real-world scenarios, such as integrating and migrating data from various sources. This experimentation will help to identify the strengths and weaknesses of each platform in handling specific telecom-related data challenges, contributing to better-informed decisions on future technological investments.

Further exploration will be guided by existing literature on cloud data platforms and Artificial Intelligence (AI) services, such as, which discusses their application in telecommunications, and, which compares different providers in terms of integration and analytics efficiency.

More specifically, the objectives of this project are:

1. Experiment with leading data and AI platforms in the cloud from AWS, Google, and Microsoft
2. Compare their functionalities, advantages, and disadvantages in data handling, processing, and analysis
3. Identify and implement use cases for Altice Labs using these platforms
4. Develop competencies in cloud-based data management and AI applications

1.4 Contributions

This thesis makes the following contributions to the study and application of cloud-based data platforms for AI and data in the telecommunications industry:

- Comprehensive Analysis of Cloud-Based Data Platforms
  - Provides an in-depth evaluation of major cloud platforms (GCP, AWS, Microsoft Azure) and their capabilities in data management, integration, and advanced analytics for telecommunications-specific use cases. Identifies the strengths and limitations of each platform with respect to scalability, cost-efficiency, and support for AI-driven solutions and their performance.
- Framework for Platform Evaluation and Comparison
  - Proposes a structured framework to evaluate the suitability of cloud platforms based on critical criteria such as data volume handling, latency, compliance, and interoperability with existing on-premises systems. Offers practical benchmarks and metrics that can guide future decision-making processes in selecting data platforms for AI applications.

- Implementation and Experimentation
  - Demonstrates the implementation of a proof-of-concept using a selected cloud platform to ingest, process, and analyze telecommunications data. Evaluates the platform's performance in real-world scenarios, focusing on key metrics such as speed, accuracy, resource consumption, and cost.
- Optimization Strategies for Telecommunications Use Cases
  - Develops optimization strategies for data workflows, including data ingestion, transformation, and analysis, tailored to the unique needs of telecommunications networks. Explores how cloud-native AI tools (e.g., automated machine learning pipelines, predictive analytics) can enhance network performance and customer insights.
- Insights into Hybrid and Multicloud Architectures
  - Investigates the challenges and potential solutions for integrating cloud-based platforms with legacy on-premises systems. Explores the feasibility of multicloud strategies to balance cost, performance, and resilience.
- Guidance on Cost and Compliance
  - Offers actionable recommendations for controlling costs associated with cloud platform adoption while ensuring compliance with data protection regulations such as GDPR.
- Strategic Recommendations for Altice Labs
  - Provides Altice Labs with a roadmap for adopting cloud-based data platforms, including key considerations for implementation, scaling, and long-term value realization.

1.5 Ethical Considerations

The use of cloud-based data platforms and AI in telecommunications raises critical ethical considerations that need to be debated for its responsible and equitable application. First and foremost, the volume and sensitivity of user information call for the handling of data privacy and security by a telecom provider. Compliances to strict regulations like General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA) are not only legal imperatives but also form the basis of user trust. Companies should make sure that data is kept secure by deploying robust encryption, access controls, and auditing mechanisms that prevent data breaches and unauthorized use.

Another critical challenge is bias reduction within AI models. Most of the AI models are built using historical data, which may inadvertently carry or magnify certain biases from the past. The presence of such biases will result in discriminatory practices against customers while segmenting them, provisioning services, or even detecting fraud. Fairness in data-driven decisions can be ensured only through extensive audits of AI training datasets, bias detection frameworks, and the inclusion of diverse perspectives during model development. Furthermore, transparency in AI decision-making processes will increase accountability.

The ecological footprint of cloud computing cannot go unnoticed. Data centers powering the cloud are great consumers of energy and result in carbon dioxide emissions. The promotion

of sustainability by optimizing server efficiency, shifting to renewable energy sources, and adopting energy-efficient coding practices becomes very important in reducing ecological footprints of the sector. Green computing can make technological advancements align with larger environmental objectives.

The last point relates to implications for the workforce: challenges and opportunities. While automation driven by AI can streamline operations and improve efficiency, it might also displace some job roles. The organizations are supposed to invest in workforce development through retraining and upskilling programs for workers who may lose their jobs to automation. This proactive step could help workers evolve into higher-value roles, which fosters a culture of continuous learning and adaptation.

Such diversified challenges are addressed in this research in its quest to ensure the responsible and ethical use of cloud-based platforms in telecommunications for the benefit of not only the organizations but also society as a whole.

1.6 Project Planning

This section outlines the project planning process, including key tasks, timelines, and milestones. A Gantt chart is presented to visually represent the schedule and ensure efficient management of the project's progress.

Figure 1.1 illustrates the initial phase of research and planning. Figure 1.2 represents the skill-building and learning phase. Figure 1.3 highlights the core implementation and migration efforts. Finally, Figure 1.4 summarizes the thesis writing and documentation phase.

DIMEI	**93 dias**	**Seg 24/02/25**	**Qua 02/07/25**	
Phase 1	**25 dias**	**Seg 24/02/25**	**Sex 28/03/25**	
Define research scope and	5 dias	Seg 24/02/25	Sex 28/02/25	
Identify cloud platforms to evaluate (AWS, Azure, GCP,	5 dias	Seg 03/03/25	Sex 07/03/25	3
Collect resources and documentation	3 dias	Seg 10/03/25	Qua 12/03/25	4
Compare features and pricing models	4 dias	Qui 13/03/25	Ter 18/03/25	5
Evaluate suitability for data engineering use	4 dias	Qua 19/03/25	Seg 24/03/25	6
Document findings and analysis	4 dias	Ter 25/03/25	Sex 28/03/25	7

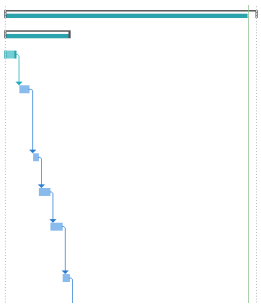


Figure 1.1: Gantt chart showing project timeline and task dependencies.

Phase 2	**20 dias**	**Seg 31/03/25**	**Sex 25/04/25**	**8**
Set Up AWS Environment	4 dias	Seg 31/03/25	Qui 03/04/25	
Learn Core AWS Services	4 dias	Sex 04/04/25	Qua 09/04/25	10
Learn AWS Data Engineering	4 dias	Qui 10/04/25	Ter 15/04/25	11
Build and Test a Sample Data	4 dias	Qua 16/04/25	Seg 21/04/25	12
Implement Security and Cost Management	4 dias	Ter 22/04/25	Sex 25/04/25	13

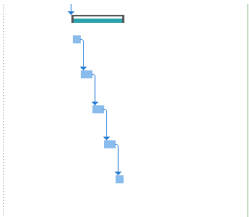


Figure 1.2: Linear timeline highlighting major project phases.

Phase 3	**27 dias**	**Sex 25/04/25**	**Seg 02/06/25**	
Assess existing on-prem workload	3 dias	Sex 25/04/25	Ter 29/04/25	
Design architecture for AWS (data pipelines, storage,	4 dias	Qua 30/04/25	Seg 05/05/25	16
Set up AWS environment	5 dias	Ter 06/05/25	Seg 12/05/25	17
Develop ETL/data pipelines (Glue, Lambda, Step	2 dias	Ter 13/05/25	Qua 14/05/25	18
Migrate data and validate	5 dias	Qui 15/05/25	Qua 21/05/25	19
Monitor and optimize workloads	5 dias	Qui 22/05/25	Qua 28/05/25	20
Conduct performance and	3 dias	Qui 29/05/25	Seg 02/06/25	21

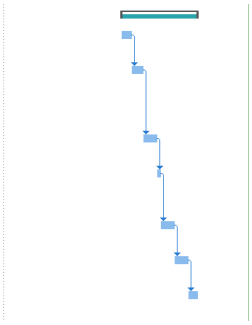


Figure 1.3: Task breakdown structure for project planning.

Phase 4	**20 dias**	**Seg 02/06/25**	**Sex 27/06/25**	
Draft thesis structur	3 dias	Seg 02/06/25	Qua 04/06/25	
Write core chapters	14 dias	Qui 05/06/25	Ter 24/06/25	24
Write conclusion	2 dias	Qua 25/06/25	Qui 26/06/25	25
Review, edit and format thesis	1 dia	Sex 27/06/25	Sex 27/06/25	26

Figure 1.4: Key project milestones and expected completion dates.

1.7 Document Structure

This thesis is structured into four main chapters, as outlined below:

- **Chapter 1: Introduction**
  The introductory chapter provides an overview of the thesis topic, objectives, and scope. It outlines the problem statement, research questions, and the significance of the study.
- **Chapter 2: Research Methodology**
  The research methodology chapter describes the approach, design, and methods used to conduct the study. Describes the data collection and analysis techniques, ensuring the reproducibility and validity of the research.
- **Chapter 3: State of the Art**
  This chapter reviews the current state of the art related to the thesis topic. Explores existing literature, technological advancements, and contemporary practices to contextualize research within its academic and practical domain.
- **Chapter 4: Analysis and design**
  This chapter presents a comprehensive summary of the analysis and proposed design solutions for the existing problem. It outlines the current issue in detail and explains the steps taken to address and resolve it.
- **Chapter 5: Implementation**
  This chapter details the practical realization of the proposed design. It covers the technologies, tools, and platforms used to implement the system, along with the development process and integration strategies.
- **Chapter 6: Results Analysis**
  This chapter evaluates the implemented solution through experiments, testing, or case studies. It analyzes the outcomes, compares them against expectations or benchmarks, and discusses the implications of the findings.

- **Chapter 7: Conclusion and Future Work**
  The final chapter summarizes the research findings, highlights key contributions, and reflects on the study's limitations. It also proposes potential directions for future research and system enhancements.

Chapter 2

Methodology

The methodology chapter serves as the foundation for understanding the systematic approach employed in this research to achieve the stated objectives. It outlines the philosophical framework, research design, data collection techniques, and analytical strategies that underpin the study. By providing a detailed account of the processes and decisions involved, this chapter ensures transparency and reproducibility, while also addressing the rigor and validity of the chosen methods.

2.1 Research Design

The research design for this study is structured to ensure a systematic and comprehensive analysis of cloud providers. It employs two primary methodologies:

Official Documentation Analysis

To establish a foundational understanding, the study will utilize the official documentation provided by leading cloud service providers. These documents offer reliable and accurate insights into the features, architecture, and capabilities of the providers, ensuring an authentic source for comparative data. The documentation serves as the basis for identifying key parameters, such as performance, scalability, pricing, and security.

Systematic Literature Review (SLR) using PRISMA Framework

To complement the primary data from official documentation, this study will perform a systematic literature review (SLR) following the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) approach ensures a rigorous, reproducible, and unbiased review process, focusing on academic articles and comparative studies of cloud providers. Articles will be sourced from trusted databases such as Institute of Electrical and Electronics Engineers (IEEE) Xplore, Springer, Association for Computing Machinery (ACM) Digital Library, and others, using relevant keywords like “cloud provider comparative analysis,” “AWS vs. Azure vs. GCP,” and “cloud computing performance benchmarks.”

The combination of official documentation and peer-reviewed literature ensures a balanced and evidence-based analysis, addressing both theoretical insights and practical evaluations.

2.2 Sampling and Data Collection Methods

This study employs purposive sampling and systematic data collection techniques to ensure the relevance and reliability of the data analyzed:

Sampling Methods

The study uses purposive sampling to select both cloud service providers and academic literature. The following criteria are applied:

- **Cloud Service Providers**: The study focuses on leading cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), Databricks and Snowflake. These providers are selected due to their market dominance, comprehensive documentation, and availability of comparative research.
- **Academic Literature**: Articles included in the systematic literature review are selected based on their relevance to the comparative analysis of cloud services. The inclusion criteria require that the articles are peer-reviewed, published within the last 5–10 years, and indexed in credible databases (e.g., IEEE Xplore, ACM Digital Library, Springer).

Data Collection Methods

1. **Primary Data from Official Documentation**
   Data related to the capabilities, features, and pricing of cloud providers are collected directly from their official documentation and white papers. This ensures that the information is up-to-date and vendor-authenticated.
2. **Secondary Data from Literature Review**
   Using the PRISMA framework, academic articles are systematically collected. The process includes:
   - Formulating search queries with keywords such as "cloud service comparison," "performance benchmarks," and "cost analysis of cloud platforms."
   - Applying filters like publication date and relevance.
   - Screening articles based on their titles, abstracts, and full text to ensure they align with the study's focus.

The integration of these data sources ensures a robust foundation for evaluating the selected cloud service providers.

2.3 Research Questions

This section presents the research questions that guide this study. These questions are structured into four key thematic areas: Comparative Analysis, Pricing models and Cost Structures, FinOps Maturity and Cost Optimization, Market Positioning and Future Relevance. Each thematic area addresses specific aspects of cloud platforms in the context of data and AI.

2.3.1 Comparative Analysis

The first thematic area focuses on comparing major cloud platforms and tools such as GCP, AWS, Azure, Databricks and Snowflake in terms of their functionalities and efficiencies for data platform use cases. The research questions under this theme are:

1. What are the fundamental similarities and differences between the leading cloud providers in terms of core service offerings?
2. How do cloud providers differentiate themselves in terms of overall value proposition to various customer segments?

2.3.2 Pricing models and Cost Structures

The second thematic involves around comparing major cloud platform provider and tools in terms of their pricing model and structures. Thus, highlight which one is more cost-effective for specific solutions.

3. How do the pricing models of major cloud providers compare across common workloads (e.g., compute, storage, data transfer)?

2.3.3 FinOps Maturity and Cost Optimization

The third thematic explores the concept of FinOps maturity and its impact on effective cloud cost optimization. It addresses one of the core research questions by examining how organizations can evolve their FinOps practices to improve financial accountability and maximize value from cloud investments.

4. What FinOps tools and practices are supported by each cloud provider to enable real-time cost management and accountability?

2.3.4 Market Positioning and Future Relevance

The fourth and final thematic investigates how strategic market positioning influences an organization's long-term relevance and competitiveness. It addresses a key research question focused on understanding the factors that shape sustained value and differentiation in a rapidly evolving industry landscape.

5. How do cloud providers position themselves strategically in the market, and what are the implications for future growth and adoption?

2.4 PRISMA

This section provides an overview of the PRISMA (Systematic Review Protocol) conducted for this study using the PRISMA [1] - Preferred Reporting Items for Systematic Reviews and Meta-Analyses. Even though PRISMA provides a well-regarded method to conduct systematic reviews, this project followed a variation of the outlined framework. PRISMA recommendations were not strictly followed in this instance but were shaped to best match the needs and scope of the research at hand. It included identification, screening, assessment of eligibility, and inclusion of studies relevant to answering the research questions.

Data Sources

The data sources used to search for academic articles and technical reports are summarized in Table 2.1.

Table 2.1: Data Sources for Literature Review

Search Engine	Link
Google Scholar	`https://scholar.google.com`
Microsoft Academic	`https://academic.microsoft.com`
BASE (Bielefeld Academic Search Engine)	`https://www.base-search.net`
Semantic Scholar	`https://www.semanticscholar.org`
CORE (COnnecting REpositories)	`https://core.ac.uk`
B-ON (Biblioteca do Conhecimento Online)	`https://www.b-on.pt`

Search Terms

The search was performed using a combination of domains and associated keywords. Table 2.2 shows the domains and keywords used.

Table 2.2: Search Terms for Literature Review

Domain	Keywords
Cloud Service Offerings	"AWS vs Azure vs GCP," "Snowflake vs Databricks," "cloud service comparison," "data platform services"
Value Proposition	"cloud provider value proposition," "differentiation strategy," "customer segmentation in cloud services," "Snowflake positioning"
Pricing Models	"cloud pricing comparison," "compute and storage pricing AWS Azure GCP," "Snowflake cost structure," "Databricks pricing model"
FinOps Practices	"FinOps tools by cloud provider," "cost management in Snowflake and Databricks," "real-time cost tracking cloud," "FinOps maturity model"
Workload Types	"common cloud workloads," "compute-intensive workloads," "data analytics workloads," "multi-cloud workload comparison"

Inclusion and Exclusion Criteria

The following criteria were applied to ensure the relevance and quality of the articles included in the review, summarized in Table 2.3:

Query Table

The search queries used across various data sources and the corresponding number of articles retrieved are summarized in Table 2.4.

Criterion Type	Criteria
Inclusion Criteria	• Articles published within the time interval 2014-2024. • Peer-reviewed journal or conference papers. • Studies focusing on AWS, Google Cloud, and Microsoft Azure. • Articles addressing at least one of the research questions.
Exclusion Criteria	• Articles not available in full text. • Studies focusing on cloud platforms outside the scope of this research. • Duplicates across data sources.

Table 2.3: Inclusion and Exclusion Criteria for Article Selection

Table 2.4: Search Queries and Results

Search Query	Articles Extracted
"core service comparison AWS Azure GCP Snowflake Databricks"	30
"value proposition of Snowflake vs Databricks vs major cloud providers"	24
"cloud pricing models for compute, storage, and data transfer"	21
"FinOps tools and practices in AWS Azure GCP Snowflake Databricks"	18
"cost optimization for analytics workloads in cloud platforms"	16

Article Utilization and Referencing

The retrieved articles from the search strings, as outlined in Table 2.4, were the main sources of information for the data to be collected in this study. Each article was analyzed for relevance, reliability, and contribution to the research objectives. Not all the articles retrieved could be used or cited in the final analysis and discussion.

Several factors influenced the decision to exclude certain articles, such as:

- **Duplication:** Articles with overlapping content were consolidated to ensure clarity and avoid redundancy.
- **Relevance:** Articles that did not align closely with the specific focus of the research were omitted.
- **Depth of Analysis:** Some articles lacked sufficient detail to support the insights required for the study.

Hence, the final list of references reflects a most relevant and impactful sources that have been retrieved from the original pool of articles extracted. This way, this enables the research to remain focused while retaining support from high-quality and relevant information.

The combination of these sources and methods ensures a comprehensive and rigorous review to support the research objectives.

2.5 Data Analysis Techniques

In this study, a combination of qualitative and quantitative data analysis techniques will be employed to address the research questions and provide comprehensive insights into the comparison of cloud platforms.

Qualitative Analysis

The qualitative study will be guided by thematic analysis of official documentation from selected cloud platforms, such as AWS, Google Cloud, and Microsoft Azure, with categorical comparison based on key features of the platforms, their capabilities, and support of AI/ML models. A systematic review will be conducted from the academic literature in order to distill common themes and challenges that relate to cloud data management, integration, and performance. A thematic analysis approach will be used to identify recurring patterns and trends in the data.

Quantitative Analysis

In the case of quantitative analysis, data for performance benchmarks, cost-efficiency metrics, and scalability tests are drawn from academic literature. Further statistical methods-descriptive statistics including means, median, and standard deviation-are considered for comparing different metrics of platform performance. It also includes plotting charts and graphs to indicate main differences, trends, and outliers. Cost models based on reported prices and usage scenarios will be evaluated for the economic efficiency of each platform.

Comparative Framework

Both qualitative and quantitative analyses are combined into a comparative framework. It highlights strengths and weaknesses for every cloud platform with regard to the research questions and allows clear and well-structured comparison by decision-makers and researchers in this area.

2.6 Limitations and Delimitations

Limitations

While the study aims to provide an objective and comprehensive comparison of cloud platforms, several limitations need to be acknowledged:

- **Scope of Platforms**: The study focuses on five major cloud platforms (AWS, Google Cloud, Microsoft Azure, Databricks and Snowflake), which may not fully represent the entire market of cloud services. Other smaller or emerging platforms (such as IBM Cloud or Oracle Cloud) are not included in this comparison.
- **Access to Data**: While official documentation and publicly available reports are utilized, some proprietary or internal performance data might be inaccessible. The analysis

will rely heavily on secondary data from the literature, which may have varying levels of quality and accuracy.

- **Time Sensitivity**: Cloud platforms are continuously evolving, and the information used in this study may become outdated over time. Although the most recent data available at the time of the study will be used, platform updates or changes that occur after data collection may not be included.
- **Subjectivity in Qualitative Assessment**: The qualitative comparison of platform features and capabilities may be influenced by the researcher's interpretation, especially when dealing with complex or vague descriptions found in official documentation or academic papers.

Delimitations

Delimitations refer to the boundaries set by the researcher to narrow the focus of the study:

- **Focus on Major Cloud Platforms**: The research is limited to the five cloud platforms (AWS, Google Cloud, Microsoft Azure, Databricks and Snowflake). This is a deliberate choice to focus on the most widely adopted and researched platforms in the industry, which are considered to be representative of the market.
- **Specific Research Areas**: The study limits its analysis to the comparison of features, cost-efficiency, FinOps Maturity and Future relevance in the context of cloud platforms. Other potential factors, such as customer support, geographic availability, and user experience, are not included in this study.
- **Data Types and Use Cases**: The research is primarily concerned with data platform use cases in the cloud, particularly in the context of telecommunications data. It does not extend to other industries or broader use cases that may also benefit from cloud services.

These limitations and delimitations help define the scope of the research and ensure a focused and manageable study while providing transparency about potential gaps in the analysis.

Chapter 3

State of the art

This chapter provides an overview of the state of the art of cloud service providers regarding their capabilities in data management and artificial intelligence/machine learning. Three big players will be discussed: Google Cloud Platform, Amazon Web Services, and Microsoft Azure. Among these, GCP will be considered as the main provider because of its key role in the implementation of the use case later in this document.

It covers those features, tools, and services relevant to these platforms, placing special emphasis on how they have fostered AI/ML workflows and how solutions provided are data-driven. The intent behind this chapter will be to develop, side by side with these providers, a clear-cut view of strengths, limitations, and what the one-stop shop proposition offered by each might be in the domain, developing an anchor in advanced practical application reviewed in later chapters.

3.1 Google Cloud Platform

Google Cloud Platform is a suite of cloud-based services offered by Google. Extremely respected for its scalability, security, and all forms of innovation, GCP natively provides an end-to-end ecosystem that supports modern data workloads through to AI/ML. By facilitating advanced analytics, enterprise-grade security features, and more, GCP allows organizations to uncover insights, automate processes, and deploy AI-driven applications efficiently. This section walks through what GCP has to offer across several key domains: data storage, integration, and governance; security; business intelligence; machine learning; and operational workflows [2].

3.1.1 Data Storage and Querying

It provides a complete suite of enterprise data storage and querying solutions on Google Cloud Platform for scalability, performance, and flexibility. Its use cases are quite varied; it can handle very structured and unstructured data, right through to real-time analytics and large-scale machine learning workloads. The GCP offering will seamlessly integrate with various tools and frameworks that will empower users to run complex queries, store large volumes of data, and scale efficiently.

Cloud Storage

Google Cloud Storage is a highly available object storage service, offering the best in class scalability and security, capable of handling structured or unstructured data. It is a single solution to store and manage data across a wide range of use cases, including data lakes,

analytics, backup and archiving, disaster recovery, content delivery, and more. Cloud Storage provides a number of storage classes-including Standard, Nearline, Coldline, and Archive-to meet different data access and cost requirements[3].

Key features include:

- **Multi-region and dual-region storage**: Cloud Storage supports global, multi-region replication, ensuring high availability and low-latency access to data across different geographical locations.
- **Object lifecycle management**: Users can define rules to automatically move data between different storage classes based on predefined conditions, optimizing cost and data management.
- **Access control and security**: Fine-grained access control with Identity Access Management (IAM) roles, as well as integration with Cloud Key Management Services (KMS) for data encryption, ensures data security at rest and during transit.
- **Integration with other GCP services**: Cloud Storage works seamlessly with services such as BigQuery, Cloud Pub/Sub, Dataflow, and Dataproc for further data processing and analysis.

These features enable enterprises to manage large datasets efficiently, while ensuring security and compliance, particularly for industries that handle sensitive data.

BigQuery

BigQuery is a serverless, highly scalable Structured Query Language (SQL)-based analytics engine that enables one to run fast, SQL-based queries on enormous datasets. Being a fully managed service, BigQuery abstracts the need to manage any infrastructure, thereby enabling users to focus on data analysis without concern about performance tuning or scalability [4].

Key features include:

- **Serverless architecture**: BigQuery automatically scales to accommodate workloads of any size, from small queries to petabyte-scale datasets. Users don't need to provision or manage any infrastructure.
- **Massive parallel processing**: BigQuery uses a distributed architecture that processes queries using multiple nodes, significantly improving performance for complex queries.
- **Federated queries**: Users can run queries across multiple data sources, including Google Cloud Storage, Google Sheets, and external databases, without moving the data into BigQuery.
- **BigQuery ML**: BigQuery provides built-in machine learning capabilities through BigQuery ML, enabling users to create and deploy machine learning models directly within the SQL interface. This allows for fast experimentation without needing to export data to external ML tools or frameworks.
- **In-memory BI engine**: BigQuery's BI Engine is a fast, in-memory analysis service that accelerates dashboarding and reporting by providing real-time query results in tools like Looker and Data Studio.

- **Real-time analytics**: BigQuery supports real-time analytics through streaming data ingestion, allowing users to analyze data as it is generated and respond instantly to business changes.
- **Cost-effective pricing model**: BigQuery uses a pay-as-you-go pricing model, charging only for the amount of data processed during queries, with the option to control costs by using table partitioning and clustering.

It aims at analysts, data engineers, and business users who want fast and cost-efficient analysis of very large data. BigQuery works with both batches and real-time data, leveraging machine learning along the way; it easily offers insights from all structured and unstructured- or semi-structured-data.

3.1.2 Data Integration and Transformation

Well, the effective integration of data into unified and actionable datasets means the creation of insights for better decision-making. GCP houses a suite of effective tools that are able to integrate, transform, and process data with efficacy, thus ensuring movement with ease across sources, formats, and storage systems. These would contribute to an enterprise smoothing its data pipelines, automating workflows, and ensuring that consistency, accuracy, and readiness for analytics and machine learning are adhered to.

Data Fusion

Google Cloud Data Fusion [5] is a completely managed, no-code/low-code Extract, Transform, Load (ETL) service that helps design, manage, and orchestrate data pipelines in a graphical way. The intuitive interface lets technical and non-technical users alike drag-and-drop the path to develop complex data integration workflows with absolutely no coding required. It provides an interface where it can easily transform data using pre-built connectors for databases, file systems, APIs, and cloud services.

Key features of Data Fusion include:

- **No-code/Low-code interface**: Data Fusion enables users to design and manage data pipelines visually, allowing non-developers to participate in data integration efforts and reducing the time and complexity of pipeline development.
- **Pre-built connectors**: Data Fusion offers a wide range of out-of-the-box connectors to integrate with various data sources, including Google Cloud Storage, BigQuery, Cloud Pub/Sub, relational databases, and external cloud services.
- **Batch and real-time support**: Data Fusion supports both batch and real-time data workflows, enabling users to process large datasets efficiently or handle streaming data from real-time sources like IoT devices and social media platforms.
- **Data transformation and enrichment**: Data Fusion allows users to perform data transformation tasks such as filtering, aggregating, joining, and applying custom logic, which can be integrated into a comprehensive ETL workflow.
- **Integration with GCP services**: Data Fusion integrates seamlessly with other Google Cloud services such as BigQuery, Cloud Pub/Sub, Cloud Dataflow, and Cloud Storage, making it easy to move and transform data across your cloud infrastructure.

- **Data Governance and Monitoring**: Data Fusion provides built-in monitoring, logging, and alerting features, helping organizations track data pipeline performance, handle errors, and maintain compliance with data governance policies.

Data Fusion's visual interface and scalability make it ideal for organizations that need to simplify the development of complex data pipelines, improve collaboration between technical and non-technical teams, and manage data integration tasks without significant manual effort.

Dataflow

Google Cloud Dataflow [6] is a serverless, in-full-managed data processing service that enables both stream and batch data processing. Powered by Apache Beam, Dataflow is designed for large-scale data ingestion, transformation, and analysis in real time or on scheduled intervals. Dataflow simplifies the complexity of building distributed data pipelines since it automatically scales resources to match the workload and guarantees efficient performance regardless of the size or volume of your data.

Key features of Dataflow include:

- **Unified stream and batch processing**: Dataflow supports both stream and batch processing models, allowing users to process real-time data (e.g., from IoT sensors, logs, or social media feeds) and historical data (e.g., from data lakes or databases) in a single pipeline.
- **Apache Beam SDK**: Dataflow is based on the Apache Beam programming model, allowing users to write data processing pipelines in Java, Python, or Go. Apache Beam provides a powerful abstraction for building complex data processing tasks, including filtering, grouping, and aggregating data.
- **Serverless infrastructure**: As a fully managed service, Dataflow automatically handles the scaling of resources, eliminating the need for users to manage infrastructure. Dataflow adjusts resource allocation in real-time based on pipeline load, optimizing performance and cost.
- **Real-time data ingestion and transformation**: Dataflow excels in real-time data processing, making it ideal for applications that require immediate analysis or event-driven processing. Users can implement real-time analytics, event streaming, or anomaly detection in their pipelines.
- **Built-in connectors and integrations**: Dataflow supports integrations with many Google Cloud services, such as Cloud Pub/Sub, BigQuery, Cloud Storage, and Cloud SQL, enabling seamless data movement across systems.
- **Advanced windowing and triggers**: Dataflow offers advanced features like windowing, triggers, and time-based processing, which allow users to group data by time intervals or events, enabling sophisticated analytics on streaming data.
- **Fault tolerance and data consistency**: Dataflow is designed with fault tolerance in mind, automatically recovering from failures and ensuring exactly-once processing semantics, which guarantees that each event is processed only once even in the case of retries or errors.

Dataflow best fits organizations operating on real-time data streams, high-velocity data, and complex transformation tasks. It lets the business analyze data as it is created and gives quicker, more valuable insights. With auto-scaling and deep integration with the Google Cloud ecosystem, Dataflow simplifies how to build, deploy, and manage sophisticated data processing pipelines.

3.1.3 Data Governance and Management

Accessibility, organization, security, and compliance are core requirements for data in modern enterprises. With organizations dealing with volumes of data that continue to increase, effective governance is not only about ensuring regulatory compliance but also about improving data quality, managing metadata, and enabling data-driven decision-making. The Google Cloud Platform is going to provide the tools and services to let an organization set up appropriate practices for data governance: making data discoverable, well-organized, and compliant with industry regulations.

Dataplex

Google Cloud Dataplex [7] is an intelligent data fabric service that offers consistent data management across multiple data lakes, databases, and other repositories. It aims to provide one control plane for management, security, and governance of all data, thereby making the data easy to access, discover, and analyze across silos and storage systems in a specific format. Dataplex is designed to make data governance easier while granting organizations the capability to manage large-scale, complex data environments with ease.

Key features of Dataplex include:

- **Unified data governance**: Dataplex brings together disparate data sources, including data lakes (e.g., Cloud Storage), data warehouses (e.g., BigQuery), and databases (e.g., Cloud SQL), into a unified data fabric that allows for comprehensive governance and management. This simplifies the complexity of managing data spread across different storage systems.
- **Data discovery**: Dataplex offers enhanced data discovery capabilities, helping users easily locate data assets across their organization. This is critical for organizations dealing with large and diverse data sets, where data may be spread across multiple departments or platforms.
- **Data lineage tracking**: With Dataplex, users can track data lineage to understand where data originated, how it was transformed, and where it is being used. This capability is particularly important for ensuring data integrity, compliance, and transparency, as it allows businesses to trace data flows across the entire data lifecycle.
- **Data classification**: Dataplex supports automatic classification of data based on user-defined rules, such as categorizing data as sensitive, personally identifiable information (PII), or business-critical data. This helps organizations enforce policies for securing and managing sensitive data in line with industry regulations.
- **Policy enforcement**: Dataplex allows organizations to enforce data management policies at scale, ensuring that data governance standards are met consistently across different teams and services. Policies may include access controls, data retention requirements, and compliance with legal frameworks (e.g., GDPR, HIPAA).

- **Metadata management**: Dataplex automatically collects and manages metadata for data assets across the organization. This metadata includes data definitions, data quality metrics, and usage statistics, making it easier for users to understand the context and quality of the data they are working with.
- **Security and access control**: Dataplex integrates with GCP's security and identity services, such as Identity and Access Management (IAM) and Cloud Key Management Service (KMS), to control who can access specific data assets and ensure that data is encrypted both in transit and at rest.
- **Collaboration and auditing**: Dataplex provides auditing capabilities, allowing organizations to track changes and actions on their data assets. This improves collaboration among teams and ensures that all data-related actions are logged for accountability and compliance.

Providing a comprehensive set of data governance tools, Dataplex lets organizations establish consistent policies and governance mechanisms on various storage systems, maintains the data well-organized and classified, with proper protection-all in collaboration-enhanced and operational complication-reduced manners.

Cloud Data Catalog

Cloud Data Catalog [8] is a fully managed service where organizations manage the discovery, governance, and management of metadata at scale. As such, Cloud Data Catalog acts like a centralized repository for metadata ingested both from GCP services and third-party sources to make the discovery of data easy, hence collaboration, with data engineers and analysts to get better analytics and ML on data in every organization. The catalog simplifies data management through the unification of metadata, enrichment of data governance, and assurance that the teams have access to the data they need with confidence.

Key features of Cloud Data Catalog include:

- **Metadata discovery**: Cloud Data Catalog allows users to discover metadata from GCP services such as BigQuery, Cloud Storage, and Dataproc, as well as from external sources. Users can search and browse metadata assets based on key attributes such as tags, labels, and descriptions.
- **Data lineage visualization**: Data Catalog automatically captures and visualizes the data lineage of assets within the catalog, enabling users to see the data's history, transformations, and how it flows through various processes and pipelines.
- **Tagging and categorization**: Users can add custom tags to data assets, enabling more granular classification and filtering of data. Tags can also be used to enforce governance rules, such as data sensitivity and access restrictions.
- **Policy management and access control**: Cloud Data Catalog integrates with GCP's IAM to enable fine-grained access control to metadata. Users can set permissions on who can view, edit, or share specific metadata entries based on roles and responsibilities.
- **Integration with other GCP services**: Data Catalog integrates seamlessly with other GCP services such as BigQuery, Dataproc, and Cloud Composer, making it easy to link metadata to specific datasets and data processing pipelines.

- **Audit and tracking**: Cloud Data Catalog provides built-in auditing and tracking features, allowing organizations to monitor and track changes to metadata, ensuring compliance with data governance policies.

Cloud Data Catalog enables data governance through the creation of organized and searchable metadata catalogs, with data lineage tracking and consistent policy enforcement across teams.

Data Management Best Practices in GCP

To effectively manage and govern data, organizations must follow best practices that align with GCP's offerings. These best practices include:

- **Data classification and tagging**: Tagging data assets with relevant metadata such as sensitivity level, classification type, and ownership ensures that data is organized and can be easily found and governed.
- **Automated data lifecycle management**: Set up rules and workflows for automatically moving, archiving, or deleting data based on business requirements and regulatory compliance.
- **Compliance with regulations**: Leverage GCP's compliance certifications, such as GDPR, HIPAA, and SOC 2, and implement data management processes that adhere to legal frameworks and industry standards.
- **Data stewardship**: Assign data stewards to manage data quality, ensure accuracy, and oversee data governance policies across departments.
- **Audit trails and logging**: Enable audit logs for all data activities to ensure transparency and track changes, deletions, and access to data.

By combining GCP's advanced governance tools with these best practices, organizations can build a robust data governance framework that ensures their data is secure, compliant, and well-managed.

3.1.4 Data Security and Privacy

Data security and privacy are the most important concerns for an organization operating in modern cloud environments, where the quantum of sensitive information handled is huge. Google Cloud Platform provides a complete suite of tools and services to protect data against unauthorized access, misuse, and exfiltration, along with mechanisms to adhere to the various global data privacy regulations such as GDPR and HIPAA.

Cloud DLP

Google Cloud Data Loss Prevention (Cloud DLP) is a fully managed service that helps organizations detect, classify, and protect sensitive information across their data ecosystem. Cloud DLP is designed to solve privacy and data protection needs by identifying patterns and data types that could indicate sensitive information, such as personally identifiable information, payment card information, and intellectual property.

Key features of Cloud DLP include:

- **Sensitive data detection**: Uses machine learning and pattern recognition to identify sensitive data types, including names, Social Security numbers, credit card numbers, and medical records, in both structured and unstructured datasets.
- **Data masking and tokenization**: Provides tools for redacting sensitive data by replacing or masking it with placeholders, ensuring privacy during data processing or analysis.
- **Data classification**: Allows organizations to classify data into predefined categories or create custom templates to suit specific business or regulatory needs.
- **Integration with GCP services**: Seamlessly integrates with services like BigQuery, Cloud Storage, and Pub/Sub, enabling real-time and batch data protection.
- **Compliance support**: Assists organizations in adhering to compliance frameworks like GDPR, HIPAA, and PCI DSS by safeguarding sensitive data.

Cloud KMS

Google Cloud Key Management Service (Cloud KMS) [9] provides a secure and scalable encryption key management solution. It provides an enterprise with a means of creating, storing, managing, and using cryptographic keys used to protect data at rest and in transit.

Key features of Cloud KMS include:

- **Encryption key management**: Enables the creation and lifecycle management of symmetric and asymmetric keys, supporting encryption, decryption, and digital signing operations.
- **Integration with GCP services**: Cloud KMS integrates seamlessly with GCP's storage and processing services, such as BigQuery, Cloud Storage, and Compute Engine, to provide transparent encryption.
- **Hardware Security Modules (HSMs)**: Supports hardware-backed key management through the Cloud HSM service, offering additional layers of security for sensitive keys.
- **Key rotation and auditing**: Facilitates automatic or manual key rotation and provides detailed audit logs of key usage for security and compliance tracking.
- **Customer-managed encryption keys (CMEK)**: Allows organizations to retain control over encryption keys used to protect their data, ensuring compliance with regulatory requirements and enhancing trust.

VPC Service Controls (VPC-SC)

Virtual Private Cloud (VPC)-SC [10] represents a security or protection service around sensitive data located in GCP services. What it does is actually to create an explicit security perimeter, ensuring ultimately that data only remains accessible when it is within well-defined boundaries, hence a reduced risk due to exfiltration and unauthorized access.

Key features of VPC-SC include:

- **Security perimeters**: Enforces service-specific boundaries to restrict data movement and interactions between GCP services and external networks.

- **Exfiltration prevention**: Blocks unauthorized access and exfiltration of data, ensuring that sensitive information remains within trusted zones.
- **Access policies**: Works in conjunction with IAM and Cloud Identity to enforce fine-grained access policies for users and service accounts within the perimeter.
- **Audit logging and monitoring**: Provides visibility into perimeter breaches and potential security violations through integration with Cloud Logging and Cloud Monitoring.
- **Compliance and regulatory adherence**: Assists in meeting data residency and sovereignty requirements by ensuring data remains within specific geographic regions.

By leveraging Cloud DLP, Cloud KMS, and VPC-SC, organizations can implement robust security measures to protect sensitive data and maintain compliance with global privacy standards.

3.1.5 Business Intelligence and Visualization

Business Intelligence and visualization tools ensure that the data is usable. GCP makes exploring, reporting, or visualizing your data more accessible by making the tools intuitively simple and the integrations seamless with their data services.

Looker Studio

Looker Studio [11], formerly known as Data Studio, is Google Cloud's powerful, self-service BI platform that enables users to create visually compelling and interactive dashboards, reports, and data visualizations. It empowers organizations to analyze data and communicate insights effectively, supporting data-driven decision-making at all levels.

Key features of Looker Studio include:

- **Customizable dashboards**: Users can design fully customizable dashboards tailored to their specific needs, incorporating charts, graphs, and widgets to visualize data trends and metrics.
- **Real-time integration**: Looker Studio integrates natively with BigQuery, Google Analytics, Cloud Storage, and other GCP services, allowing real-time data exploration and updates.
- **Collaboration and sharing**: Enables teams to collaborate on reports and share interactive dashboards via web links or embedded views, enhancing organizational transparency.
- **Support for multiple data sources**: Looker Studio connects to a wide array of data sources, including third-party platforms like Salesforce, MySQL, and PostgreSQL, providing a unified view of data across the organization.
- **Advanced data transformations**: Offers built-in tools for data preparation, filtering, and aggregation, simplifying the process of creating meaningful insights from raw data.
- **Interactive reporting**: Dashboards and reports are fully interactive, allowing users to drill down into details, apply filters, and explore data at different levels of granularity.
- **Templates and pre-built connectors**: Provides pre-built templates and connectors for common use cases, accelerating report creation and deployment.

By leveraging Looker Studio, organizations can democratize access to data insights, enabling non-technical users to participate in data analysis and decision-making processes. This fosters a culture of data-driven innovation and enhances the overall value derived from enterprise data assets.

3.1.6 Machine Learning and AI

It fully equips one of the most powerful, large-scale toolkits and services that exist for the development, deployment, and management of machine learning and artificial intelligence solutions on the Google Cloud Platform. These services range from data scientists and ML engineers to business analysts, who can help organizations extract more value from AI and apply it to foster innovation and a competitive advantage.

Vertex AI

Vertex AI [12] is GCP's single platform that unifies the entire machine learning lifecycle. It simplifies building, deploying, and operationalizing ML models with the integration of key tools and automation of repetitive tasks. Vertex AI supports both AutoML for users with minimal expertise in machine learning and also custom model development for advanced uses.

Key features of Vertex AI include:

- **Integrated ML pipeline support**: Combines data preparation, model training, hyperparameter tuning, deployment, and monitoring into a single, cohesive platform.
- **AutoML capabilities**: Enables users to train high-quality models using pre-built algorithms without requiring extensive knowledge of ML, reducing development time.
- **Custom model training**: Supports custom model development with frameworks like TensorFlow, PyTorch, and scikit-learn, offering flexibility for advanced use cases.
- **Managed notebooks**: Provides Jupyter-based managed notebooks with pre-configured environments, facilitating rapid experimentation and collaboration among teams.
- **Feature Store**: Allows centralized management and sharing of features across ML models, ensuring consistency and reusability.
- **Model monitoring**: Automates drift detection and performance monitoring for deployed models, ensuring they remain accurate and reliable in production.
- **MLOps integration**: Includes tools for Continuous Integration / Continuous Deployment (CI/CD) pipelines, version control, and automated deployments, supporting robust machine learning operations (MLOps).
- **Explainable AI**: Offers tools to interpret model predictions, such as feature attribution and visualization, enhancing trust and compliance in AI systems.
- **Seamless integration with GCP**: Works seamlessly with BigQuery, Cloud Storage, and other GCP services, enabling efficient data access and processing.

AI Infrastructure and Pre-built Models

Beyond Vertex AI, GCP provides additional AI resources to accelerate development and deployment:

- **TPUs and GPUs [13]**: High-performance hardware accelerators for training and inference workloads, reducing time and cost.
- **AI APIs [14]**: Pre-trained APIs for common use cases, such as natural language processing (NLP), computer vision, and translation. Examples include the Cloud Vision API, Cloud Natural Language API, and Dialogflow for conversational AI.
- **AI Workbench [15]**: A collaborative environment for researchers and developers to work on advanced AI models, including generative AI and reinforcement learning.

3.1.7 DataOps and MLOps

Operationalizing the data workflows and machine learning models brings scalability, reliability, and efficiency desired by modern data-driven organizations. GCP provides a very rich set of tools to implement all aspects of DataOps and MLOps practices, thus enabling seamless workflows and collaboration between groups.

Cloud Composer

Cloud Composer [16] is a fully managed workflow orchestration service built on Apache Airflow. It is designed to automate, monitor, and manage complex data pipelines, ensuring consistency and reliability across data processes.

Key features of Cloud Composer include:

- **Workflow automation**: Simplifies the orchestration of data workflows by connecting multiple GCP services, such as BigQuery, Cloud Storage, and Dataflow, as well as third-party tools.
- **Scalability and performance**: Offers elastic scaling capabilities to handle workflows of varying complexity and volume.
- **Integration with Airflow ecosystem**: Leverages Apache Airflow's vast ecosystem of plugins and operators to support diverse data processing tasks.
- **Monitoring and alerting**: Provides built-in monitoring and alerting mechanisms for tracking workflow execution and handling failures proactively.
- **Multi-cloud and hybrid support**: Enables cross-cloud and hybrid data orchestration, providing flexibility for organizations with diverse infrastructure setups.

Vertex AI Pipelines

Vertex AI Pipelines are a set of powerful tools designed to help with the seamless operationalization of machine learning workflows from end to end. It automates steps in the ML lifecycle for consistency, repeatability, and efficiency in model development and deployment.

Key features of Vertex AI Pipelines include:

- **Pipeline automation**: Automates every stage of the ML lifecycle, including data preprocessing, feature engineering, model training, evaluation, deployment, and monitoring.

- **Kubeflow Pipelines integration**: Built on Kubeflow Pipelines, Vertex AI Pipelines provides a framework for defining and running ML workflows in a containerized environment.
- **Versioning and reproducibility**: Tracks pipeline versions and parameters, ensuring that models and workflows are reproducible and traceable for audit and compliance purposes.
- **Continuous integration and deployment (CI/CD)**: Supports CI/CD practices for ML models, enabling rapid iteration, testing, and deployment while maintaining quality and performance.
- **Custom components and templates**: Allows users to create custom components or leverage pre-built templates to accelerate pipeline development.
- **Scalability and performance**: Runs pipelines efficiently on GCP's infrastructure, scaling resources dynamically to optimize cost and performance.
- **Integrated monitoring and logging**: Tracks pipeline execution and provides logs and metrics for debugging and optimization.

Benefits of DataOps and MLOps in GCP

By leveraging Cloud Composer and Vertex AI Pipelines, organizations can achieve:

- **Enhanced collaboration**: Facilitates better coordination between data engineers, data scientists, and ML engineers through standardized workflows and shared tools.
- **Improved reliability**: Ensures consistent and reliable execution of data and ML workflows, reducing errors and downtime.
- **Faster time-to-market**: Accelerates the deployment of data products and machine learning models, enabling quicker realization of business value.
- **Scalable operations**: Supports workflows of any scale, ensuring that processes remain efficient and cost-effective as data volumes and complexity grow.
- **Compliance and governance**: Provides audit trails and versioning to meet regulatory and organizational compliance requirements.

By integrating DataOps and MLOps practices into their workflows with GCP, organizations can build resilient, efficient, and scalable systems for handling data and AI initiatives.

3.2 Amazon Web Services

Amazon Web Services (AWS) is an integrated and popular cloud platform from Amazon. It comprises more than 200 services, all fully featured, in computation, storage, networking, machine learning, analytics, and more. Known for its scalability, flexibility, and reach to almost any place in the world, AWS serves millions of customers, from small startups and enterprises to public sector organizations. The section covers an in-depth look into AWS's main offerings in the areas of data storage, integration, governance, security, business intelligence, machine learning, and operational workflows.

3.2.1 Data Storage and Querying

AWS provides a full set of scalable and reliable data storage solutions that fit the needs of diverse use cases, from unstructured object storage to relational and NoSQL databases. These services are designed for high volumes of data and security, and are integrated with other AWS services to support analytics, backup, disaster recovery, and more.

S3 (Simple Storage Service)

Amazon S3 [17] is a highly scalable object storage service designed to store and retrieve any amount of data from anywhere on the web. It is commonly used for backup and archiving, big data analytics, and serving static assets like media files. Key features include:

- **Durability and availability**: Amazon S3 offers 99.999999999% (11 nines) durability by redundantly storing data across multiple devices in various Availability Zones. Multiple storage classes allow users to optimize costs based on access frequency.
- **Lifecycle management**: Enables automatic transitions of objects between storage classes based on defined policies, reducing costs while maintaining accessibility. It also supports automated data retention policies for compliance and efficient resource management.
- **Security features**: Provides robust security mechanisms such as encryption at rest (using AWS Key Management Service or customer-managed keys) and in transit (using HTTPS). S3 also supports fine-grained access control through bucket policies and integration with AWS Identity and Access Management (IAM).
- **Versioning and replication**: Helps protect against accidental overwrites or deletions by maintaining multiple versions of objects. Cross-region replication enhances data durability and facilitates global data distribution and disaster recovery strategies.

Redshift

Amazon Redshift [18] is a fully managed, petabyte-scale cloud data warehouse solution tailored for high-performance analytical processing. It is designed to efficiently query structured and semi-structured data, making it a popular choice for business intelligence and reporting. Key features include:

- **Columnar storage**: Stores data in a columnar format to significantly reduce the amount of data read during queries, enhancing performance for analytical workloads.
- **Query federation**: Allows users to extend their Redshift queries to data stored in S3, relational databases, and other supported sources without the need to move data, providing a unified query experience.
- **Integration with AWS services**: Redshift seamlessly connects to services like Amazon S3 for data lake integration, AWS Glue for ETL processes, and Amazon QuickSight for visualization, enabling end-to-end data pipelines.
- **Concurrency scaling**: Automatically adds query processing capacity in response to demand, ensuring consistent performance even with many users accessing the data warehouse simultaneously.

Aurora

Amazon Aurora [19] is a MySQL and PostgreSQL-compatible relational database engine that combines the high availability and performance of high-end commercial databases at a fraction of the cost. It is fully compatible with MySQL and PostgreSQL, making migration of existing applications easy.Features include:

- **High availability**: Aurora provides multi-AZ deployments, with automated failover to a standby replica in the event of a failure. It also ensures minimal downtime with continuous backups to Amazon S3 and point-in-time recovery.
- **Auto-scaling**: Dynamically adjusts database storage and compute capacity in response to workload demands, eliminating the need for manual scaling and ensuring consistent performance.
- **Serverless options**: Aurora Serverless offers an on-demand auto-scaling configuration, ideal for variable workloads or development environments. It enables applications to scale seamlessly without requiring database capacity provisioning.
- **Advanced security**: Includes encryption of data at rest and in transit, network isolation through Amazon VPC, and integration with AWS IAM for fine-grained access control.

3.2.2 Data Integration and Transformation

AWS provides a robust suite of tools for data integration and transformation, enabling organizations to build efficient data pipelines, orchestrate workflows, and process large volumes of data across diverse sources. These services are designed to handle complex data tasks while reducing operational overhead.

AWS Glue

AWS Glue [20] is a fully managed ETL (Extract, Transform, Load) service that simplifies data integration workflows. It is ideal for building, automating, and scaling data pipelines to prepare data for analytics or machine learning. Key features include:

- **Serverless architecture**: Eliminates the need for provisioning or managing infrastructure, allowing users to focus solely on data processing tasks.
- **Data crawlers**: Automatically scan, discover, and catalog metadata from structured and unstructured data sources, ensuring that the data catalog remains up-to-date.
- **Python-based scripts**: Provides flexibility to customize and define complex transformations using Python or PySpark scripts, enabling support for diverse business logic and custom workflows.
- **Job scheduling**: Offers built-in tools to automate and schedule recurring ETL tasks, ensuring timely and consistent data processing.

AWS Data Pipeline

AWS Data Pipeline [21] is a data workflow orchestration service that simplifies the scheduling, movement, and transformation of data between AWS services and on-premises systems. It is particularly useful for recurring data workflows. Key features include:

- **Scheduled execution**: Allows users to define and schedule data workflows at specific intervals, ensuring data processing tasks are executed consistently.
- **Diverse integration**: Supports integration with a wide array of AWS services such as S3, Redshift, and RDS, as well as on-premises databases and storage systems.
- **Resiliency**: Includes built-in monitoring, failure detection, and automatic retries, ensuring workflows remain robust and complete even in the face of transient failures.
- **Custom activities**: Enables users to define and execute custom scripts or applications as part of the pipeline workflow for additional flexibility.

Kinesis

Amazon Kinesis [22] is a platform designed for real-time data streaming and processing, making it invaluable for scenarios where immediate insights from streaming data are required. It supports large-scale data ingestion from diverse sources like IoT devices, application logs, and social media. Key features include:

- **Data Streams**: Enables real-time ingestion and processing of massive amounts of data with low latency, supporting applications such as live dashboards and monitoring systems.
- **Kinesis Firehose**: Simplifies the delivery of streaming data to destinations such as S3, Redshift, or Elasticsearch, with built-in support for data transformation and compression.
- **Kinesis Analytics**: Allows users to run real-time analytics on streaming data using SQL-like queries, reducing the time to derive insights from live data streams.
- **Scalability**: Automatically scales to handle increased data throughput, ensuring performance is maintained even as workloads grow.

3.2.3 Data Governance and Management

AWS provides the most advanced tools for implementing sound data governance practices for ensuring security, compliance, and efficient management of data within an organization. These services make it easier to organize, protect, and provide access to data.

AWS Lake Formation

AWS Lake Formation [23] is a service designed to streamline the creation and management of secure data lakes, making it easier to aggregate, catalog, and govern large datasets. Key features include:

- **Centralized security policies**: Provides a unified mechanism to define and enforce permissions across various data sources and consumers, simplifying access control and compliance.
- **Automated data discovery**: Employs built-in crawlers to automatically discover and classify data, ensuring that data lakes are well-organized and metadata is up-to-date.
- **Secure data sharing**: Facilitates governed access to data, allowing organizations to share datasets with internal teams or external stakeholders while maintaining strict control over permissions.

- **Integration with analytics tools**: Works seamlessly with AWS services like Athena, Redshift, and SageMaker, enabling secure and efficient data analysis.

AWS Glue Data Catalog

The AWS Glue Data Catalog [24] acts as a central metadata repository for datasets stored within AWS, serving as the backbone for data discovery and management workflows. Features include:

- **Schema discovery**: Automatically detects new datasets and infers their schema, ensuring that metadata remains current and accurate.
- **Cross-service integration**: Integrates with AWS services such as Redshift, Athena, and S3, allowing users to easily query and analyze data across different platforms.
- **Partitioning and indexing**: Organizes metadata into partitions and indexes, which significantly improve query performance for large datasets by enabling faster data retrieval.
- **Custom tagging**: Supports user-defined tags for datasets, making it easier to classify, search, and manage data assets within the organization.

3.2.4 Data Security and Privacy

AWS provides a comprehensive suite of security services and tools that help protect sensitive data, preserve privacy, and assist in meeting compliance requirements. These services provide strong mechanisms for access control, encryption, and monitoring to safeguard customer data across various use cases.

AWS Identity and Access Management (IAM)

AWS IAM [25] is a foundational security service that enables organizations to manage access to AWS resources securely and efficiently. Key features include:

- **Role-based access control**: Allows the creation of roles with specific permissions, which can then be assigned to users, groups, or applications, ensuring the principle of least privilege is upheld.
- **Multi-factor authentication (MFA)**: Adds an extra layer of security by requiring users to provide additional verification, such as a time-based one-time password, during the login process.
- **Integration with other services**: Works seamlessly across AWS services, providing unified access management and simplifying the implementation of secure architectures.
- **Policy simulator**: Offers tools to test and validate permissions, reducing the risk of misconfigurations.

AWS KMS (Key Management Service)

AWS KMS [26] is a centralized service for creating, managing, and securely storing encryption keys used across AWS. It is integral to securing data at rest and in transit. Features include:

- **Key rotation**: Automatically rotates keys based on defined schedules to reduce the risk of key compromise.
- **Integration**: Compatible with a wide range of AWS services, such as S3 for object storage encryption, RDS for database encryption, and Redshift for data warehousing.
- **Custom key policies**: Enables granular control over who can access or use keys, ensuring compliance with organizational security policies.
- **FIPS compliance**: Supports Federal Information Processing Standard (FIPS) 140-2 validated hardware security modules (HSMs) for enhanced security.

Amazon Macie

Amazon Macie [27] uses machine learning to provide data visibility and protection, helping organizations safeguard sensitive information. Key features include:

- **PII detection**: Automatically discovers and classifies personally identifiable information (PII) such as credit card numbers, social security numbers, and other sensitive data stored in S3 buckets.
- **Continuous monitoring**: Tracks data access patterns and alerts administrators to anomalies or unauthorized access attempts, helping to identify potential data breaches.
- **Compliance support**: Assists organizations in meeting regulatory requirements like GDPR, CCPA, and HIPAA by ensuring sensitive data is adequately protected and monitored.
- **Integration with AWS services**: Works seamlessly with AWS Security Hub and CloudWatch for centralized incident monitoring and response.

3.2.5 Business Intelligence and Visualization

AWS offers tools to enable insightful data visualization and business intelligence, empowering users to analyze and communicate data effectively through interactive dashboards and reports.

Amazon QuickSight

Amazon QuickSight [28] is a fully managed BI service that allows users to create engaging, interactive dashboards and reports without the need for complex infrastructure management. Features include:

- **SPICE engine**: Utilizes an in-memory computation engine optimized for rapid query execution, supporting large datasets and concurrent users.
- **Embedded analytics**: Offers the ability to integrate dashboards into web and mobile applications, providing end-users with real-time data visualization capabilities.
- **Multi-source connectivity**: Connects seamlessly to a wide range of data sources, including Redshift, S3, RDS, Snowflake, and on-premises databases.
- **Natural language querying**: Supports Amazon QuickSight Q, allowing users to ask questions in plain language and receive insights instantly.

AWS Data Exchange

AWS Data Exchange [29] simplifies the process of discovering, subscribing to, and managing third-party datasets, enabling businesses to augment their internal analytics with external insights. Features include:

- **Subscription management**: Provides a streamlined interface for discovering and subscribing to data products from a variety of providers, eliminating the complexity of manual data acquisition.
- **Secure delivery**: Ensures data is securely delivered directly to S3 buckets, maintaining integrity and compliance with organizational security policies.
- **Diverse datasets**: Offers a broad range of datasets, including financial, healthcare, and demographic data, which can be leveraged to enhance analytics and decision-making.

3.2.6 Machine Learning and AI

AWS provides an extensive suite of machine learning (ML) and artificial intelligence (AI) services that enable developers and data scientists to build, train, and deploy ML models at scale.

Amazon SageMaker

Amazon SageMaker [30] is a fully managed ML platform that supports the entire ML lifecycle, from data preparation and model building to training and deployment. Key features include:

- **Model training**: Supports distributed training and hyperparameter tuning to accelerate training processes for large datasets and complex models.
- **Pre-built algorithms**: Includes a library of optimized algorithms for common use cases, such as regression, classification, and anomaly detection, reducing the time required to build models.
- **Monitoring and debugging**: Provides tools like SageMaker Debugger to identify training issues and SageMaker Model Monitor to continuously track model performance in production.
- **Notebook integration**: Offers fully managed Jupyter notebooks for collaborative model development and experimentation.

AWS AI Services

AWS AI Services [31] provide pre-trained machine learning models that let developers add the capabilities of AI to their applications without requiring deep ML expertise. These services cover a wide range of use cases, including:

- **Image recognition**: Amazon Rekognition enables image and video analysis for tasks such as object detection, facial recognition, and scene understanding. It is widely used in security, retail, and media industries.

- **Language processing**: Amazon Comprehend provides natural language processing (NLP) capabilities to extract insights, such as sentiment analysis, key phrase detection, and entity recognition, from text data.
- **Speech-to-text**: Amazon Transcribe converts spoken language into text, making it ideal for applications like transcription services, call center analytics, and real-time captioning.
- **Text-to-speech**: Amazon Polly transforms text into natural-sounding speech, enabling applications like interactive voice response systems and content accessibility.
- **Translation services**: Amazon Translate delivers fast, high-quality language translation to enable global reach for content and applications.

3.2.7 DataOps and MLOps

AWS provides strong support for operational workflows with tools aimed at improving the management of data and machine learning pipelines. These services can enable automation, orchestration, and consistency of workflows, hence greatly improving efficiency and reliability.

AWS Step Functions

AWS Step Functions [32] is a serverless orchestration service that simplifies the coordination of distributed applications and workflows. It helps in automating complex processes with minimal manual intervention. Key features include:

- **Serverless orchestration**: Allows developers to build workflows that integrate seamlessly with AWS services like Lambda, S3, and DynamoDB, without provisioning or managing servers.
- **Stateful workflows**: Tracks the state of each step in a process, enabling retry logic, error handling, and branching for complex workflows.
- **Visual workflow editor**: Provides a user-friendly interface for designing workflows using drag-and-drop tools, reducing development time.
- **Event-driven execution**: Triggers workflows in response to events, ensuring real-time processing and reducing delays.

SageMaker Pipelines

SageMaker Pipelines [30] is a purpose-built service for machine learning workflows, enabling data scientists and developers to automate and streamline the entire ML lifecycle. Key features include:

- **End-to-end automation**: Automates repetitive tasks, including data preparation, model training, tuning, and deployment, freeing up time for innovation.
- **CI/CD for ML**: Integrates with version control and deployment tools to ensure that ML workflows are consistent, repeatable, and scalable.
- **Reproducibility**: Maintains a record of all steps in a workflow, ensuring that models can be reproduced for compliance or further experimentation.

- **Integration with SageMaker Studio**: Offers a seamless interface for monitoring and managing pipelines alongside other SageMaker tools.

AWS CloudFormation

AWS CloudFormation [33] provides a robust infrastructure-as-code (IaC) solution, enabling developers and operations teams to define and manage AWS resources in a repeatable and consistent manner. Features include:

- **Template-driven provisioning**: Allows users to define infrastructure configurations using JSON or YAML templates, ensuring repeatable deployments.
- **Dependency management**: Automatically handles resource dependencies, ensuring that resources are created or updated in the correct order.
- **Change sets**: Provides visibility into proposed changes before applying them, reducing the risk of unintended modifications to infrastructure.
- **Drift detection**: Identifies discrepancies between deployed resources and the defined templates, enabling corrective actions to maintain consistency.
- **Extensibility**: Integrates with AWS service APIs and third-party tools, allowing advanced customization and automation of deployment workflows.

3.3 Microsoft Azure

Microsoft Azure [34] is an integrated cloud platform that offers a variety of computing, storage, analytics, and intelligence services provided by Microsoft Company. It has been designed for a wide range of industries and use cases, including but not limited to data analytics, machine learning, IoT, and business applications. Known for its hybrid cloud, strong integrations with Microsoft's enterprise software, such as Windows Server, SQL Server, and Office 365, Azure thus enables the businesses to scale and innovate efficiently. This section outlines key services that Azure provides in data storage, integration, governance, security, business intelligence, machine learning, and operational workflows.

3.3.1 Data Storage and Querying

Azure offers a broad set of services for storing and querying structured and unstructured data to meet a wide variety of use cases, including analytics, operational workloads, and archiving. These services are designed to be highly scalable, reliable, and integrate well with the rest of the Azure ecosystem..

Azure Blob Storage

Azure Blob Storage [35] is a highly scalable object storage service ideal for managing unstructured data like documents, media files, logs, and backups. Key features include:

- **Data tiers**: Supports Hot, Cool, and Archive tiers to optimize storage costs based on the frequency of data access. Data can be seamlessly transitioned between tiers using lifecycle policies.
- **Scalability**: Handles massive datasets, scaling to petabytes or even exabytes while maintaining low-latency access.

- **Data lifecycle management**: Provides tools for automated data retention and deletion policies, simplifying compliance and cost management.
- **Integration**: Integrates natively with Azure Data Lake for advanced analytics, Azure Synapse Analytics for big data processing, and Azure Media Services for media workflows.
- **Security**: Includes encryption at rest and in transit, role-based access control, and support for private endpoints for secure data access.

Azure SQL Database

Azure SQL Database [36] is a fully managed relational database service that offers SQL Server capabilities with built-in intelligence and scalability for modern applications. Key features include:

- **Scalability**: Provides automatic scaling options to handle dynamic workloads, with support for hyperscale configurations that extend storage capacity into terabytes.
- **Intelligent performance**: Features automatic query tuning, indexed recommendations, and performance insights powered by AI to optimize query execution.
- **Advanced security**: Includes features like Transparent Data Encryption (TDE), Advanced Threat Protection (ATP), and Always Encrypted for enhanced data security.
- **Business continuity**: Offers built-in high availability with 99.99% SLA, automatic backups, point-in-time restore, and active geo-replication for disaster recovery.
- **Integration with Azure ecosystem**: Easily connects to Azure Data Factory, Power BI, and Logic Apps for streamlined data workflows and visualization.

Azure Synapse Analytics

Azure Synapse Analytics [37] is a unified platform that combines big data integration and enterprise-grade data warehousing to deliver end-to-end analytics solutions. Its key capabilities include:

- **Distributed query execution**: Leverages massively parallel processing (MPP) to execute complex queries over large datasets efficiently, ideal for handling petabytes of data.
- **Serverless options**: Enables ad-hoc querying of data in Azure Data Lake without requiring upfront infrastructure setup, reducing operational overhead.
- **Unified workspace**: Provides an integrated development environment for querying, data preparation, and pipeline management across both structured and unstructured data.
- **Tight integration**: Seamlessly works with Power BI for real-time visualization, Azure Machine Learning for predictive analytics, and Azure Data Factory for data ingestion and transformation.
- **Security and compliance**: Includes role-based access control, data masking, and encryption, ensuring enterprise-grade data governance.

Azure Cosmos DB

Azure Cosmos DB [38] is a globally distributed, multi-model database service engineered for modern, high-performance applications. It offers unmatched scalability and availability. Features include:

- **Multi-model support**: Natively supports document (NoSQL), key-value, graph, and column-family models, enabling diverse data storage needs in a single service.
- **Global distribution**: Provides multi-region replication with turnkey global distribution, ensuring ultra-low-latency access to data anywhere in the world.
- **Elastic scalability**: Automatically adjusts throughput and storage based on demand, offering cost efficiency and consistent performance.
- **Comprehensive SLAs**: Guarantees 99.999% availability, consistent low-latency responses, and high throughput performance.
- **Advanced analytics**: Integrates seamlessly with Azure Synapse Link, allowing near real-time analytics on operational data without affecting performance.
- **Security and compliance**: Offers enterprise-grade security with encryption at rest and in transit, role-based access control, and compliance with major industry standards like GDPR and HIPAA.

3.3.2 Data Integration and Transformation

Azure has various tools and services that facilitate the seamless integration or transformation of data across hybrid or cloud environments, thus enabling business users to prepare efficient and scaled workflows for dealing with data arriving from diverse sources.

Azure Data Factory

Azure Data Factory [39] is a fully managed ETL (Extract, Transform, Load) and data integration service designed for orchestrating and automating data movement and transformation. Its key capabilities include:

- **Built-in connectors**: Supports over 90 connectors for popular databases, file systems, SaaS applications, and big data platforms, including SQL Server, Salesforce, SAP, and Hadoop.
- **Data flow automation**: Offers visually designed data flows with drag-and-drop interfaces, enabling no-code and low-code development for data transformation pipelines.
- **Hybrid data movement**: Facilitates secure data transfer between on-premises systems and the Azure cloud using self-hosted Integration Runtime.
- **Flexible scheduling and monitoring**: Supports custom schedules for pipeline execution and provides real-time monitoring with alerts and performance metrics.
- **Integration with Azure services**: Works seamlessly with Azure Synapse Analytics, Azure Data Lake, and Power BI for end-to-end data processing and visualization workflows.

Azure Databricks

Azure Databricks [39] is an Apache Spark-based analytics platform optimized for large-scale data engineering, machine learning, and analytics workloads. Features include:

- **Collaborative workspace**: Provides a shared environment where data engineers, data scientists, and analysts can collaborate on notebooks, pipelines, and machine learning models in real-time.
- **Seamless Azure integration**: Integrates natively with Azure services such as Data Lake Storage, Blob Storage, Event Hubs, and Power BI for streamlined data pipelines and visualization.
- **Scalable computing**: Dynamically provisions and scales clusters, ensuring optimal resource utilization for computational workloads.
- **Optimized performance**: Includes Azure-specific enhancements like optimized connectors for improved data ingestion and retrieval speeds.
- **Security and compliance**: Supports enterprise-grade security with role-based access control, encryption, and integration with Azure Active Directory.

Azure Stream Analytics

Azure Stream Analytics [40] is a real-time data analytics service designed to process and analyze streaming data from various sources. Key features include:

- **Event hub integration**: Easily connects to Azure Event Hubs, IoT Hub, and Kafka for ingesting high-velocity data streams from applications, devices, and sensors.
- **SQL-like query language**: Provides a simple, declarative query language for filtering, transforming, and aggregating streaming data in real-time.
- **Output flexibility**: Enables processed data to be directed to multiple destinations, such as Power BI for visualization, Blob Storage for archiving, or Azure SQL Database for further analysis.
- **Custom functions**: Supports user-defined JavaScript and C# functions for implementing complex business logic.
- **Scalability and reliability**: Automatically scales to handle high-throughput workloads and provides built-in fault tolerance to ensure consistent processing.

3.3.3 Data Governance and Management

Azure offers robust tools for data governance, empowering organizations to maintain control over their data, ensure compliance with regulations, and manage resources effectively.

Azure Purview

Azure Purview [41] is a unified data governance solution that helps organizations catalog, track, and manage their data assets across various environments. Features include:

- **Data cataloging**: Automatically scans, discovers, and catalogs metadata from on-premises, multi-cloud, and SaaS sources, creating a centralized repository of data assets.

- **Data lineage tracking**: Provides end-to-end visibility into data flows, showing how data is transformed and consumed across systems.
- **Governance insights**: Offers dashboards and reports on data usage, access patterns, and compliance risks to support decision-making.
- **Policy enforcement**: Allows organizations to define and enforce data access and security policies across their data ecosystem.
- **Integration with Azure services**: Works seamlessly with Azure Synapse, Data Factory, and Power BI to enhance governance across analytics workflows.

Azure Data Share

Azure Data Share [42] is a secure data-sharing service that simplifies sharing datasets between organizations without the need for data duplication. Key features include:

- **No data duplication**: Allows sharing of data in place without creating redundant copies, reducing storage costs and simplifying management.
- **Granular control**: Provides fine-grained controls for specifying what data can be shared, with the ability to set expiration dates and usage restrictions.
- **Audit logs**: Maintains a detailed log of data-sharing activities, enabling compliance monitoring and ensuring accountability.
- **Flexible sharing models**: Supports snapshot-based sharing for static datasets and incremental updates for dynamic data.
- **Cross-tenant support**: Facilitates secure sharing of data across different Azure tenants and organizations.

Azure Policy

Azure Policy [43] is a comprehensive service for enforcing governance and compliance rules across Azure resources. Key capabilities include:

- **Policy definition**: Provides a library of built-in policy templates for common scenarios, such as enforcing tagging standards, restricting resource creation, and ensuring encryption.
- **Custom policies**: Allows the creation of custom policy definitions tailored to specific organizational requirements.
- **Automated remediation**: Supports remediation tasks to bring non-compliant resources into compliance automatically.
- **Compliance reporting**: Offers detailed reports and dashboards showing the compliance status of resources, helping organizations track and address violations.
- **Integration with Azure Blueprints**: Works with Azure Blueprints to define and deploy governance strategies across entire environments.

3.3.4 Data Security and Privacy

Azure ensures robust data security and privacy by providing advanced services designed to protect sensitive data from unauthorized access, breaches, and potential threats.

Azure Key Vault

Azure Key Vault [44] is a secure service for storing and managing sensitive information such as encryption keys, secrets, and certificates. Its key features include:

- **Centralized management**: Simplifies the management of secrets, such as API keys, passwords, and certificates, from a single, centralized location.
- **Access control**: Integrates with Azure Active Directory (AAD) to manage permissions and ensure secure access to secrets and keys, using role-based access control (RBAC).
- **Hardware Security Module (HSM)**: Supports FIPS 140-2 Level 2 compliance by leveraging hardware security modules to securely store cryptographic keys.
- **Key rotation**: Automatically rotate keys and secrets to enhance security and reduce the risk of exposure.
- **Audit logs**: Provides detailed activity logs to track key usage and access, helping to meet compliance and regulatory standards.

Azure Active Directory (AAD)

Azure Active Directory [45] is a cloud-based identity and access management service that enables secure access to applications and resources. Key features include:

- **Single sign-on (SSO)**: Enables users to access multiple applications with a single set of credentials, enhancing security and simplifying the user experience.
- **Conditional access**: Implements policy-driven access control, allowing organizations to enforce access requirements based on user location, device, and risk level.
- **Identity protection**: Uses machine learning and risk-based policies to detect and mitigate identity-related threats, such as compromised accounts or unusual sign-ins.
- **Multi-factor authentication (MFA)**: Provides additional layers of security by requiring multiple forms of verification for access.
- **B2B collaboration**: Facilitates secure external collaboration by enabling partner organizations to access resources with their own credentials, maintaining control over permissions.

Azure Security Center

Azure Security Center [46] provides unified security management, offering advanced threat protection and insights to protect workloads. Key features include:

- **Threat detection**: Utilizes machine learning algorithms and threat intelligence to detect potential vulnerabilities and security incidents across Azure resources.
- **Security recommendations**: Delivers actionable insights based on security best practices and regulatory compliance, guiding users to strengthen their security posture.

- **Compliance monitoring**: Continuously monitors resource compliance with industry standards and regulations, such as ISO, SOC 2, and GDPR.
- **Automated remediation**: Leverages automation to apply security best practices and remediates non-compliant configurations or vulnerabilities.
- **Advanced cloud defense**: Integrates with Microsoft Defender for Identity, Defender for Endpoint, and other services to provide end-to-end threat protection.

Azure Confidential Computing

Azure Confidential Computing [47] enables secure data processing within isolated environments, providing robust protection for sensitive data in use. Features include:

- **Trusted Execution Environments (TEEs)**: Uses hardware-based isolation to protect data during processing, ensuring confidentiality and integrity even when processed in shared environments.
- **Data encryption in use**: Maintains the confidentiality of data while it is being processed, as well as when it is at rest or in transit, ensuring end-to-end security.
- **Regulatory compliance**: Helps organizations comply with strict data privacy standards and regulations, such as GDPR, HIPAA, and PCI DSS, by ensuring data remains secure throughout its lifecycle.
- **Enterprise-grade security**: Integrates with Azure Security Center and Azure Key Vault to offer a comprehensive security solution for sensitive workloads.

3.3.5 Business Intelligence and Visualization

Azure provides powerful business intelligence and data visualization to drive informed decision-making and actionable insights. These services are designed to make complex data accessible and understandable for business users.

Power BI

Power BI [48] is a comprehensive business analytics service that enables users to create interactive reports and dashboards, driving data-driven decision-making. Key features include:

- **Data connectivity**: Seamlessly connects to various data sources, including Azure services such as Synapse Analytics, SQL Database, and Power BI datasets, as well as third-party sources like Google Analytics and Salesforce.
- **AI-powered insights**: Leverages machine learning and artificial intelligence to automatically detect trends, anomalies, and patterns, providing actionable insights to business users.
- **Custom visualizations**: Supports the creation of custom visualizations to meet specific business requirements, and allows integration with other applications for a consistent user experience.
- **Real-time analytics**: Facilitates real-time data updates and interactive dashboards, enabling users to monitor key performance indicators (KPIs) and other business metrics live.

- **Collaboration and sharing**: Offers robust sharing and collaboration features, allowing users to share reports, dashboards, and insights securely with colleagues or external partners.
- **Embedded analytics**: Enables businesses to embed Power BI reports and dashboards into custom applications, providing a seamless user experience.

3.3.6 Machine Learning and AI

It includes all the services that Azure provides for building, training, and deploying AI and machine learning models at scale for everything from predictive analytics to computer vision.

Azure Machine Learning

Azure Machine Learning [49] is a fully managed cloud service designed to support the end-to-end machine learning lifecycle. Key features include:

- **AutoML**: Automates the process of selecting the best model and fine-tuning hyperparameters, making it easier for users with varying levels of expertise to create effective machine learning models.
- **Experimentation**: Facilitates tracking and logging of experiments, including model performance, hyperparameter settings, and metrics, ensuring reproducibility and transparency in model development.
- **Deployment options**: Supports flexible deployment to multiple environments, including cloud, edge devices, and on-premises, enabling organizations to scale models wherever they are needed.
- **Model monitoring and management**: Tracks model performance post-deployment to ensure that models remain accurate and up to date, and allows easy retraining with new data.
- **Integration with popular frameworks**: Supports popular machine learning frameworks like TensorFlow, PyTorch, and Scikit-Learn for a wide variety of ML workloads.

Azure Cognitive Services

Azure Cognitive Services [50] is a collection of pre-built APIs and SDKs for adding intelligent capabilities to applications. These services enable developers to add AI-driven features without needing deep machine learning expertise. Key capabilities include:

- **Vision**: Includes services for image recognition, facial analysis, and object detection, enabling use cases such as content moderation, security, and accessibility.
- **Speech**: Provides speech-to-text, text-to-speech, speech translation, and speaker identification, powering applications like transcription services and voice assistants.
- **Language**: Delivers natural language processing (NLP) services such as sentiment analysis, language translation, and text analytics, enhancing applications with context-aware language capabilities.
- **Decision**: Includes services like anomaly detection and personalized recommendations, helping businesses to optimize decisions and user experiences.

- **Search**: Offers intelligent search services with image, video, and web search capabilities that can be integrated into applications for rich, AI-driven search experiences.

Azure Bot Services

Azure Bot Services [51] provides a comprehensive platform for building conversational AI applications. Key features include:

- **Multi-channel support**: Enables integration with popular messaging platforms such as Microsoft Teams, Facebook Messenger, Slack, and voice platforms like Amazon Alexa, ensuring bots can reach users across various communication channels.
- **Pre-built templates**: Offers a collection of pre-built templates to accelerate bot development for common use cases, such as customer service, FAQs, and information retrieval.
- **AI-driven interactions**: Leverages natural language processing (NLP) and machine learning to deliver intelligent, context-aware interactions with users, improving the user experience.
- **Dialog management**: Provides tools to manage complex multi-turn conversations, ensuring that bots handle context effectively and deliver relevant responses.
- **Integration with Cognitive Services**: Enhances bot capabilities with built-in AI features such as speech recognition, language understanding, and sentiment analysis, improving bot effectiveness.

3.3.7 DataOps and MLOps

With Azure, DataOps and MLOps capabilities ensure that the tools and infrastructure can manage the automation, scaling, and management of data and machine learning pipelines, hence allowing data and models to flow smoothly in their lifecycles from development into production.

Azure DevOps

Azure DevOps [52] is a suite of development tools and services designed to support continuous integration and continuous delivery (CI/CD) for software and machine learning workflows. Key features include:

- **Git repositories**: Provides cloud-based source control for versioning and collaboration on code, supporting distributed Git repositories with advanced branch management.
- **Pipeline automation**: Automates the build, test, and deployment processes for both code and machine learning models, allowing teams to continuously deliver updates to applications or models.
- **Integration with Azure ML and Data Factory**: Seamlessly integrates with Azure Machine Learning and Azure Data Factory, allowing users to automate the deployment and scaling of data pipelines and machine learning models.
- **Release management**: Supports automated release management, enabling smooth transitions from development to production environments.

- **Collaboration tools**: Includes features like dashboards, boards, and wikis for project tracking and team collaboration, facilitating agile workflows and sprint management.

Azure ML Pipelines

Azure ML Pipelines [53] automates and orchestrates machine learning workflows, enhancing collaboration and scalability for ML model development. Key features include:

- **Orchestration**: Automates the sequence of tasks in the machine learning lifecycle, including data preparation, model training, validation, and deployment, ensuring that models are developed in a repeatable and consistent manner.
- **Scalability**: Supports distributed training, allowing models to be trained on large datasets using multiple compute nodes, significantly reducing time-to-train for complex models.
- **Monitoring and tracking**: Provides real-time monitoring of pipeline performance, enabling users to track model training progress, identify bottlenecks, and optimize workflows.
- **Reproducibility**: Ensures that all steps in the pipeline are versioned, providing a consistent and reproducible environment for model development and deployment.

Azure Data Factory

Azure Data Factory [39] is a cloud-based data integration service that enables data movement and transformation for data workflows, and it plays a crucial role in DataOps by automating data pipelines. Key features include:

- **Pipeline templates**: Provides pre-built templates for commonly used data processing tasks, such as ETL, data migration, and data transformation, making it easier for users to create pipelines.
- **Real-time monitoring**: Enables users to monitor the execution of data pipelines in real-time, with detailed logs and alerts for failed tasks, helping to identify and address issues promptly.
- **Event-driven triggers**: Allows users to trigger data workflows based on specific events, such as the arrival of new data or scheduled intervals, enabling automation of data processing and reducing the need for manual intervention.
- **Integration with Azure services**: Seamlessly integrates with a wide range of Azure services like Azure SQL Database, Blob Storage, and Azure Machine Learning, ensuring smooth data flow across the ecosystem.
- **Data lineage tracking**: Supports tracking of data movement and transformation across different systems, which is essential for auditability and compliance.

3.4 Databricks

Databricks[54] is an AI-powered data engineering and analytics platform built on Apache Spark. It boasts a **Lakehouse Architecture**, which unites the flexibility of data lakes with data warehouses' performance and governance. That's why Databricks becomes the optimal solution for companies handling big-data workloads, artificial intelligence (AI), and machine learning (ML).

Originally developed by the Apache Spark authors, Databricks is available on top cloud platforms—AWS, Azure, and Google Cloud—with extensive platform integration. Its single-pane-of-glass analytics solution supports real-time data processing, enterprise AI applications, and the entire ML lifecycle.

3.4.1 Architecture & Core Components

Databricks is centered on the **Lakehouse paradigm**, offering a single data platform that harmonizes agility, governance, and performance.

a. Delta Lake [55] A fortified data storage layer that adds ACID transactions, indexing, and schema enforcement to regular data lakes. Data reliability is offered by Delta Lake and supports data versioning to prevent corruption and maintain consistency.

b. Apache Spark Compute Engine The distributed computing processing engine tuned for large data sets. It is written in a number of languages including Python, SQL, Scala, Java, and R.

c. MLflow and AI Integration Tooling integrated throughout the entire machine learning workflow—from model training and tracking to deployment. AutoML capabilities also exist with Databricks for automated model selection and feature engineering.

3.4.2 Data Processing and Analytics

Databricks is optimized for both real-time and batch data processing, thus it is geared towards big data workloads as well as AI-driven analytics.

- **Apache Spark Engine** [56] Computation scale support on thousands of nodes that enable fast data analytics of large data sets.
- **ETL Pipelines** Includes end-to-end data transformation processes—from raw data intake to structured, queryable data sets.
- **Streaming Analytics** Facilitates real-time data processing using Spark's Structured Streaming API.
- **SQL-based Analytics** Facilitates data exploration and analysis using SQL, Python, Scala, and R, allowing data engineers and analysts to have flexibility.

3.4.3 AI and Machine Learning Capabilities

Databricks has a robust AI/ML toolset, and thus it is the platform of choice for building, deploying, and managing enterprise-level AI applications.

- **MLflow Integration** [57] Offers end-to-end ML lifecycle management, including experiment tracking, model registry, and deployment management.

- **Databricks AutoML**[58] Automates feature engineering, model selection, and hyperparameter tuning, making the model development process faster.
- **Distributed AI Training** Uses Spark clusters to train machine learning models parallelly across nodes.
- **Integration with External AI Tools** Integrates smoothly with mainstream frameworks such as TensorFlow, PyTorch, XGBoost, and Hugging Face.

3.4.4 Security, Governance, and Compliance

Databricks delivers enterprise-grade security and governance features to offer secure operations, data privacy, and regulatory compliance on cloud platforms.

- **Unity Catalog** [59] Provides a central layer of governance to manage data assets, metadata, and access rules.
- **Fine-Grained Access Control** Implements role-based as well as attribute-based access controls to safeguard sensitive data.
- **Audit Logging & Data Lineage** Keeps meticulous logs of user actions, data changes, and lineage tracing for compliance and visibility.
- **Cloud-Native Security** [60] Provides integration with AWS IAM, Azure Active Directory, and Google IAM to handle identity and access management.

3.5 Snowflake

Snowflake [61] is a cloud-native data platform that is designed to handle current workloads for analytics, data warehousing, and AI/ML. It differs from traditional data warehouses as it is founded on a **multi-cluster shared data architecture** that separates compute, storage, and services. By separating architecture, elasticity, high performance, and cost efficiency at scale are made possible.

Originally built for AWS, Snowflake now supports Microsoft Azure and Google Cloud as well, enabling multi-cloud strategies. It natively handles both structured and semi-structured data, provides seamless sharing of data, and enterprise-grade security built-in, making it a market leader in cloud data platforms.

3.5.1 Architecture and Core Design

Snowflake's architecture [62] is centered around scalability, efficiency, and simplicity. It is built on top of three decoupled but integrated layers:

a. Storage Layer ""]] Handles storage of semi-structured and structured data types (e.g., Parquet, Avro, JSON) in column format and compressed. Data is stored immutably and handled entirely on the supported cloud environments.

b. Compute Layer Composed of independent virtual warehouses (i.e., isolated compute clusters) which elastically scale based on workload requirement. Every warehouse runs queries independently without interfering with others in order to ensure high concurrency and workload isolation.

c. Services Layer Manages metadata, access control, and query optimization. The layer provides native security, automated query optimization, and governance capabilities that hide operational complexity from users.

3.5.2 Data Analytics and Processing

Snowflake offers robust data analytics functionality, enabling users to process, transform, and analyze data simply and at large scale.

- **SQL-Based Analytics** Complies fully with ANSI SQL, so organizations can seamlessly transition from traditional data warehouses and leverage current SQL expertise.
- **Support for Semi-Structured Data** Natively supports JSON, Avro, and Parquet formats for flexible schema-on-read data processing without preprocessing.
- **Elastic Scaling** Workloads are dynamically dispatched across virtual warehouses to support multiple users and tasks running concurrently without performance bottlenecks.
- **Time Travel & Zero-Copy Cloning** Allows access to historical data states for recovery or auditing, and for immediate data replication without additional costs of storage.

3.5.3 AI and Machine Learning Capabilities

While Snowflake has data warehousing origins, it expands its features to support machine learning and AI workloads through the provision of built-in tools and external frameworks.

- **Snowpark** [63] A developer sandbox from which Python, Java, and Scala code can be executed natively within Snowflake, enabling high-level data preparation and model development.
- **Integration with AI/ML Tools** [64] Integrate with popular machine learning libraries such as TensorFlow, PyTorch, and H2O.ai. Supports attaching to external platforms like AWS SageMaker, Azure ML, and Google Vertex AI.
- **AutoML and Predictive Analytics** Enables model training and inference within the platform or via integration with third-party AutoML offerings for effective predictive analytics.

3.5.4 Security, Governance, and Compliance

Snowflake provides strong security and governance features, designed to protect data in multi-cloud environments and meet stringent compliance requirements [65].

- **End-to-End Encryption** All data is encrypted both in transit and at rest, ensuring secure data access and storage.
- **Role-Based Access Control (RBAC)** Implements fine-grained access permissions based on user roles and responsibilities.
- **Data Masking & Tokenization** Protects sensitive information using dynamic data masking and tokenization techniques to enforce privacy policies.
- **Compliance Certifications** [66] Snowflake complies with standard top-level standards like GDPR, HIPAA, SOC 2, and PCI DSS, and thus can be applied in highly regulated industries.

3.6 Comparative Analysis

This subsection is dedicated to addressing the research questions outlined in the previous chapter through a comprehensive comparative analysis of major cloud providers and data platforms—namely AWS, Azure, GCP, Snowflake, and Databricks. By systematically evaluating each platform across multiple dimensions, we aim to provide a nuanced understanding of their respective strengths, weaknesses, and suitability for various organizational contexts.

The comparison is structured around five key criteria: (1) a high-level generic comparison, (2) pricing models and cost structures, (3) technical capabilities and architectural flexibility, (4) FinOps maturity and cost optimization strategies, and (5) current market positioning alongside future relevance. These criteria have been chosen to reflect both technical and strategic considerations that are essential for informed cloud decision-making.

The results are presented in tabular form to allow for a clear, side-by-side evaluation of each platform. This format facilitates a direct comparison and highlights the differentiators that may impact an organization's cloud adoption strategy. The following sections will delve into each criterion, presenting findings and insights derived from both qualitative and quantitative analysis.

3.6.1 Comparative Summary

Attribute	**AWS**	**GCP**	**Azure**	**Databricks**	**Snowflake**
Inception Date / Place	2006 / U.S.	2008 / U.S.	2010 / U.S.	2013 / U.S.	2012 / U.S.
Company Type	Public	Public	Public	Private	Public
Main Focus	Cloud Infrastructure, Data & AI Services	Cloud Computing, Data & AI Services	Enterprise Cloud Services	Data Engineering, Data Science, Machine Learning	Data Warehousing, Business Intelligence
Architecture	Multi-service Cloud Platform	Multi-service Cloud Platform	Multi-service Cloud Platform	Data Lakehouse, Spark-based	Cloud-native Data Warehouse
Key Strengths	Broadest cloud ecosystem, fully managed services	Advanced AI/ML services, strong data analytics	Strong enterprise adoption, hybrid cloud capabilities	Spark processing, ML capabilities, data lakehouse	Scalability, performance, ease of use, SQL-centric
Community & Ecosystem	Largest cloud community	Growing cloud community	Large enterprise focus	Large & active community	Large & growing community
Key Differentiator	Highly scalable, deeply integrated with cloud-native tools	AI/ML innovation, seamless integration with Google services	Deep integration with Microsoft products, strong hybrid solutions	Strong ML capabilities, Spark-centric, MLflow integration	High-performance data warehousing, SQL-centric
Market Share & Adoption	Largest market share globally	Expanding, strong in AI/ML	Leading in enterprise adoption	Growing rapidly in data science	Dominant in cloud data warehousing
Industry Focus	Broad industry support	Strong in AI, fintech, media	Enterprise, government, healthcare	Finance, healthcare, tech	Finance, retail, healthcare
Compliance & Security	SOC 2, HIPAA, ISO 27001, FedRAMP	SOC 2, HIPAA, ISO 27001, FedRAMP	SOC 2, HIPAA, ISO 27001, FedRAMP	SOC 2, HIPAA, ISO 27001	SOC 2, HIPAA, GDPR
Data Sovereignty	Global, regional compliance options	Strong compliance offerings	Multi-region & local compliance	Flexible across cloud providers	Hosted on major cloud providers

Attribute	AWS	GCP	Azure	Databricks	Snowflake
Cost Optimization Tools	Reserved instances, Spot, Savings Plans	Committed use discounts, auto-scaling	Reserved instances, Hybrid Benefits	Cluster auto-scaling, usage monitoring	Tiered pricing, auto-scaling
Egress Costs	✓Charged	✓Charged	✓Charged	✓Charged	✓Charged
Multi-cloud Support	Limited	Strong multi-cloud options	Limited	Available across AWS, GCP, Azure	Available across cloud providers
AI/ML Capabilities	SageMaker, AI-powered services	Vertex AI, TensorFlow, AutoML	Azure ML, OpenAI integration	MLflow, deep ML/AI integration	Snowpark for ML
Developer & User Experience	CLI, SDKs, API-first design	API-driven, strong developer tools	Deep integration with Microsoft tools	Python, Spark, notebook-based	SQL-first approach, intuitive UI
Customer Support & SLAs	99.99% uptime, enterprise support	99.99% uptime, enterprise support	99.99% uptime, enterprise support	Varies based on plan	99.9% uptime, premium support
Future Roadmap & Innovation	Expanding AI/ML, quantum computing	Expanding AI/ML, sustainability focus	Quantum computing, AI advance-ments	Enhancing MLflow, data lakehouse	Performance optimization, Snowpark ML

3.6.2 Pricing model Comparative summary

This subsections aims to compare and provide the results of how these cloud providers price their services, showcasing the advanatges and drawbacks of each one.

Pricing Aspect	AWS	GCP	Azure	Databricks	Snowflake
Pricing Model	PAYG Consumption-based	PAYG Consumption-based	PAYG Consumption-based	PAYG Consumption-based	PAYG Consumption-based
Key Factors	Compute, storage, data transfer, services	Compute, storage, data transfer, services	Compute, storage, data transfer, services	Data transfer, storage, compute	Data transfer, storage, compute
Billing Units	Varies by service	Varies by service	Varies by service	Databricks Units (DBUs)	Warehousing Credits

Pricing Aspect	AWS	GCP	Azure	Databricks	Snowflake
Options	Pre-purchased commitments, spot pricing	Sustained use discounts, committed use contracts, spot pricing	Reserved instances, hybrid benefit, spot pricing	Pre-purchased commitments	Pre-purchased commitments, tiered subscriptions
Free Tier Availability	✓Free tier with limits	✓Free tier with limits	✓Free tier with limits	✓Free trial available	✓Free trial available
Enterprise Licensing	✓Available	✓Available	✓Available	✓Available	✓Available
Hybrid & On-Prem Pricing	✓AWS Outposts & Hybrid pricing	✓Anthos for hybrid	✓Azure Arc for hybrid	✓Hybrid with AWS, GCP, Azure	✓Snowflake Private Deployment
Auto-scaling Cost Optimization	✓Auto-scaling pricing varies	✓Auto-scaling pricing varies	✓Auto-scaling pricing varies	✓Auto-scaling pricing varies	✓Auto-scaling pricing varies
Egress Costs	✓Charged	✓Charged	✓Charged	✓Charged	✓Charged
Discount Programs	✓Savings Plans, Reserved Instances	✓Committed Use, Sustained Discounts	✓Reserved Instances, Hybrid Benefits	✓Volume-based discounts	✓Volume-based discounts
Marketplace Integration	✓AWS Marketplace	✓GCP Marketplace	✓Azure Marketplace	✓Partner integrations	✓Snowflake Partner Connect
Multi-cloud Pricing Strategy	✓AWS multi-cloud solutions	✓GCP multi-cloud solutions	✓Azure multi-cloud solutions	✓Available	✓Available

3.6.3 Technical capabilities comparative analysis

Capability	AWS	GCP	Azure	Databricks	Snowflake
Cloud Storage	✓	✓	✓	✓	✓
SQL Engine	✓	✓	✓	✓	✓
Data Processing (Batch/Stream)	✓	✓	✓	✓	✓
Data Integration	✓	✓	✓	✓	✓
Workflow Orchestration	✓	✓	✓	Partial	✓
Real-time Message Streaming	✓	✓	✓	✓	✗
Change Data Capture (CDC)	✓	✓	✓	✓	Partial

Capability	AWS	GCP	Azure	Databricks	Snowflake
Data Catalog	✓	✓	✓	✓	✓
Data Lake	✓	✓	✓	✓	✗
Data Warehouse	✓	✓	✓	✓	✓
Data Security & Privacy	✓	✓	✓	✓	✓
Notebook Development Environment	✓	✓	✓	✓	✓
VCS Integration	✓	✓	✓	Partial	✗
Terraform IaC Provider	✓	✓	✓	✓	✓
MLFlow Native Support	✓	API-based	✓	✓	✗
Serverless Computing	✓	✓	✓	Partial	Partial
Customer-managed VPC Cloud Deployments	✓	✓	✓	✓	✓
Auto-scaling	✓	✓	✓	✓	✓
Cost Management & Optimization	✓	✓	✓	✓	✓
Data Lineage & Governance	✓	✓	✓	✓	✓
Federated Query Engine	✓	✓	✓	✓	✓
Native AI/ML Services	✓	✓	✓	✓	✓
Data Masking & Encryption	✓	✓	✓	✓	✓
Cross-cloud Compatibility	Partial	✓	Partial	✓	✓
Data Virtualization	✓	✓	✓	Partial	✓
Graph Processing	✓	✓	✓	✗	✓
Geospatial Analytics	✓	✓	✓	✓	✓
Low-code/No-code Data Prep	✓	✓	✓	Partial	✓
Integration with BI Tools	✓	✓	✓	✓	✓
Edge Computing Support	✓	✓	✓	✗	✗
Streaming Analytics	✓	✓	✓	✓	✗
Multi-tenancy Support	✓	✓	✓	✓	✓

3.6.4 FinOps and Cost Optimization

Capability	AWS	GCP	Azure	Databricks	Snowflake
Cost Visibility & Monitoring	Yes	Yes	Yes	Yes	Yes
Budgeting & Forecasting	Yes	Yes	Yes	Yes	Yes
Chargeback & Showback	Yes	Yes	Yes	Partial	Yes
Automated Cost Optimization	Yes	Yes	Yes	Partial	Yes
Reserved Instance & Savings Plan Management	Yes	Yes	Yes	No	Yes

Capability	AWS	GCP	Azure	Databricks	Snowflake
Spot & Preemptible Instance Support	Yes	Yes	Yes	No	No
Sustainability & Carbon Footprint Tracking	Yes	Yes	Yes	No	Yes

3.6.5 Current and future relevance

Evaluation Aspect	AWS	GCP	Azure	Databricks	Snowflake
Present Relevance	Very High	High	Very High	High	Very High
Future Viability	Very High	Very High	Very High	Very High	High
Investment Value	Excellent	Strong	Excellent	Strong	Strong
Potential Risks	Potential vendor lock-in due to proprietary services	Complex pricing, smaller market share than AWS & Azure	Vendor lock-in, complexity in hybrid deploy-ments	Some reliance on Spark ecosystem, may require expertise	Proprietary SQL engine, potential vendor lock-in

Chapter 4

Analysis and Design

This chapter conducts an overall analysis and design of the solution, focusing on moving a Data and Artificial Intelligence workload to the cloud. The analysis attempts to understand the current state of the workload, define the main requirements, constraints, challenges, consider suitable cloud platforms and services. Subsequently, the design phase determines the architectural elements, component choice, and integration strategy to ensure efficient, secure, and scalable migration. The objective is to ensure that the chosen solution satisfies business needs and technical best practices and will serve as a solid foundation for the implementation outlined in the following chapter.

4.1 Analysis

Analysis is performed to describe extensively an identified problem in the current system according to Hershel and Owens. It also attempts to clearly define the nature and scope of the problem, examine its causes, and detail the implications of permitting it to stand unresolved. It then explains the justification for intervention by illustrating how the problem affects system performance, data quality, and user experience. The analysis is the foundation for recommending a successful solution in the next design cycle to ensure that any incremental improvements are data-driven, user-led, and business-led.

4.1.1 Existing System

Problem definition

In the current iteration of the SIGO system, the "cause" and "resolution" fields in support tickets frequently have data quality problems—either they are blank or are filled in inconsistently. This lack of standardization significantly compromises the integrity and reliability of the data and, in turn, hinders accurate reporting, root cause analysis, and the ability to extract meaningful insights. Of special concern is the "cause" field, which has been the most variable and error-ridden, often deficient in both clarity and completeness. These defects not only impede effective troubleshooting and long-term problem resolution but also impose friction on the workflow of support staff, ultimately reducing the overall effectiveness of the system SIGO is an internal platform designed to manage technical support tickets for various operational units within the company. It is a centralized system through which on-site report issues, detail malfunctions, and log interventions and resolutions. The system aids communication between technical support and field teams, while also serving as a historical record of operational incidents.

Objectives

The primary objectives of this work are outlined as follows:

- **Identify and analyze data quality issues** in the `cause` and `resolution` fields of SIGO support tickets, with the goal of uncovering patterns of missing, incomplete, or inconsistent entries that compromise the reliability of the system's data.
- **Implement intelligent, automated assistance** mechanisms to support users in accurately and consistently completing the `cause` and `resolution` fields. This includes leveraging machine learning models trained on historical ticket data to suggest appropriate values in real-time.
- **Enhance the overall user experience** during both ticket creation and closure phases by reducing cognitive load, minimizing manual effort, and improving the efficiency of data entry processes.
- **Generate actionable insights** for the SIGO development team, enabling informed decisions about improvements to the system's user interface, data validation workflows, and long-term information architecture.

Stakeholders

The successful implementation and impact of this solution depend on the involvement and needs of two primary stakeholder groups:

- **SIGO Users**: These are operational personnel who interact directly with the SIGO system as part of their daily work. They open, update, and close support tickets, including manual input of `cause` and `resolution` information. As end-users, they are both producers and consumers of data quality improvements. Their experience needs to be enhanced with smart suggestions and confirmation tools to nudge standardization of data input and reduce friction in their workflows.
- **SIGO Developers**: This group comprises the technical unit responsible for maintaining and evolving the SIGO platform. They will apply the learned insights from analysis to guide system development, including enhancing the user interface, incorporating machine learning aspects, and modifying data entry pipeline. Their input plays a key role in translating analysis findings into operational system-level modifications that enhance functionality as well as user experience

Requirements

The solution must fulfill a set of functional and non-functional requirements to address the identified problems and deliver value to both users and developers of the SIGO system.

Table 4.1: Requirements categorized using the FURPS+ model

Category	Requirement Description
Functionality (F)	• Suggest likely `cause` values during ticket creation based on contextual input and historical patterns. • Provide resolution recommendations based on similar past tickets and interventions. • Validate entered cause against ticket content and alert inconsistencies. • Enable re-suggestion or revision of the cause field post-intervention.
Usability (U)	• Ensure intuitive integration into the SIGO UI without disrupting existing workflows. • Provide clear, explainable suggestions that build user confidence.
Reliability (R)	• Maintain consistent behavior in generating suggestions across similar inputs. • Minimize false positives or incorrect recommendations.
Performance (P)	• Guarantee low latency in real-time suggestion delivery. • Scale efficiently with increasing ticket volumes and user interactions.
Supportability (S)	• Allow retraining of models as new data becomes available. • Enable logging and monitoring for auditing suggestion accuracy and usage.
+ (e.g., Security, Scalability)	• Implement access control and IAM policies for data privacy. • Support cloud-native scalability and managed service integration (e.g., SageMaker, Glue, Lambda).

Existing System Analysis

The current deployment of the SIGO system is an absolute reliance on **manual input** for the `cause` and `resolution` fields of help desk tickets. While this approach gives users freedom to define their issues and solutions in their own terms, it has resulted in some major shortcomings:

- **Incomplete and inconsistent data entries**: The users tend to omit the `cause` or `resolution` fields entirely or insert weak and inconsistent descriptions, causing the data to be non-uniform.
- **Lack of smart aid**: There are no built-in attributes to support users in selecting correct causes or solutions. Auto-completion, past-pattern recommendations, or real-time validation are not provided by the system.
- **Lack of quality control and automation**: Because there are no automated mechanisms to detect anomalies, recommend values, or flag inconsistencies, the system cannot furnish high-quality information. This manual, error-prone process is inefficient and undermines the integrity of ticket-based analytics and decision-making.

In general, the existing architecture limits the ability to leverage historic data for more comprehensive user support and data governance, so a compelling case can be made for the embrace of intelligent enhancements.

Use Cases

The following use cases describe the key features that the solution put forth in this proposal is intended to introduce. The use cases are designed to enhance data quality and ease the interaction of users throughout a support request's life cycle.

Automatic Cause Suggestion

When the users begin opening a new support ticket, the system dynamically analyzes the ticket's initial metadata—description, category, and historical patterns—to generate a ranked list of likely `cause` suggestions. Not only does this automate the data entry process, but it also improves consistency by drawing upon earlier-approved cases.

Active Ticket Cause Verification

Following any observed `interventions` on a ticket, the system rechecks the initially selected `cause`. When intervention data shows an inconsistency or when new more probable `causes` are found, the system will notify the user and provide updated recommendations. This verification process ensures that the `cause` field is correct throughout the life of the ticket.

Resolution Suggestions

The moment a support ticket is placed in the resolution phase, previous ticket history is utilized by the system to recommend good `resolutions`. These are offered through comparing the current issue against previously solved issues in the past, providing users well-educated, data-based recommendations for how to resolve the issue in an efficient manner.

Data Quality Feedback for Developers

Apart from assisting end-users, the system also provides the `developers` with valuable information by identifying missing, inconsistent, or poor-quality data patterns within the support ticket life cycle. Depending on the frequency and circumstances of such exceptions, sections of the SIGO interface or workflow that are most likely to be presenting issues with data entry are identified by the system. The development team makes incremental updates

to the usability of the platform, validation logic, as well as data integrity in general with this kind of information.

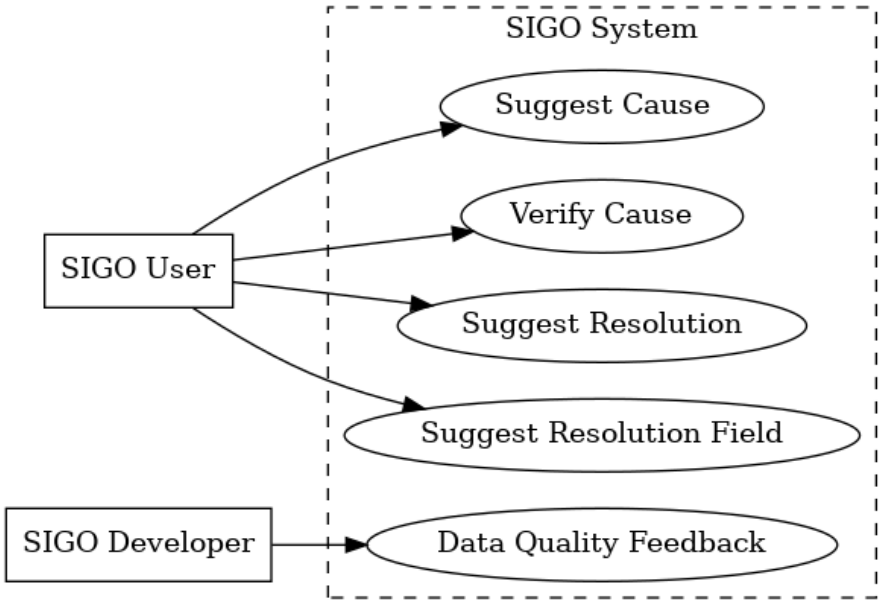


Figure 4.1: Sigo Use Cases Diagram

Required Data

To support the development of intelligent suggestions and validations for the `cause` and `resolution` fields, the solution requires access to a variety of structured and semi-structured data from the SIGO system. These data sources are essential for conducting exploratory analysis, training machine learning models, and ensuring real-time inference capabilities.

- **Avarias (Tickets)**: This dataset contains the core metadata related to support tickets, including textual problem descriptions, categorization tags, timestamps of creation and resolution, and ticket statuses. It serves as the foundational layer for identifying patterns in how issues are reported and classified.
- **Participações (Participations)**: This table tracks user interactions with each ticket—identifying who has accessed or modified the ticket, along with the actions taken and their corresponding timestamps. This helps build a temporal understanding of the ticket's lifecycle and the roles of various stakeholders.
- **Intervenções (Interventions)**: This data source captures records of technical interventions or operational steps taken to resolve reported issues. It is crucial for validating or reclassifying the cause and supports inference of likely resolutions based on observed actions.

Decision-Making Process

The system is designed to support and enhance user decision-making throughout the entire lifecycle of a support ticket. By leveraging historical data and machine learning models, the system can provide contextual recommendations and validations at key moments. The decision-support process unfolds across three critical stages:

- **During Ticket Creation:** As users begin to create a ticket, the system analyzes the inputted problem description, category, and related metadata to provide a ranked list of likely `cause` values. This intelligent assistance reduces manual effort, increases consistency across entries, and promotes the use of standardized terminology.
- **During Ticket Resolution:** At the point of resolution, the system references similar historical cases to suggest suitable `resolution` values. In parallel, it re-evaluates the previously selected `cause` in light of newly available data—such as updated descriptions or interventions—and notifies users if a more accurate classification is likely.
- **Post-Intervention Review:** Once an intervention has been performed and logged, the system reprocesses the full context of the ticket. If any discrepancies between the intervention and the original `cause` or `resolution` are detected, it proactively flags these issues and recommends corrections. This continuous validation process ensures high data fidelity and enables corrective action before ticket closure.

Business Advantages

Utilization of intelligent cause and resolution suggestion mechanisms in SIGO is expected to bring the following major benefits to the organization:

- **Data Reliability and Consistency Improvement:** Through automating and standardizing portions of the data entry process, the system delivers higher quality data in the `cause` and `resolution` fields, which is crucial for strategic decision-making accuracy, reporting, as well as trend analysis.
- **Reduced User Workload and Cognitive Load:** Auto-suggestions and validations lower the workload of users during ticket filing, minimize errors, and speed up the process. This can lead to increased productivity and user satisfaction.
- **Continuous Feedback Loop for Improvement:** Results from system usage and data quality monitoring provide actionable feedback to the UX designers and development team. The feedback allows for iterative refinement to the SIGO platform and reinforces a culture of continuous improvement.

Risks and Challenges

While the future advantage is anticipated, the project has some risks and challenges that must be properly managed:

- **Dependence on History Data Quality:** The quality and relevance of automated suggestions are heavily dependent upon the quality of historical data available. Poorly recorded or inconsistent history can reduce the performance of the model and mislead the users.
- **User Over-Reliance and Reduced Vigilance:** The danger that the users would overrely on the automated recommendations and fail to exercise critical judgment, leading to error perpetuation or inapplicable cause/resolution assignment.
- **Resistance to Change:** Changes to current workflows and user interfaces may be resisted by end-users, particularly if the new system is seen as intrusive or disruptive. Strong change management and training will be necessary to secure acceptance.

4.1.2 Cloud Migration Assessment

The final objective of this data use case is beyond data quality improvement—it is the tactical migration and modernization of the existing workload from an on-premise environment to a cloud-native environment on AWS. The migration is intended to:

- **Leverage Cloud Scalability and Flexibility:** Enable dynamic scaling of storage, compute, and processing resources to manage variable workloads and growing data sizes without the limitations of physical hardware.
- **Enhance Data Accessibility and Integration:** Consolidate operational access to heterogeneous data sources such as tickets, participations, and interventions through cloud-managed services such as Amazon S3, AWS Glue, and Amazon SageMaker.
- **Increase Operational Efficiency and Automation:** Utilize managed cloud services to automate data ingestion, transformation, model training, and inference, thereby reducing manual overhead and time-to-insight.
- **Deliver Robust Security and Compliance:** Use AWS security features—like encryption, IAM policies, and compliance certifications—to protect sensitive ticketing data during and after migration.
- **Facilitate Continuous Integration and Delivery (CI/CD):** Establish automated pipelines to release updates to data processing workflows and machine learning models, supporting agile iterations and minimizing downtime.
- **Support Cost Optimization:** Shift from upfront, capital-intensive infrastructure investment to a flexible, pay-as-you-go model that aligns costs with real usage.

The migration will finally modernize the SIGO data ecosystem to be more resilient, agile, and able to provide actionable insight at scale.

4.2 Design

This section outlines the architectural and strategic design of the proposed solution to migrate the existing AI workload from an on-premise environment to a cloud-based infrastructure using Amazon Web Services (AWS). AWS was chosen as the target platform in alignment with internal company policies and existing infrastructure standards, as it is already the primary cloud provider used within the organization. This ensures compatibility, streamlined integration, and adherence to established governance and security practices.

The design process is structured in two key stages:

- **Current State Overview:** This section outlines the proposed design for migrating the current AI workload to AWS, the company's preferred cloud provider. The existing on-premise system handles data ingestion and processing for tasks such as ticket cause classification and resolution recommendation.
- **Cloud Solution Architecture:** A detailed cloud-native design is proposed, leveraging AWS services to replicate, enhance, and scale the existing workload. This includes the use of Amazon S3 for data storage, AWS Glue for data cataloging and ETL, Amazon SageMaker for model deployment and inference, and other supporting services for orchestration, security, and CI/CD.

The goal of this design is not only to migrate the workload but to optimize it for performance, scalability, maintainability, and cost-efficiency in a modern cloud environment. Diagrams and component breakdowns are included to illustrate both the existing and target architectures clearly.

4.2.1 Current System Architecture

Architecture Overview

This subsection provides a high-level overview of the current system architecture in which the AI workload is embedded. The existing implementation is deployed in an on-premise environment and integrates with the SIGO system to extract, process, and utilize ticket data for predictive analytics. The architecture highlights the interaction between human users, the SIGO database, and the locally hosted machine learning pipeline, which includes data storage, preparation, model training, and inference.

Understanding the current structure is essential to identify performance bottlenecks, limitations in scalability, and integration challenges. This also establishes the baseline against which improvements and transformations through cloud migration—specifically to AWS—can be measured.

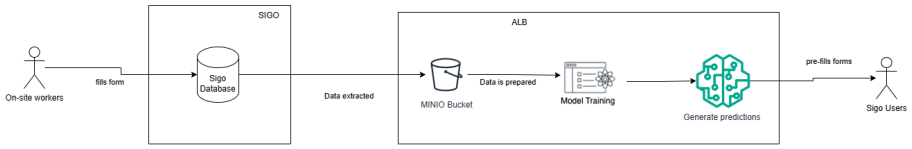


Figure 4.2: Sigo current Architecture

Data Flow Overview

The data flow begins with SIGO users interacting with the system through the ticket creation and resolution interface. When users file or update a ticket, information is recorded across three main data stores: the Ticket Database, the Intervention Database, and the Participation Database. These datasets collectively contain descriptive metadata, intervention details, and user interactions.

Once recorded, this data is ingested by the EDA (Exploratory Data Analysis) and Suggestion Engine, which processes the inputs to identify patterns, generate field suggestions (e.g., for the "cause" and "resolution" fields), and detect anomalies. This engine not only enhances the ticketing workflow in real-time but also generates quality insights. These insights are passed along to SIGO developers to inform improvements in data collection practices, interface design, and recommendation algorithms.

This data flow enables a feedback loop that promotes continuous improvement in data quality, user experience, and system intelligence.

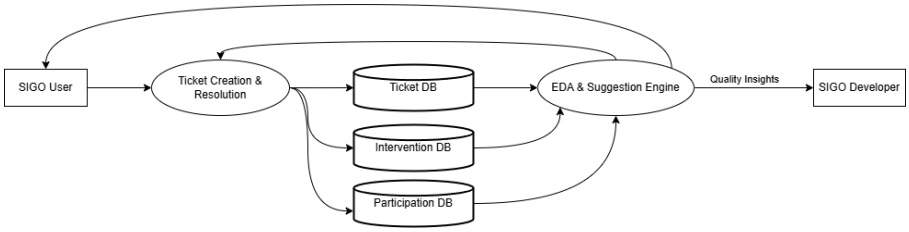


Figure 4.3: Sigo current data flow

4.2.2 Target AWS cloud Architecture

Architecture overview

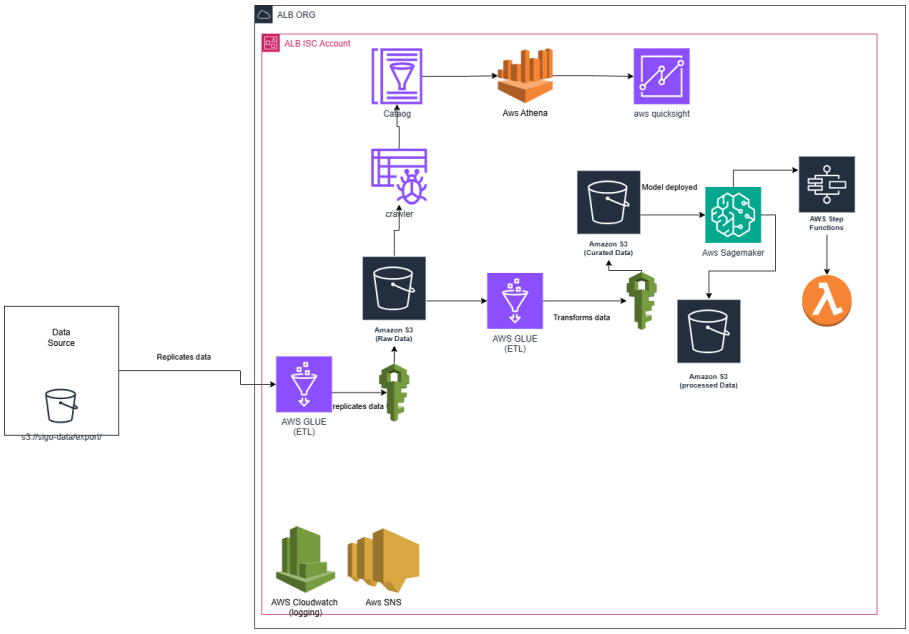


Figure 4.4: Aws Target cloud Architecture diagram

AWS Services

To successfully migrate the current AI workload from an on-premise environment to the AWS cloud, a set of purpose-built services will be adopted to ensure scalability, reliability, automation, and seamless integration. Each AWS service has been carefully selected to address a specific aspect of the workflow—from data ingestion to model inference and continuous integration.

- **Amazon S3 (Simple Storage Service)**: Serves as the central data lake in the cloud architecture. Raw ticket data, historical intervention records, and processed outputs are securely stored in S3 buckets. The service provides durable, scalable, and highly available storage, which forms the backbone of data operations in the solution.
- **AWS Glue**: Facilitates data preparation by crawling and cataloging S3-stored datasets. It also powers ETL pipelines that clean, normalize, and restructure ticket and intervention data. This ensures the data is queryable and Machine Learning (ML)-ready, reducing manual preparation overhead.

- **Amazon SageMaker**: Hosts the machine learning lifecycle—including data exploration, model training, evaluation, and inference. SageMaker enables the deployment of scalable endpoints that serve real-time predictions for "cause" and "resolution" fields. Additionally, the platform supports automated retraining and batch inference when triggered by new data ingested into S3.
- **AWS Lambda (Optional)**: Lightweight serverless functions may be used to wrap inference calls to SageMaker, enabling fast and cost-effective real-time predictions. Lambdas can also handle pre-processing (e.g., parsing ticket text) and post-processing (e.g., formatting model outputs) before passing results to the SIGO backend.
- **Amazon API Gateway**: Provides a managed layer to expose RESTful APIs to the SIGO application. These APIs interface with AWS Lambda and/or SageMaker endpoints, enabling the SIGO platform to access model suggestions in a secure, scalable, and standardized manner.
- **AWS IAM, CloudWatch, and CloudTrail**: Identity and Access Management (IAM) ensures granular access control to all AWS services. CloudWatch monitors service metrics, logs, and errors, while CloudTrail tracks all Application Programming Interface (API)-level interactions for security auditing and operational transparency.
- **Amazon QuickSight**: Delivers business intelligence and interactive dashboards by visualizing insights from processed data stored in Amazon S3 or queried via AWS Glue Data Catalog. QuickSight allows stakeholders to monitor model performance, track key metrics, and explore support trends without requiring direct access to the underlying infrastructure.

This service architecture enables a robust and modular pipeline, where new data ingested into S3 can automatically trigger Glue jobs and model retraining in SageMaker. This ensures the ML system remains up to date with evolving patterns in support tickets while minimizing operational overhead.

An AWS cloud architecture diagram should be included here to illustrate the end-to-end data flow and interactions between these services.

4.2.3 Data Design

Data Entities

The data consists of three main entities, described in Table 4.2.

Table 4.2: Description of Dataset Entities

Entity	Description
Tickets	Correspond to problem tickets that are identified and generally require some action to be resolved.
Reports	When a fault is created, one or more teams may be called upon to participate in its resolution, contributing to the closure of the ticket.
Interventions	Represent actions taken in the field. Each report may result in one or more interventions. In some cases, an intervention may be directly associated with the fault itself, without a prior report, particularly when an unplanned action is taken and needs to be recorded.

Data model

The core data model is structured around three primary entities: Avarias (tickets), Participações (user participations), and Intervenções (interventions). These entities interact as follows:

Each Avaria is always associated with at least one Participação, representing the involvement of a user in addressing the issue.

An Avaria may optionally be linked to one or more Intervenções, depending on the complexity or nature of the issue.

Additionally, a Participação can sometimes lead to an Intervenção, indicating that a user's involvement may escalate into a formal intervention process.

This relational structure captures both the mandatory and conditional interactions among entities, supporting a flexible yet traceable model of ticket resolution within the SIGO system.

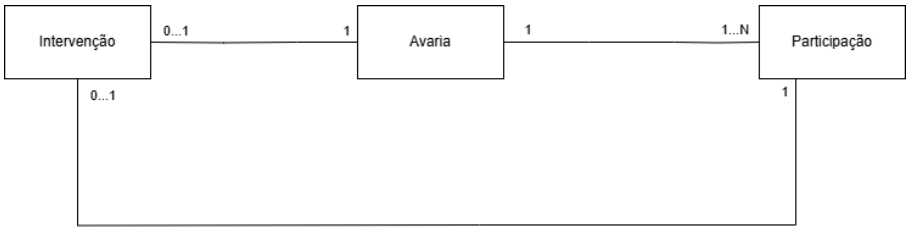


Figure 4.5: Data model Diagram

Entity relationship

The entity-relationship (ER) diagram illustrates the core data model underlying the SIGO ticketing system. At the center is the TTK entity, representing individual support tickets, each uniquely identified by a primary key UniqueID. The model establishes several one-to-many relationships between TTK and other key entities. Each ticket may have zero or more associated Pendencias (pending issues), Intervencoes (interventions performed), and Participacoes (user participations). A many-to-many relationship between Participacoes and

Intervencoes is managed through the Participacoes-Intervencoes associative entity. Additionally, the TTKs-Master-Slave table captures hierarchical relationships between tickets (e.g., follow-ups or dependencies), allowing a single ticket to be associated with multiple related tickets in a master-slave configuration. This schema effectively models the operational complexity of ticket tracking and user interactions, enabling detailed analysis of issue resolution workflows.

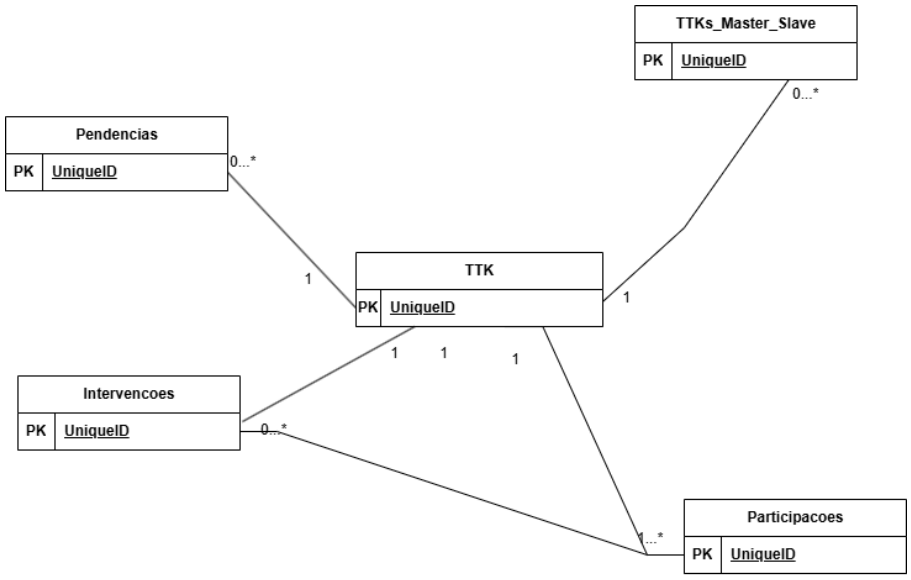


Figure 4.6: ER Diagram

Data sources

The primary data sources used in this work consist of **Parquet files**, which provide a columnar storage format optimized for efficient querying and analysis. These files contain historical and operational data related to the three core entities of the ticketing system:

- **Avarias (Tickets)**: Records of system faults or technical failures reported and tracked within the ticketing infrastructure.
- **Participações (Reports)**: User- or system-submitted reports that trigger or contribute to the creation of fault records or interventions.
- **Intervenções (Interventions)**: Logs of actions or service events performed in response to reported faults or participations.

Each Parquet file corresponds to one of these entities and serves as the foundational input for the data model designed in this study. The structured nature of the Parquet format ensures scalability and supports complex analytics over the large volume of records involved.

Analytical Layer and Dashboarding

To enable data-driven business insight and decision-making, the architecture provides an analytical layer using AWS Glue, Amazon Athena, and Amazon QuickSight. The layer takes

raw ticket data and organizes it into queryable formats to render them accessible for real-time analysis and dashboarding.

1. Glue Crawlers and Data Catalog: AWS Glue Crawlers are configured to crawl Amazon S3 directories of raw and processed ticket information in CSV or JSON files automatically. Periodically, the crawlers are used to infer and update table schemas, which are registered within the AWS Glue Data Catalog as structured metadata records. A conceptual dataset of every type—such as `avarias`, `participacoes`, and `intervencoes`—is projected into an associated Glue table with schema versioning support.

2. Query Layer with Amazon Athena: Once data tables in the Glue Data Catalog are available, Amazon Athena provides a serverless query endpoint to run SQL-based analytics against the data stored in S3. Analysts and developers can query historical support ticket trends, identify common root causes, or measure resolution times without data movement or redundancy. This layer supports ad-hoc analysis and building reusable query templates.

3. Dashboarding with Amazon QuickSight: QuickSight is also connected to Athena as a data source and used to build visual dashboards for operation and strategy monitoring. The dashboards can contain KPIs such as average resolution time, most frequent causes, ticket trends over time, and intervention impact. Access controls allow different user roles (e.g., developers, operation managers) to access corresponding visualizations securely.

This analytical chain is not only important for monitoring and reporting but also finishing the feedback cycle of the ML system's performance so stakeholders and users can continually assess data quality, system performance, and usage patterns.

Data Transformation Pipeline

The data transformation pipeline executes a series of operations to prepare ingested support ticket data for downstream analytics and AI-driven processes. Initially, raw data files are replicated from an ingestion S3 bucket to a curated destination bucket. During this replication phase, timestamp values are normalized by converting them from nanoseconds to milliseconds to ensure consistency across services such as Glue and Athena. The data is then grouped by entity type (e.g., tickets, interventions, participations) and consolidated into structured, partitioned files, typically stored in columnar Parquet format to optimize storage efficiency and query performance.

Following restructuring, the pipeline applies automated data quality checks to detect and flag anomalies, such as missing fields, inconsistent values, or schema violations. These validations ensure that only high-quality, consistent data is made available for analytical queries and model training. Where applicable, failed records are redirected to a quarantine location for manual review. The process is orchestrated using AWS Lambda and EventBridge to support scheduled execution and event-driven reprocessing. S3 versioning is enabled to support data traceability, and the design supports schema evolution through Glue's automatic schema detection. This robust transformation layer ensures that downstream services operate on clean, structured, and reliable data.

4.2.4 Security and Access Management

Security is the fundamental building block of the design in the intended cloud model. It is controlled by AWS IAM policies and roles and all the authorized components and entities have

access to specific resources. For example, IAM roles are created for Glue jobs, SageMaker notebooks, Lambda functions, and S3 buckets with the least privilege principle.

All the information stored within Amazon S3 is encrypted at rest by AWS Key Management Service (KMS). Data in transit between services is encrypted with SSL/TLS in order to encrypt sensitive information. Since ticket details may contain personally identifiable information (PII), access is traced and logged by AWS CloudTrail, and internal privacy policy adherence is managed by IAM and audit policies.

These mechanisms together ensure confidentiality, integrity, and traceability for all data operations within the cloud environment.

4.2.5 Monitoring and Observability

To provide continued insight into system activity, the architecture takes advantage of native AWS monitoring features. AWS CloudWatch collects metrics and logs from various components in the architecture, including Glue ETL jobs, Lambda functions, and SageMaker endpoints. Resource utilization and pipeline performance are monitored using custom CloudWatch dashboards.

AWS CloudTrail is enabled to capture API-level activity for auditing and debugging. For machine learning workloads, SageMaker Model Monitor identifies data drift, bias, and model performance degradation, with the option to retrain or alert if issues arise.

Log retention policies and notification are configured such that operational anomalies such as ETL failures, inference timeouts, or data ingestion delays are immediately detected and acted upon.

4.2.6 CI/CD and Automation

A CI/CD pipeline is applied to automate machine learning model and data pipeline life cycles. AWS services such as EventBridge and Lambda allow event-driven workflows, where arrivals of new data in S3 can be utilized to make models retrain or re-execute ETL jobs.

Model training and deployment are managed by SageMaker Pipelines with reproducible and auditable ML pipelines. Infrastructure changes (i.e., new Glue jobs or IAM policy updates) are checked into AWS CloudFormation templates for version control and collaboration.

These automation stages provide agility, reduce manual overhead, and allow rapid iteration with evolving data or requirements.

4.2.7 Cost Management Considerations

Cost efficiency is a key consideration in the cloud design. Data stored in S3 is managed using Intelligent-Tiering to automatically transition less frequently accessed files to cheaper storage classes. Glue ETL jobs are scheduled to run during off-peak hours and optimized to minimize execution time.

SageMaker training jobs use spot instances whenever feasible, reducing compute costs significantly. Additionally, data lifecycle policies are applied to logs and staging datasets to avoid unnecessary storage accumulation.

Monitoring and budget alarms are configured using AWS Budgets and CloudWatch to maintain visibility and control over monthly expenditures.

4.2.8 Scalability and Future Enhancements

The cloud-native architecture is future extensible and elastic. The primary elements, such as S3, Lambda, and SageMaker, are serverless or auto-scalable by default, which means that the system can ingest more ticket data without any shift in architecture.

The framework can be expanded to include other departments or domains by integrating new datasets into the pipeline with minimal configuration modification. Machine learning capability can also be augmented to predict ticket severity, estimated resolution time, or identify patterns across different support units.

Improvement in the future may be in the form of integration with AWS QuickSight for richer analytics dashboards and using AWS Step Functions to coordinate more complex workflows.

Chapter 5

Implementation

Functional deployment of the cloud solution intended to revolutionize the SIGO ticket management system is discussed in this chapter. The most important part of the implementation involves establishing a fault-tolerant data engineering pipeline that ingests, processes, and shapes operational support data using elastic AWS services. As the shift from a monolithic on-premises setup to a modular cloud-native environment occurs, the system becomes more agile, easier to keep up with, and analytically monitorable.

Implementation primarily spans the data ingestion and transformation pipeline built with Amazon S3, AWS Glue, and AWS Athena and ending with deploying interactive dashboards using Amazon QuickSight. The interactive dashboards provide stakeholders with real-time visibility into ticket data, improve operational decision-making, and expose data quality shortfalls.

While the architecture supports integration with Amazon SageMaker for machine learning-based recommendations, this chapter only deals with the data processing and analytics layers. The ML components are described in subsequent chapters or future developments.

The implementation is structured in the following sections:

- **Infrastructure Setup:** Provisioning of AWS resources, S3 bucket structure, and access configuration.
- **Data Ingestion and Storage:** Raw ticket data organization in S3, data formats, and replication strategy.
- **ETL Pipeline using AWS Glue:** Schema discovery, data transformation, and loading of Parquet-formatted structured data to the curated layer.
- **Enabling Analytical Access:** SQL-based access in Athena and dashboards creation in QuickSight.
- **Automation and Monitoring:** Pipeline reliability ensured through logging, alerting, and job monitoring.

This design forms the backbone for scalable analytics and ensuing machine learning upgrades while guaranteeing real-time usability in the form of visual and queryable insights to data.

5.1 Infrastructure Setup

For hosting the migration and processing pipeline, a cloud-based modular and scalable infrastructure was set up using Amazon Web Services (AWS). The infrastructure was created

with best-practice data isolation, security, and service interoperability in mind. The most important services developed are Amazon S3 for storage, AWS Glue for ETL processing, AWS Athena for querying, and Amazon QuickSight for visualization.

5.1.1 S3 Bucket Structuring

Amazon S3 is the core data lake where raw and processed data reside. Buckets were organized in accordance with a multi-layered architecture using common data lake patterns:

- `sigo-raw/` stores raw files that are imported from the on-premise export (CSV or JSON). It is a dumping ground with no structure imposed, allowing flexibility when ingesting disparate source data formats.
- `sigo-processed/` holds processed Parquet files structured for analytic loads. Its semi-structured organization by business domains and data use cases supports efficient access and processing.
- `sigo-logs/` holds Glue job and Lambda function logs, allowing operational monitoring and debugging.

5.1.2 IAM Roles and Permissions

Role-based access was done through AWS Identity and Access Management (IAM):

- Glue and Athena jobs were provided with scoped access to required S3 prefixes and Data Catalog entries.
- QuickSight was configured for federated access to pre-curated data.
- Logs and metrics were protected with CloudWatch log group policies and KMS encryption.

An example of a scoped IAM policy for a Glue ETL job is shown in Listing 5.1, which grants limited permissions to read from the raw data bucket and write to the processed data bucket without granting unnecessary access.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:PutObject"
    ],
    "Resource": [
      "arn:aws:s3:::sigo-raw/*",
      "arn:aws:s3:::sigo-processed/*"
    ]
  }]
}
```

Listing 5.1: Scope-limited IAM policy for Glue ETL job

5.1.3 Networking and Isolation of Resources

All services were set up in one AWS account within a private VPC for isolation. VPC endpoints were used to avoid S3 and Glue exposure to the public internet. CloudTrail was activated for infrastructure changes and attempted access monitoring.

5.1.4 Automation of Resource Provisioning

For versioning and reproducibility, the entire infrastructure components were provisioned using AWS CloudFormation templates. This includes IAM roles, Glue crawlers, scheduled jobs, and S3 buckets. This automation also makes deployment consistent across environments (i.e., production and staging).

Figure 5.1 illustrates a CloudWatch dashboard that was configured as part of the infrastructure automation. It provides visual monitoring of Glue job runtimes and S3 data ingest volumes, supporting operational visibility across environments.

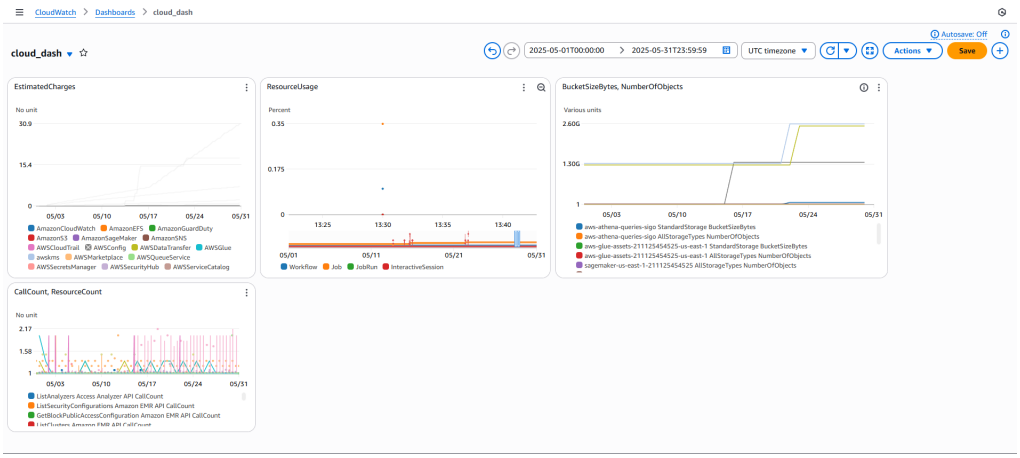


Figure 5.1: CloudWatch dashboard tracking Glue job runtimes and S3 ingest volumes.

5.2 Data Ingestion and Storage

Data ingestion is done to replicate existing ticketing data from the on-premise environment to the cloud in a structured and queryable format. It ensures all data for analytics and machine learning is available, reliable, and optimally arranged.

5.2.1 On-Premise Data Export

Data from the SIGO system is exported on a regular basis as **Parquet files**. The files are first delivered in an **unstructured format**, all within a single directory without any preorganizing.

As part of the preprocessed step, the files are subsequently **sorted into structured folders** by the type of entity contained within. The structured organization includes:

- **TTKs_SOURCE/** – Information of malfunction tickets (TTKs), including:
  - `PENDENCIAS/` – Pending issues related to tickets.
  - `SERVICOLIGACAO/` – Service connection details.
  - `TAGS/` – Tags or labels associated with tickets.
  - `TT/` – Core ticket records.
- **PARTs_SOURCE/** – Support staff participation data.
- **PARTINT_REGS_SOURCE/** – Intervention participant logs.
- **INTs_SOURCE/** – Intervention technical steps and resolution actions.
- **AGG_MASTERSLAVE_SOURCE/** – Aggregated data that may represent master-slave ticket relationships.

To automate this folder-based organization, a Python script is used to identify and move files into their appropriate categories within the S3 bucket. The script relies on naming patterns (e.g., `MEO_TTK_PENDENCIAS`) to determine the correct destination for each file. Files are moved by copying them to the correct path and then deleting the original object. Listing 5.2 shows the exact implementation of this logic.

```
import boto3

s3 = boto3.client("s3")
bucket_name = "sigo-raw-data"
source_prefix = "MEO/TTKs_SOURCE/"
categories = ["PENDENCIAS", "SERVICOLIGACAO", "TAGS", "TT"]

response = s3.list_objects_v2(Bucket=bucket_name, Prefix=source_prefix)

# Make sure there are contents
if 'Contents' in response:
    for obj in response['Contents']:
        key = obj['Key']
        filename = key.split('/')[-1]

        if filename.startswith("MEO_TTK_"):
            for category in categories:
                if category in filename:
                    new_key = f"{source_prefix}{category}/{filename}"

```

```
                    # Copy file to new location
                    s3.copy_object(
                        Bucket=bucket_name,
                        CopySource={'Bucket': bucket_name, 'Key': key},
                        Key=new_key
                    )

                    # Delete old file
                    s3.delete_object(Bucket=bucket_name, Key=key)

                    print(f"Moved {filename} to {category}/")
                    break  # Stop checking other categories for this
    file
else:
    print("No files found under the specified prefix.")
```

Listing 5.2: Sorting SIGO files within an S3 bucket based on filename patterns.

Each file contains a **timestamp field in nanoseconds (ns)** and must be **normalized** before it is used for further processing operations.

5.2.2 Landing Zone in Amazon S3

Raw Parquet files are ingested first into the `sigo-raw` bucket of Amazon S3. A partitioned directory structure is used during ingestion based on the ingestion date:

`sigo-raw/entity_name/year=YYYY/month=MM/`

This architecture supports efficient data lake operation, including scalable crawling, query pruning for performance, and lifecycle management (e.g., archiving or deletion) on an automated scale.

5.2.3 First Data Transformation

During ingestion, an AWS Glue job is executed to perform first-time transformations:

- Timestamps are shifted from nanoseconds to milliseconds.
- Schema integrity and the lack of key fields are verified for the data.
- Data is mapped into Apache Parquet format for query and space efficiency.
- Entity records are logically aggregated together (e.g., all *participações* in one).

The transformed data is saved to the `sigo-processed` bucket, with identical folder-based partitioning schema by entity and ingestion date.

An excerpt of the AWS Glue PySpark job used for this transformation is shown in Listing 5.3. This script performs initial data cleaning and structuring as part of the ETL process. It reads raw data from the `sigo-raw` S3 bucket in Parquet format, converts high-precision timestamps to standard Spark timestamps, and filters out incomplete records missing critical fields. The processed data is then saved to the `sigo-processed` bucket using a partitioning scheme based on `year` and `month`, which improves query performance downstream.

```
import sys
from awsglue.transforms import *
```

```
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from pyspark.sql.functions import col

# Initialize Glue job
args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

# Read from sigo-raw S3 bucket
raw_df = spark.read.format("parquet").load("s3://sigo-raw/entity_name/")

# Convert timestamps from nanoseconds to milliseconds
df = raw_df.withColumn("timestamp", (col("timestamp") / 1_000_000).cast(
    "timestamp"))

# Validate schema: drop rows with nulls in critical fields
required_columns = ["id", "timestamp", "description"]  # Modify as
    needed
for col_name in required_columns:
    df = df.filter(col(col_name).isNotNull())

# Write to processed zone with same partitioning scheme
df.write.mode("overwrite") \
    .partitionBy("year", "month") \
    .format("parquet") \
    .save("s3://sigo-processed/entity_name/")

job.commit()
```

Listing 5.3: AWS Glue script for initial data transformation.

5.2.4 Data Catalog and Crawling

AWS Glue Crawlers are configured to crawl raw and processed S3 destinations:

- Crawlers will automatically discover new partitions.
- Glue Data Catalog maintains up-to-date schemas for Athena and QuickSight to consume.
- **Individual crawlers are configured per entity type and zone (e.g., raw vs. processed) in order to facilitate modular updates and fine-grained control.**
- **Crawler schedules are synchronized with ingestion workflows in order to reduce schema lag.**

The crawlers have a tailored classification plan and table naming conventions specified in order to capture both the data zone (e.g., `raw_ttks`, `processed_partint`) and the associated business domain. This makes it easier to filter and govern in AWS Glue and downstream tools.

Schema evolution is handled through incremental crawling, support for adding new fields included as long as the compatibility is not lost. When incompatible changes are issued, schema versioning or table branching is implemented.

The result is a semi-structured, self-describing data lake architecture that supports downstream querying and visualization.

Athena example queries can return specific entities over time, leveraging the partition structure to improve scan performance and reduce cost.

5.3 Data Transformation and Processing

After the first ingestion, several processing steps are performed to prepare the data for analytics and model consumption. The pipeline ensures data integrity, performance, and usability for services like Amazon Athena, Amazon SageMaker, and QuickSight that rely on it.

5.3.1 Entity Normalization and Parquet Conversion

The first drastic change step is reformating all data consumed to Apache Parquet. Parquet is a columnar and compressed file format that conserves storage space and offers better scan efficiency for query engines such as Amazon Athena.

In this stage, entity-specific data sets—once received as standalone Parquet files in a flat, unstructured manner—are reorganized into logically partitioned directories. This normalization encourages separation of concerns and enables data governance. The resulting structure typically consists of:

- `sigo-processed/ttks/` – Core malfunction ticket records.
- `sigo-processed/participacoes/` – Support staff action and participations.
- `sigo-processed/intervencoes/` – Technical interventions and resolution steps.
- `sigo-processed/partint_regs/` – Intervention participations registries. - Master-slave ticket relations aggregates.

All converted files maintain their original schema by the Glue Data Catalog definition, with timestamp fields normalized from nanoseconds to milliseconds. Files are partitioned on temporal dimensions (e.g., `year=YYYY/month=MM/`), enabling efficient pruning and parallel processing.

This conversion and reorganization process improves read performance as well as laying the foundation for schema enforcement, data versioning, and downstream analytics pipelines.

5.3.2 Timestamp Normalization

Timestamps originally stored in nanoseconds (from the on-premise world) get normalized to milliseconds through a simple scalar conversion:

```
normalized_timestamp = raw_timestamp / 1,000,000
```

That makes them compatible with Glue, Athena, and time functions.

5.3.3 Data Quality Checks

Validating logic is applied to arriving records to ensure consistency, reliability, and downstream compatibility. Such data quality (DQ) tests are included:

- **Missing required fields:** Checks for required fields like `ticket_id`, `timestamp`, or `cause`.
- **Timestamp validation:** Flags entries with null, erroneous, or future-timed timestamps.
- **Schema compliance:** Checks for fields of the wrong type or structure deviations from the Glue Data Catalog schema.
- **Duplicate detection:** Identifies duplicate records based on a composite key or hash.
- **Referential checks:** Optionally verifies referenced IDs (e.g., `intervention_id` in `participacoes`) exist in upstream data sets.

Failed records are optionally redirected to a quarantine path for later inspection:

`s3://sigo-logs/failures/`

The following PySpark snippet, shown in Listing 5.4, illustrates a basic implementation of data quality checks prior to persisting the records into the validated zone. The script reads preprocessed ticket data from the `sigo-processed` bucket and applies validation rules to ensure records meet certain criteria: critical fields such as `ticket_id`, `timestamp`, and `cause` must be present, and timestamps must be in the past relative to the job's execution time.

```
from pyspark.sql.functions import col, current_timestamp

# Load preprocessed data
df = spark.read.parquet("s3://sigo-processed/ttks/")

# Filter valid records
valid_df = df.filter(
    col("ticket_id").isNotNull() &
    col("timestamp").isNotNull() &
    (col("timestamp") < current_timestamp()) &
    col("cause").isNotNull()
)

# Identify invalid records for quarantine
invalid_df = df.subtract(valid_df)

# Write valid records back to main path
valid_df.write.mode("overwrite").parquet("s3://sigo-validated/ttks/")

# Redirect failed records to quarantine
invalid_df.write.mode("overwrite").parquet("s3://sigo-logs/failures/ttks
    /")
```

Listing 5.4: Example data quality checks applied in PySpark before writing to S3.

5.3.4 Enrichment for Analytics

An enrichment step is introduced into the processed datasets to facilitate use with Amazon QuickSight and ad-hoc querying via Amazon Athena. The step is done via AWS Glue jobs and according to this logic:

- **Derived time columns:** Timestamps are mapped to human-readable fields such as `event_day`, `event_month`, `event_year`, and ISO week numbers to facilitate temporal filtering and groupings.
- **Analytical metrics:** Metrics such as `ticket_duration`, `time_to_first_intervention`, and `resolution_latency` are calculated to function as operational KPIs.
- **Entity joins:** Flat views are formed by joining related entities (e.g., tickets with participations and interventions) in order to make it easier for end users.
- **Category and label mappings:** Technical codes or enums are replaced with descriptive labels for better readability in dashboards.

The enhanced data sets are saved to a dedicated path within the `sigo-analytics` bucket and made available through Glue Data Catalog. These enhancements make business analysts and users more usable, enabling self-service analytics in BI tools irrespective of explicit knowledge of the raw schema.

5.3.5 Dataset Creation

To support Data Science workflows such as predictive modeling, clustering, and anomaly detection, these datasets are produced as part of the analytics pipeline. They are designed to be used directly, minimize preprocessing overhead, and ensure experiment consistency.

Some notable characteristics of these datasets are:

- **Pre-joined entities:** Data from multiple sources (e.g., tickets, interventions, participations) are joined on business keys, creating wide tables with complete context.
- **Feature-ready structure:** Derived features precomputed such as ticket resolution time, escalation count, technician involvement, and patterns of delay.
- **Data normalization:** Fields are cleaned, encoded (e.g., categorical to numerical), and normalized to facilitate ingestion into machine learning pipelines.
- **Balanced samples:** Class balancing (optional, e.g., binary classification use cases) and stratified sampling are done where applicable.
- **Versioning and lineage:** Each dataset is versioned with metadata that tracks its source tables, transform operations, and generation date, so that they become reproducible.

These data sets are stored in a particular `sigo-ds/` S3 bucket, organized by project or modeling goal (e.g., `churn-prediction/`, `sla-violations/`, `forecasting/`). Access is made available to Data Science teams authorized via Athena or direct S3 integration with such pieces as SageMaker, JupyterHub, or local notebooks.

This modular data set structure accelerates experimentation with stored data governance, traceability, and quality of analytics. As shown in Listing 5.5, multiple entities—such as malfunction tickets (TTKs), interventions, and participation records—are merged to construct a unified dataset suitable for downstream modeling tasks, such as SLA violation prediction.

The PySpark job begins by filtering the TTKs to a relevant time range, then joins them with intervention records and participation data based on shared keys like `ticket_id` and `intervention_id`. Additional features, such as resolution time in seconds, are derived to enrich the dataset. The final dataset is saved to a dedicated path in the `sigo-ds` zone for data science use cases.

```

from pyspark.sql.functions import col, to_date

# Load base malfunction tickets (TTKs)
ttks = spark.read.parquet("s3://sigo-processed/ttks/")
ttks_filtered = ttks.filter(
    (col("event_date") >= "2024-01-01") &
    (col("event_date") <= "2024-06-30")
)

# Load interventions
interv = spark.read.parquet("s3://sigo-processed/intervencoes/")

# Load participations
parts = spark.read.parquet("s3://sigo-processed/participacoes/")

# Join interventions to TTKs on ticket_id
ttk_with_interv = ttks_filtered.join(
    interv,
    on="ticket_id",
    how="left"
)

# Join participations
dataset = ttk_with_interv.join(
    parts,
    on=["ticket_id", "intervention_id"],  # Adjust keys as needed
    how="left"
)

# Optional feature: compute resolution time
from pyspark.sql.functions import unix_timestamp
dataset = dataset.withColumn(
    "resolution_duration_sec",
    unix_timestamp(col("end_timestamp")) - unix_timestamp(col("
    start_timestamp"))
)

# Write final dataset for data science
dataset.write.mode("overwrite").parquet("s3://sigo-ds/sla-violations/v1/
    ")
```

Listing 5.5: Merging multiple entities to build a dataset for modeling (e.g., TTKs + interventions + participations).

5.4 Amazon QuickSight Visualization and Reporting

As a means to deliver business user and technical stakeholder decision-making information, Amazon QuickSight is used as the primary Business Intelligence (BI) tool. It supports interactive dashboard creation from data residing in Amazon S3 and cataloged by AWS Glue.

5.4.1 Data Lake Integration

QuickSight connects with processed SIGO datasets via Amazon Athena. The following components are integrated:

- The data sets are accessed from Athena queries, using the Glue Data Catalog.
- Parquet-partitioning optimizes load times and query performance.
- Custom SQL is used to denormalize relations between
  textttavarias,
  texttttparticipações, and
  textttintervenções.

5.4.2 Dashboard Composition

Various dashboards were created to illustrate prominent characteristics of the support ticket life cycle:

- **Ticket Volume and Trend Analysis:** Track number of tickets over time, by cause and resolution.
- **Intervention Efficiency:** Compare average resolution time by department or intervention category.
- **Data Completeness and Quality:** Identify missing causes, erroneous timestamps, or unassigned tickets.

Each dashboard has date, department, and ticket status filters to support exploratory analysis.

5.4.3 User Access and Governance

User roles in QuickSight correspond to IAM and Active Directory policies:

- View-only users (for example, support personnel) see published dashboards.
- Datasets can be created and updated by ad-hoc users who are analysts.
- Dataset permissions are used to limit access to Sensitive PII where required.

5.4.4 Scheduled Refresh and Cost Drivers

Data refreshes run on a daily basis to keep up with the ETL pipeline:

- SPICE-based ingestion by QuickSight for faster visualization.
- Datasets refreshed by scheduled queries from Athena.

- Cost is controlled through the lowering of SPICE usage and leveraging Athena partition pruning.

This architecture offers near real-time insight into operations and is cost-effective and scalable.

5.5 Automation and Monitoring

To achieve reliability, maintainability, and cost-saving of the solution, a number of automation and monitoring features have been introduced in the architecture.

5.5.1 ETL Job Monitoring with CloudWatch

AWS Glue data transformation and replication jobs are configured with integration to Amazon CloudWatch:

- Failure and success metrics are automatically recorded.
- There are dedicated CloudWatch Alarms to indicate job failure or unusually long run times.
- Job duration and data throughput metrics are tracked for performance analysis.

Listing 5.6 shows an example of a CloudWatch alarm configuration that monitors the failure of a specific AWS Glue job. The alarm, named `GlueJobFailureAlarm`, tracks the `FailedJobs` metric within the `AWS/Glue` namespace. If one or more failures occur within a five-minute window, the alarm triggers and sends a notification to the `etl-alerts` SNS topic. The alarm is scoped to a single job identified by the `JobName` dimension set to `my-glue-etl-job`.

```
{
  "AlarmName": "GlueJobFailureAlarm",
  "AlarmDescription": "Triggers when Glue job fails",
  "ActionsEnabled": true,
  "AlarmActions": ["arn:aws:sns:us-east-1:123456789012:etl-alerts"],
  "MetricName": "FailedJobs",
  "Namespace": "AWS/Glue",
  "Statistic": "Sum",
  "Period": 300,
  "EvaluationPeriods": 1,
  "Threshold": 1,
  "ComparisonOperator": "GreaterThanOrEqualToThreshold",
  "Dimensions": [
    {
      "Name": "JobName",
      "Value": "my-glue-etl-job"
    }
  ]
}
```

Listing 5.6: CloudWatch Alarm for Glue Job Failure

5.5.2 Alerting and Notifications

To effectively respond to incidents, Amazon Simple Notification Service (SNS) is used in combination with CloudWatch:

- Email alerts on ETL failures or data quality issues are sent only to stakeholders involved in development.
- Alerts get propagated to Slack or ticketing platforms for faster triage.

This ensures that issues are identified and corrected early without manual monitoring of the pipeline, as illustrated in Figure 5.2.

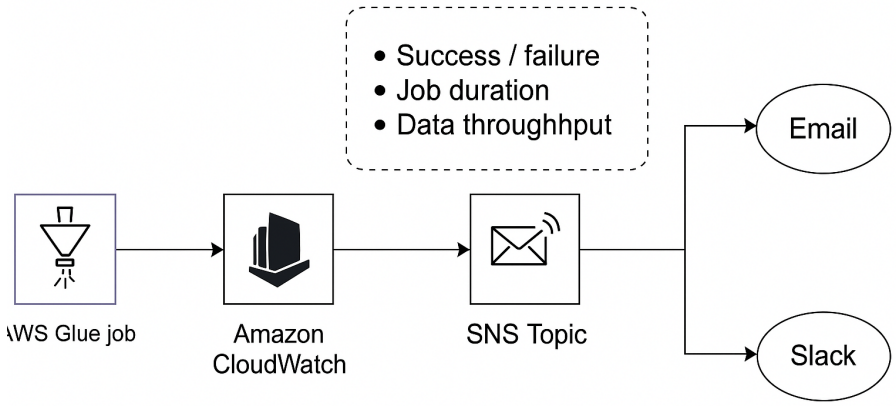


Figure 1: CloudWatch-based monitoring pipeline for ETL workflows

Figure 5.2: CloudWatch and SNS integration for automated alerting

5.5.3 Lifecycle Management and Cost Optimization

To reduce storage costs and maintain a clean data lake:

- Amazon S3 Lifecycle Rules are applied to transition older, infrequently accessed data to `INTELLIGENT_TIERING` or `GLACIER`.
- Temporary or intermediate files (e.g., staging exports or quarantined records) are configured for automatic deletion after 30 days.
- Glue job scheduling is optimized to avoid idle usage or redundant processing.

To optimize storage costs and manage data lifecycle effectively, an S3 lifecycle configuration is implemented as shown in Listing 5.7. This configuration automatically transitions files stored under the `processed/` prefix to the `INTELLIGENT_TIERING` storage class after 30 days, moves them to `GLACIER` for long-term archival after 180 days, and expires these files after one year. Additionally, temporary files under the `tmp/` prefix are configured for automatic deletion after 30 days to reduce unnecessary storage costs.

```
{
  "Rules": [
    {
      "ID": "TransitionOldFiles",
      "Prefix": "processed/",
      "Status": "Enabled",
      "Transitions": [
        {
```

```
            "Days": 30,
            "StorageClass": "INTELLIGENT_TIERING"
          },
          {
            "Days": 180,
            "StorageClass": "GLACIER"
          }
        ],
        "Expiration": {
          "Days": 365
        }
      },
      {
        "ID": "DeleteTempFiles",
        "Prefix": "tmp/",
        "Status": "Enabled",
        "Expiration": {
          "Days": 30
        }
      }
    ]
}
```

Listing 5.7: S3 Lifecycle Configuration for Cost Optimization

These measures ensure that the cloud environment remains lean, efficient, and scalable as data volume grows.

Chapter 6

Results Analysis

This chapter analyzes the results of relocating the SIGO system's AI-accelerated workload and analytical workflows from an on-premise environment to a cloud-native infrastructure on AWS. The objective of this analysis is to determine the effect of the new architecture on system performance, operational efficiency, data quality, and organizational agility.

The migration introduced some architectural improvements that introduced scalable storage in Amazon S3, data cataloging with AWS Glue, on-demand analytics with Amazon Athena, and interactive dashboards with Amazon QuickSight. All the improvements were designed not only to improve data accessibility but also to reduce manual effort in support workflows, ensure consistent classification, and introduce more velocity in data-driven decision-making.

In the following sections, qualitative and quantitative effects of this migration are presented—from technical metrics to workflow enhancement and organizational outcomes.

6.1 Better Data Transformation Pipeline

A robust, cloud-native data engineering pipeline orchestrated with AWS Glue automates all the phases of transformation—from ingestion to schema enforcement. Timestamp normalization, logical entity grouping, and Parquet conversion are now orchestrated tasks. This replaced the brittle custom Python scripts that were earlier run via cron on legacy servers.

Additionally, schema enforcement via AWS Glue jobs prevents badly formed data from reaching the processed area. This offers clean, query-ready datasets to Athena and QuickSight and reduces downstream debugging and support efforts.

6.2 Faster Access to Operational Insights

Whereas previously analysts requested IT for database dumps, stakeholders now see real-time insights via Athena queries and QuickSight dashboards embedded. This has greatly cut down the insight-to-decision cycle.

The shift to a self-service analytics model has empowered non-technical users to independently explore operational data without relying on data engineers or IT support. By integrating automated data pipelines and near real-time query access, business units can now monitor key metrics continuously, enabling more proactive and data-driven decision-making. This has also improved agility in responding to operational anomalies, seasonal trends, and customer behavior changes.

6.3 Detailed Performance Metrics

This section presents detailed performance benchmarks collected during a four-week post-migration period. Each metric includes statistical context such as averages, standard deviation, number of execution samples, and comparisons with the legacy on-premise environment. Data was gathered via AWS CloudWatch logs, Athena query history, and Glue job metrics. A total of 112 ETL jobs and 237 analytical queries were evaluated for performance and consistency.

6.3.1 ETL Job Execution Time

Description: Measures the duration of daily ETL jobs responsible for transforming incoming datasets. AWS Glue executes these jobs using distributed Spark clusters, replacing sequential Python scripts previously managed via cron.

Table 6.1: ETL job execution time comparison

Metric	Unit	On-Premise	AWS Glue	Improvement
Avg Duration	minutes	18.2 ± 2.4	7.1 ± 0.8	61% faster
P95 Duration	minutes	21.4	8.4	60.7% faster
Sample Size	runs	84	84	—

Across 84 runs for both environments, Glue consistently outperformed the on-premise setup. AWS Glue's autoscaling and Spark optimizations significantly reduced runtime variance, especially during high-volume days.

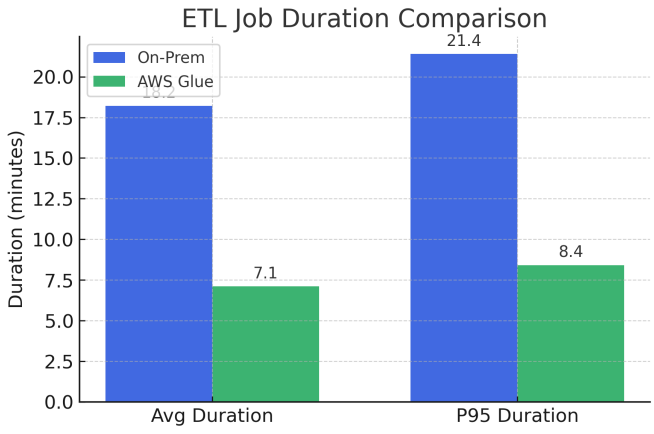


Figure 6.1: Average and P95 ETL job duration (On-Premise vs AWS Glue).

6.3.2 Athena Query Latency

Description: Captures execution latency of representative analytical queries, grouped into aggregation, join, and ranking operations. Results were sampled from 237 production queries issued by operations and engineering teams.

Table 6.2: Athena query latency (CSV vs Parquet)

Query Type	Unit	CSV Format	Parquet Format	Speedup
Simple Aggregation	sec	12.7	1.9	6.7x faster
Filtered Join	sec	19.3	4.8	4.0x faster
Top-N / Window	sec	22.6	6.3	3.6x faster
Sample Size	queries	237 total	237 total	—

By leveraging Parquet with columnar compression and partition pruning, Athena queries achieved 3–7x speed improvements and reduced data scanned by over 90% on average (see Figure 6.2). CSV-based queries often became bottlenecks during peak reporting periods.

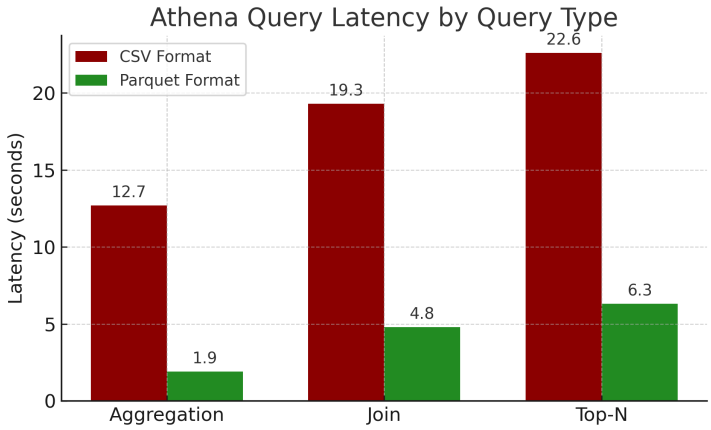


Figure 6.2: Athena query latency by query type (CSV vs Parquet).

6.3.3 Data Availability Lag

Description: Measures the time between ingestion of raw data and its availability on dashboards and reports. This is a key factor for operational responsiveness.

Table 6.3: Data availability latency post-ingestion

Metric	On-Premise	AWS (Glue + S3)	Improvement
Avg Availability Lag	24 hours	22 minutes	>98% reduction
Sample Size	28 days	28 days	—

In the legacy system, data ingestion was batched and delayed until the following day. In the cloud-native pipeline, Glue jobs trigger automatically upon file arrival, enabling dashboards in QuickSight to reflect fresh data within the same hour (see Figure 6.3).

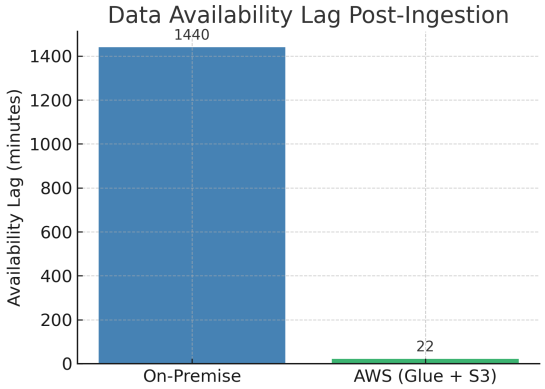


Figure 6.3: Data availability lag comparison (On-Premise vs AWS).

6.3.4 System Throughput

Description: Indicates records processed per hour during ETL execution windows, reflecting scalability and processing efficiency.

Table 6.4: Pipeline throughput and job reliability

Metric	On-Premise	AWS	Improvement
Throughput (records/hour)	48,000	205,000	4.3x faster
Peak Throughput	60,000	245,000	4x peak capacity
Job Success Rate	92.1%	99.4%	+7.3% reliability
Retry Rate	6%	<1%	Lower errors
Sample Size	112 runs	112 runs	—

AWS Glue scales horizontally with data volume and partitioning, reaching over 245,000 records/hour in high-volume batches. Consistent Spark partitioning and autoscaling contributed to both higher throughput and fewer job interruptions.

6.3.5 Pipeline Reliability (Job Success Rate)

Description: Measures the rate of successful ETL job executions, including retry frequency and primary causes of failure.

Table 6.5: ETL pipeline reliability comparison

Metric	Legacy System	AWS Glue
Success Rate	92.1%	99.4%
Retry Rate	6%	<1%
Failure Causes	Schema/connection issues	IAM/network timeouts
Sample Size	112 executions	112 executions

Legacy jobs frequently failed due to rigid schema assumptions or unavailable data sources. In the AWS environment, retries were automatically handled with exponential backoff. Logs and alerts via CloudWatch enabled faster troubleshooting, reducing overall incident resolution time.

6.4 Reliable Monitoring and Logging

The integration of CloudWatch and AWS Glue job metrics allows the team to visualize failed runs, retry attempts, and bottlenecks with clarity. Job duration-based, record count anomaly-based, or lag threshold-based alerts have made pipeline management more proactive than reactive.

This observability framework not only improves incident visibility but also supports root cause analysis by correlating logs, metrics, and events. Historical logs now serve as a feedback mechanism for optimizing job configurations and scheduling. As a result, the team has been able to reduce downtime, shorten recovery windows, and increase the overall resilience of the data workflows.

6.5 Cost Efficiency and Data Lifecycle Management

Migration leveraged features such as S3 Intelligent-Tiering and Glue job triggers on actual usage. In combination with archive and deletion lifecycle policies, storage and compute costs were optimized. This architecture allows the SIGO platform to scale storage without incurring proportional operating costs.

Additionally, the adoption of a pay-as-you-go model ensures that resources are only consumed when needed. By identifying and cleaning up idle jobs and orphaned datasets, the team has further minimized waste. This financial discipline not only reduces monthly cloud spend but also aligns IT resource usage more closely with business value.

6.6 Faster Incident Triage

With richer data and faster access, operations teams can spot emerging service problems sooner. For instance, QuickSight anomaly detection flagged an uptick in `intervenções` that was related to a firmware update, leading to a faster rollback and more uptime.

Integration with monitoring dashboards and alerting systems has enabled near real-time visibility into operational health. Teams can now correlate user behavior with backend metrics to pinpoint issues before they escalate. This has led to measurable improvements in incident response times and overall service reliability.

6.7 Enablement of Data-Driven Culture

Having queryable, structured data available has exposed analytics to a broader segment of the organization. Business users now engage with KPIs and trends on their own without requiring engineers to manually pull the data out.

The democratization of data has fostered a culture of curiosity and accountability. Teams across marketing, operations, and product now make data-backed decisions, often using

interactive dashboards and scheduled reports. This cultural shift not only improves strategic alignment but also drives innovation through evidence-based experimentation.

6.8 Comparative Metrics

Quantitative metrics reinforce the benefits of the migration. Table 6.6 summarizes key performance and efficiency improvements measured before and after the AWS implementation.

Table 6.6: Key metric improvements from cloud migration

Metric	**On-Premise**	**Post-Migration (AWS)**
ETL Runtime per batch	18 mins	7 mins
Query Latency (avg)	10–15s	1.5–4s
Manual Report Requests	4/day	<1/day
Data Access Latency	24h delay	Near real-time
Monthly Storage Cost	Fixed, high	Scalable, tiered
Incident Response Time	1.5 hours	<30 minutes

Chapter 7

Conclusion

The concluding chapter outlines the goals attained during the project, acknowledges the obstacles encountered during the implementation phase, and foresees prospective lines for further development and research.

7.1 Objectives

The primary goal of this project was to design and deploy a stable data engineering pipeline enabling scalable, maintainable, and secure data processing for the SIGO platform in a cloud-native context. This specifically involved:

- Moving current on-premise data workflows to the cloud environment.
  - *This migration was successfully completed, enabling more flexible resource management and cloud scalability.*
- Collating raw datasets (e.g., `avarias`, `participações`, and `intervenções`) into partitioned and query-optimized data lake architecture.
  - *Data was reorganized into partitioned Parquet files in S3, improving query performance and storage efficiency.*
- Developing automated ETL jobs using AWS Glue for data replication, transformation, and validation.
  - *Multiple Glue jobs were implemented, ensuring consistent, automated processing with built-in error monitoring and retry capabilities.*
- Registering processed data using AWS Glue Data Catalog for ad-hoc querying using Amazon Athena.
  - *The Glue Data Catalog was populated and maintained, allowing seamless querying and metadata management via Athena.*
- Facilitating seamless access to clean and well-organized data sets for downstream Data Science teams to leverage for machine learning experimentation and business analytics.
  - *Validated datasets were made available in curated S3 locations, enabling efficient data science workflows and faster experimentation cycles.*
- Conducting comparative analysis of cloud service providers and scalable data infrastructure architectural strategies.

  - *A thorough evaluation informed the choice of AWS services, balancing cost, scalability, and operational overhead.*

These objectives have been met. The developed system has better performance, better data availability, and better operational efficiency. It presents a good foundation for AI decision support in the SIGO platform.

7.2 Challenges and Limitations

Although the project succeeded in meeting its main objectives, there were certain limitations and challenges encountered in the development and deployment stages:

- **Learning Curve:** Most difficult of all was learning the curve that came with cloud-native technologies, particularly AWS Glue, IAM policies, and service integrations. Learning them required time and experimentation.
- **Cost Management:** Estimating and optimizing cloud cost during development proved tricky, particularly where pay-as-you-go services like Athena and S3 are utilized with diverse access patterns.
- **Tooling Complexity:** Although the AWS platform is backed with powerful tools, integration between services meant complex configuration (e.g., S3, SNS, and Glue permissions) could create insidious errors if not well managed.
- **Limited Scope of AI Integration:** While the data engineering layer was designed to support AI models, machine learning workflow (e.g., deployment in SageMaker) integration was not covered under the scope and is a long-term goal.

In spite of the limitations, the architecture remains extensible and modifiable, and learning through the process enhances the long-term sustainability of the project.

7.3 Future Work

The result achieved by this pipeline planning and migration provides a solid foundation for future growth and innovation. Some promising areas for future work are:

- **Machine Learning Pipeline Integration:** Building an end-to-end integrated MLOps pipeline with Amazon SageMaker for model training, deployment, and monitoring, leveraging the ready datasets.
- **Data Quality and Governance:** Including additional advanced data quality checks, anomaly detection, and metadata governance through AWS Glue DataBrew or third-party software.
- **CI/CD for Data Engineering:** Utilizing continuous integration and deployment pipelines (e.g., with AWS CodePipeline or Terraform) to manage Glue jobs, data schema, and infrastructure updates more automatically.
- **Improvements in Cost Optimization:** Automation of cost monitoring and optimization methods such as predictive scaling, Athena query monitoring, and intelligent S3 tier migration.
- **Dashboards Focused on Users:** The ability to allow QuickSight dashboards to integrate predictive insight and user-modified features for expanded operational awareness.

These enhancements in the future are intended to smarten up the SIGO platform, make it cost-effective, and more resilient—enabling teams to respond faster to data and business shifts.

References

[1] Matthew J. Page et al. *Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) 2020 Statement*. Available at: http://prisma-statement.org/. Mar. 2021. url: http://prisma-statement.org/.

[2] *Google Cloud Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com (visited on 01/02/2025).

[3] *Google Cloud Storage Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/storage (visited on 01/02/2025).

[4] *BigQuery*. Accessed: 2025-01-02. url: https://cloud.google.com/bigquery (visited on 01/02/2025).

[5] *Google Cloud Data Fusion Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/data-fusion (visited on 01/02/2025).

[6] *Dataflow*. Accessed: 2025-01-02. url: https://cloud.google.com/dataflow (visited on 01/02/2025).

[7] *Dataplex*. Accessed: 2025-01-02. url: https://cloud.google.com/dataplex (visited on 01/02/2025).

[8] *Google Cloud Data Catalog Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/data-catalog (visited on 01/02/2025).

[9] *Google Cloud Key Management Service (KMS) Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/kms (visited on 01/02/2025).

[10] *Google Cloud Virtual Private Cloud (VPC) Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/vpc (visited on 01/02/2025).

[11] *Looker Studio*. Accessed: 2025-01-02. url: https://lookerstudio.google.com (visited on 01/02/2025).

[12] *Vertex AI*. Accessed: 2025-01-02. url: https://cloud.google.com/vertex-ai (visited on 01/02/2025).

[13] *Google Cloud TPUs and GPUs Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/tpu (visited on 01/02/2025).

[14] *Google Cloud AI and Machine Learning APIs Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/products/ai (visited on 01/02/2025).

[15] *Google Cloud AI Workbench Documentation*. Accessed: 2025-01-02. url: https://cloud.google.com/vertex-ai-notebooks (visited on 01/02/2025).

[16] *Cloud Composer*. Accessed: 2025-01-02. url: https://cloud.google.com/composer (visited on 01/02/2025).

[17] *S3*. Accessed: 2025-01-02. url: https://aws.amazon.com/s3/ (visited on 01/02/2025).

[18] *Redshift*. Accessed: 2025-01-02. url: https://aws.amazon.com/redshift/ (visited on 01/02/2025).

[19] *Aurora*. Accessed: 2025-01-02. url: https://aws.amazon.com/rds/aurora/ (visited on 01/02/2025).

[20] *AWS Glue*. Accessed: 2025-01-02. url: https://aws.amazon.com/glue/ (visited on 01/02/2025).

[21] *AWS Data Pipeline*. Accessed: 2025-01-02. url: https://aws.amazon.com/datapipeline/ (visited on 01/02/2025).
[22] *Kinesis*. Accessed: 2025-01-02. url: https://aws.amazon.com/kinesis/ (visited on 01/02/2025).
[23] *Lake Formation*. Accessed: 2025-01-02. url: https://aws.amazon.com/lake-formation/ (visited on 01/02/2025).
[24] *AWS glue catalog*. Accessed: 2025-01-02. url: https://docs.aws.amazon.com/glue/latest/dg/catalog-and-crawler (visited on 01/02/2025).
[25] *AWS IAM Documentation*. Accessed: 2025-01-02. url: https://aws.amazon.com/pt/iam/ (visited on 01/02/2025).
[26] *AWS KMS Documentation*. Accessed: 2025-01-02. url: https://aws.amazon.com/pt/kms/ (visited on 01/02/2025).
[27] *AWS macie Documentation*. Accessed: 2025-01-02. url: https://aws.amazon.com/pt/macie/ (visited on 01/02/2025).
[28] *Amazon QuickSight*. Accessed: 2025-01-02. url: https://aws.amazon.com/quicksight/ (visited on 01/02/2025).
[29] *AWS data exchange Documentation*. Accessed: 2025-01-02. url: https://docs.aws.amazon.com/data-exchange/ (visited on 01/02/2025).
[30] *SageMaker*. Accessed: 2025-01-02. url: https://aws.amazon.com/sagemaker/ (visited on 01/02/2025).
[31] *AWS ai Documentation*. Accessed: 2025-01-02. url: https://aws.amazon.com/pt/ai/services/ (visited on 01/02/2025).
[32] *AWS Step Functions*. Accessed: 2025-01-02. url: https://aws.amazon.com/step-functions/ (visited on 01/02/2025).
[33] *AWS $Cloud_formationDocumentation$*. Accessed: 2025-01-02. url: https://aws.amazon.com/pt/cloudformation/ (visited on 01/02/2025).
[34] *Azure Documentation*. Accessed: 2025-01-02. url: https://learn.microsoft.com/en-us/azure/ (visited on 01/02/2025).
[35] *Azure Blob Storage*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/storage/blobs/ (visited on 01/02/2025).
[36] *Azure sql database Documentation*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/products/azure-sql/database (visited on 01/02/2025).
[37] *Azure Synapse Analytics*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/synapse-analytics/ (visited on 01/02/2025).
[38] *Azure Cosmos DB*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/cosmos-db/ (visited on 01/02/2025).
[39] *Azure Data Factory*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/data-factory/ (visited on 01/02/2025).
[40] *Azure Stream Analytics*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/stream-analytics/ (visited on 01/02/2025).
[41] *Azure Purview*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/purview/ (visited on 01/02/2025).
[42] *Azure Data Share*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/data-share/ (visited on 01/02/2025).
[43] *Azure policy*. Accessed: 2025-01-02. url: https://learn.microsoft.com/en-us/azure/governance/policy/overview (visited on 01/02/2025).
[44] *Azure Key Vault*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/key-vault/ (visited on 01/02/2025).

[45] *Azure Active Directory*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/active-directory/ (visited on 01/02/2025).
[46] *Azure Security Center*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/security-center/ (visited on 01/02/2025).
[47] *Azure confidential compute*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/solutions/confidential-compute (visited on 01/02/2025).
[48] *Azure Power BI*. Accessed: 2025-01-02. url: https://powerbi.microsoft.com/ (visited on 01/02/2025).
[49] *Azure Machine Learning*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/machine-learning/ (visited on 01/02/2025).
[50] *Azure Cognitive Services*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/cognitive-services/ (visited on 01/02/2025).
[51] *Azure Bot Services*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/bot-services/ (visited on 01/02/2025).
[52] *Azure DevOps*. Accessed: 2025-01-02. url: https://azure.microsoft.com/en-us/services/devops/ (visited on 01/02/2025).
[53] *Azure ML Pipelines*. Accessed: 2025-01-02. url: https://learn.microsoft.com/en-us/azure/machine-learning/how-to-create-your-first-pipeline (visited on 01/02/2025).
[54] Databricks. *The Databricks Lakehouse Platform*. 2023. url: https://www.databricks.com/product.
[55] Databricks. *What is Delta Lake?* 2023. url: https://docs.databricks.com/delta/index.html.
[56] Matei Zaharia et al. "Apache Spark: a unified engine for big data processing". In: *Communications of the ACM* 59.11 (2016), pp. 56–65.
[57] Databricks. *MLflow Documentation*. 2023. url: https://mlflow.org/docs/latest/index.html.
[58] Databricks. *Databricks AutoML: Simplifying Machine Learning*. 2023. url: https://docs.databricks.com/applications/machine-learning/automl.html.
[59] Databricks. *Unity Catalog Overview*. 2023. url: https://docs.databricks.com/data-governance/unity-catalog/index.html.
[60] Databricks. *Security and Trust at Databricks*. 2023. url: https://www.databricks.com/trust/security.
[61] Snowflake Inc. *Snowflake: The Data Cloud*. 2023. url: https://www.snowflake.com/platform-overview/.
[62] Snowflake Inc. *Snowflake Architecture Guide*. 2023. url: https://docs.snowflake.com/en/user-guide/intro-key-concepts.
[63] Snowflake Inc. *Snowpark Developer Guide*. 2023. url: https://docs.snowflake.com/en/developer-guide/snowpark.
[64] Snowflake Inc. *Machine Learning AI with Snowflake*. 2023. url: https://www.snowflake.com/blog/machine-learning-and-ai-with-the-data-cloud/.
[65] Snowflake Inc. *Security and Governance in Snowflake*. 2023. url: https://www.snowflake.com/security/.
[66] Snowflake Inc. *Compliance Certifications*. 2023. url: https://www.snowflake.com/trust/compliance/.

Appendix A

Appendice

A table illustrating potential risks and their classifications is presented below.

Risk ID	Description	Cause	Effect	Risk Owner	Probability (1-5)	Impact (1-5)	PI Score	Expected Result, No Action	Risk Response Type	Response description
	Description of the risk	Cause of the risk	Effect on the project	Name of person who monitors the risk	Group sourced rough estimate of how likely this is to occur	Rough estimate of how significant the impact of this risk	Probability multiplied by Impact	What will happen if the risk becomes an issue and no action is taken	Decision made by group on how to respond to this risk (see above in blue)	How do you know it is time to put the response into play
R1	Data Migration and Integration Risks	Differences in data structures, formats, and quality	Delays in timelines, data loss, or corruption	Data Engineer	4	4	16	Data is improperly migrated, leading to project failure	Mitigation	Use data profiling tools and validate data before migration
R2	Cost Overruns	Poor cost tracking or resource mismanagement	Exceeding budget constraints	Finance Manager	3	5	15	Resources are misallocated, resulting in halted project progress	Mitigation	Monitor costs using cloud-native tools and optimize resource usage
R3	Performance Issues	Platform limitations or improper configuration	Reduced project efficiency	Project Manager	3	4	12	Platform fails to meet required performance levels for experimentation	Mitigation	Conduct performance testing and optimize configurations
R4	Vendor Lock-In	Reliance on proprietary tools of a single vendor	Reduced flexibility and increased future costs	IT Manager	3	4	12	Unable to switch to another provider without significant time and cost investment	Avoidance	Use open-source tools and standard APIs
R5	Regulatory and Compliance Risks	Lack of awareness of applicable regulations	Legal penalties and reputational damage	Compliance Officer	2	5	10	Non-compliance results in fines and reputational harm	Mitigation	Conduct audits and use compliance monitoring tools
R6	Security and Privacy Risks	Weak authentication or misconfigured security settings	Unauthorized access or data breaches	Security Officer	4	5	20	Loss of sensitive data and trust	Mitigation	Implement encryption and identity/access management
R7	Technical Skill Gaps	Limited expertise in cloud platforms or AI tools	Inefficiencies in project execution	Team Lead	3	3	9	Poor evaluation and delayed project progress	Mitigation	Provide training and access to vendor support
R8	Model Performance and Drift	Real-world changes in data patterns	Reduced accuracy of ML models	Data Scientist	3	4	12	Decision-making suffers due to outdated model predictions	Monitoring	Set up continuous monitoring and retraining pipelines
R9	Multi-Cloud Complexity	Differences in APIs, features, and configurations	Increased time and effort for integration	Cloud Architect	4	3	12	Slow project progress due to difficulties managing multiple platforms	Mitigation	Use automation tools like Terraform
R10	Ethical and Social Implications	Lack of consideration for ethical concerns	Public backlash and reputational damage	Ethics Officer	2	4	8	Deployment of biased or unfair AI models damages public trust	Mitigation	Perform ethical reviews and test for bias in models

Figure A.1: Table of risks (general view)

The first Gantt chart depicting project milestones and timelines is shown below.

	Master thesis' Writting	**195 dias?**	**Ter 01/10/24**	**Sex 27/06/25**			**João**
	PREPD	70 dias	Ter 01/10/24	Sáb 04/01/25			
	DIMEI	125 dias?	Seg 06/01/25	Sex 27/06/25	2		
	Learning GCP Fundamentals	**11 dias?**	**Sex 01/11/24**	**Sex 15/11/24**			**João**
	Learn GCP conceps	**6 dias?**	**Sex 01/11/24**	**Sex 08/11/24**			
	IAM	2 dias	Sex 01/11/24	Seg 04/11/24			
	Networking	2 dias	Ter 05/11/24	Qua 06/11/24	6		
	Billing services	2 dias	Qui 07/11/24	Sex 08/11/24	7		
	Seting up required APIs	1 dia	Seg 11/11/24	Seg 11/11/24	5		
	Understand basic workflows and project setup	2 dias	Ter 12/11/24	Qua 13/11/24	9		
	Gains hands-on experience	2 dias	Qui 14/11/24	Sex 15/11/24	10		
	Milestone 1 - fundamentals	0 dias	Sex 15/11/24	Sex 15/11/24	11		
	Storage: Cloud Storage	**5 dias?**	**Seg 18/11/24**	**Sex 22/11/24**	**4**		**João**
	Set up buckets and organize raw data	1 dia	Seg 18/11/24	Seg 18/11/24			
	Implement lifecycle and security settings	2 dias	Ter 19/11/24	Qua 20/11/24	14		
	Test access for later stages	2 dias	Qui 21/11/24	Sex 22/11/24	15		
	Milestone 2 - Clou storage	0 dias	Sex 22/11/24	Sex 22/11/24	16		

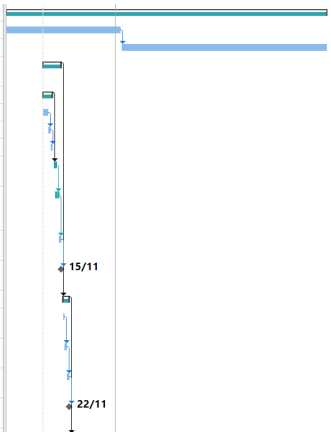
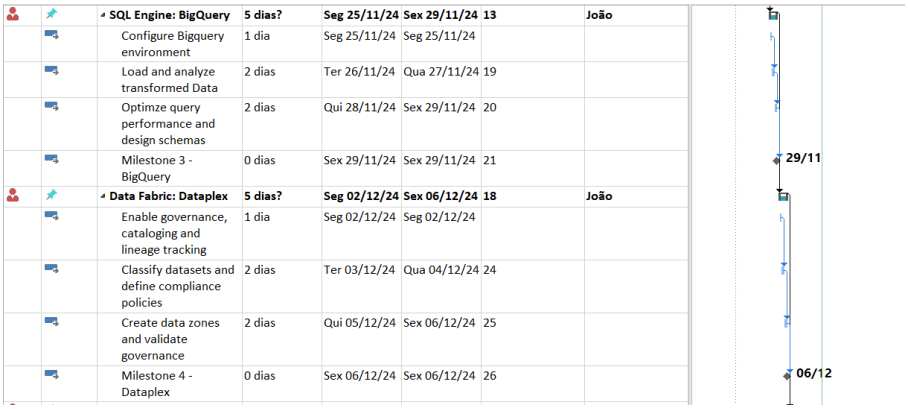


Figure A.2: Gantt Diagram 1/6

The second Gantt chart for additional project phases is provided below.



		SQL Engine: BigQuery	**5 dias?**	**Seg 25/11/24**	**Sex 29/11/24**	**13**	**João**
		Configure Bigquery environment	1 dia	Seg 25/11/24	Seg 25/11/24		
		Load and analyze transformed Data	2 dias	Ter 26/11/24	Qua 27/11/24	19	
		Optimze query performance and design schemas	2 dias	Qui 28/11/24	Sex 29/11/24	20	
		Milestone 3 - BigQuery	0 dias	Sex 29/11/24	Sex 29/11/24	21	
		Data Fabric: Dataplex	**5 dias?**	**Seg 02/12/24**	**Sex 06/12/24**	**18**	**João**
		Enable governance, cataloging and lineage tracking	1 dia	Seg 02/12/24	Seg 02/12/24		
		Classify datasets and define compliance policies	2 dias	Ter 03/12/24	Qua 04/12/24	24	
		Create data zones and validate governance	2 dias	Qui 05/12/24	Sex 06/12/24	25	
		Milestone 4 - Dataplex	0 dias	Sex 06/12/24	Sex 06/12/24	26	

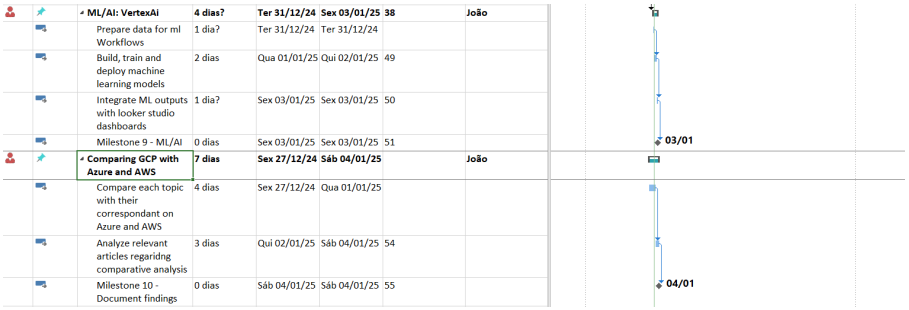
Figure A.3: Gantt Diagram 2/6

The third Gantt chart detailing further project activities follows.

		ELT+ETL (data fusion and data flow)	**4 dias?**	**Seg 09/12/24**	**Qui 12/12/24**	**23**	**João**
		Design and build pipelines base don chosen design	1 dia?	Seg 09/12/24	Seg 09/12/24		
		Extract, transform and load data into cloud storage or	2 dias	Ter 10/12/24	Qua 11/12/24	29	
		Validate Pipeline reliability	1 dia?	Qui 12/12/24	Qui 12/12/24	30	
		Milestone 5 - ELT + ETL	0 dias	Qui 12/12/24	Qui 12/12/24	31	
		Data Loss Prevention: Cloud DLP	**4 dias?**	**Sex 13/12/24**	**Qua 18/12/24**	**28**	**João**
		Scan datasets for sensitive information	1 dia?	Sex 13/12/24	Sex 13/12/24		
		Implement anonymization and de-identification rules	2 dias	Seg 16/12/24	Ter 17/12/24	34	
		Integrate DLP with Dataplex for compliance	1 dia?	Qua 18/12/24	Qua 18/12/24	35	
		Milestone 6 - DLP	0 dias	Qua 18/12/24	Qua 18/12/24	36	

Figure A.4: Gantt Diagram 3/6

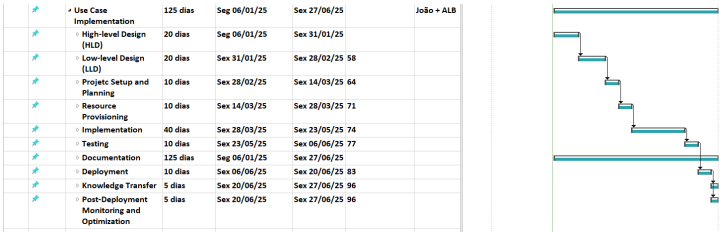
The fourth Gantt chart outlining additional scheduling information is displayed below.



		ML/AI: VertexAi	**4 dias?**	**Ter 31/12/24**	**Sex 03/01/25**	**38**	**João**
		Prepare data for ml Workflows	1 dia?	Ter 31/12/24	Ter 31/12/24		
		Build, train and deploy machine learning models	2 dias	Qua 01/01/25	Qui 02/01/25	49	
		Integrate ML outputs with looker studio dashboards	1 dia?	Sex 03/01/25	Sex 03/01/25	50	
		Milestone 9 - ML/AI	0 dias	Sex 03/01/25	Sex 03/01/25	51	
		Comparing GCP with Azure and AWS	**7 dias**	**Sex 27/12/24**	**Sáb 04/01/25**		**João**
		Compare each topic with their correspondant on Azure and AWS	4 dias	Sex 27/12/24	Qua 01/01/25		
		Analyze relevant articles regaridng comparative analysis	3 dias	Qui 02/01/25	Sáb 04/01/25	54	
		Milestone 10 - Document findings	0 dias	Sáb 04/01/25	Sáb 04/01/25	55	

Figure A.5: Gantt Diagram 4/6

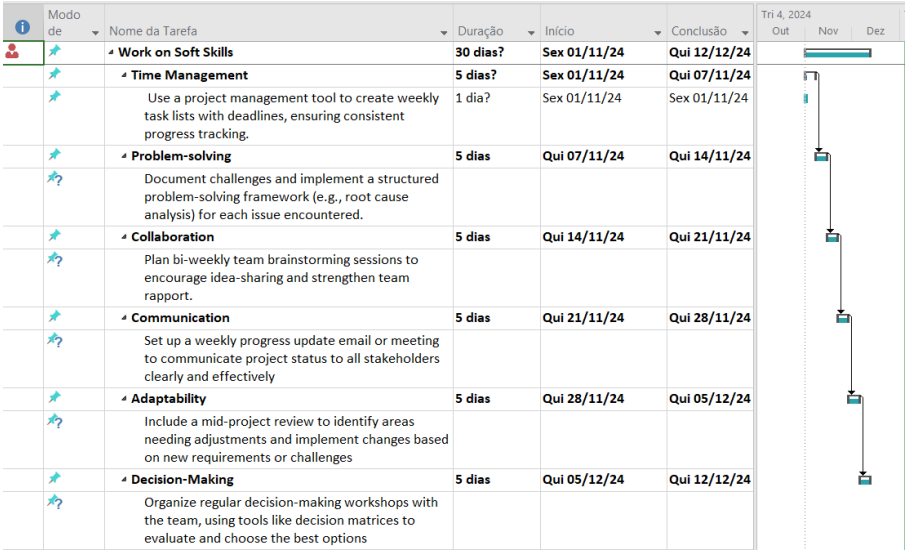
The fifth Gantt chart summarizing the remaining project stages is shown below.



	Use Case Implementation	125 dias	Seg 06/01/25	Sex 27/06/25		João + ALB
	High-level Design (HLD)	20 dias	Seg 06/01/25	Sex 31/01/25		
	Low-level Design (LLD)	20 dias	Sex 31/01/25	Sex 28/02/25	58	
	Projetc Setup and Planning	10 dias	Sex 28/02/25	Sex 14/03/25	64	
	Resource Provisioning	10 dias	Sex 14/03/25	Sex 28/03/25	71	
	Implementation	40 dias	Sex 28/03/25	Sex 23/05/25	74	
	Testing	10 dias	Sex 23/05/25	Sex 06/06/25	77	
	Documentation	125 dias	Seg 06/01/25	Sex 27/06/25		
	Deployment	10 dias	Sex 06/06/25	Sex 20/06/25	83	
	Knowledge Transfer	5 dias	Sex 20/06/25	Sex 27/06/25	96	
	Post-Deployment Monitoring and Optimization	5 dias	Sex 20/06/25	Sex 27/06/25	96	

Figure A.6: Gantt Diagram 5/6

The final Gantt chart summarizing the soft skills intended to be worked on and how.



Modo de	Nome da Tarefa	Duração	Início	Conclusão
	Work on Soft Skills	**30 dias?**	**Sex 01/11/24**	**Qui 12/12/24**
	Time Management	**5 dias?**	**Sex 01/11/24**	**Qui 07/11/24**
	Use a project management tool to create weekly task lists with deadlines, ensuring consistent progress tracking.	1 dia?	Sex 01/11/24	Sex 01/11/24
	Problem-solving	**5 dias**	**Qui 07/11/24**	**Qui 14/11/24**
	Document challenges and implement a structured problem-solving framework (e.g., root cause analysis) for each issue encountered.			
	Collaboration	**5 dias**	**Qui 14/11/24**	**Qui 21/11/24**
	Plan bi-weekly team brainstorming sessions to encourage idea-sharing and strengthen team rapport.			
	Communication	**5 dias**	**Qui 21/11/24**	**Qui 28/11/24**
	Set up a weekly progress update email or meeting to communicate project status to all stakeholders clearly and effectively			
	Adaptability	**5 dias**	**Qui 28/11/24**	**Qui 05/12/24**
	Include a mid-project review to identify areas needing adjustments and implement changes based on new requirements or challenges			
	Decision-Making	**5 dias**	**Qui 05/12/24**	**Qui 12/12/24**
	Organize regular decision-making workshops with the team, using tools like decision matrices to evaluate and choose the best options			

Figure A.7: Gantt Diagram 6/6

A.1 PREPD

The two major components that make any undertaking successful are effective project planning and skill management. This chapter gives a vivid description of the strategies and methodologies adopted in designing, organizing, and implementing the project with a view to optimizing the available skills and resources. This also includes the identification of project objectives, delegation of responsibilities, planning, and the inclusion of practices pertaining to competence management to meet the present and future needs of the project. In this chapter, the approach to be taken is that planning based on the strengths of the team is the backbone of structured project management in the delivery of desired results.

A.1.1 Skill Management

Skill management is about project success in ensuring the identification, evaluation, and development of the right competencies to meet the requirements. This section develops the key aspects of skill management: identification of the essential skills, assessing current capability, and developing strategies for enhancing these skills. These will ensure that the project attains an optimum fit of its objectives and the user's capabilities through a step-by-step procedure.

A.1.2 Skills Identification

Identification of proficient skills is the foundational step in the process of effectively managing those skills, as it aligns the competency of a person with the project requirements. During this process, the competencies required for completing the objectives would be clearly defined, and it would be analyzed where the existent skills match or not. Clearly identify the required skill so the team can direct effort toward bridging the gap for the successful execution of projects.

A.1.3 Skills Assessment

In this subsection, the key competencies, both soft and hard-skills, required for the project were identified and evaluated based on their importance and self-assessed proficiency levels. The table below provides an overview of the skills, their criticality to the project (rated on a scale of 1 to 5).

Skill	Importance	Current Level	Description
Time Management	5	2	Managing time efficiently is crucial to achieve success
Continuous Learning	5	3	Quickly mastering new cloud tools or machine learning frameworks.
Collaboration	4	3	Working efficiently in cross-functional teams
Problem-Solving	4	3	Troubleshooting issues in cloud environments or debugging AI pipelines.
Communication	3	4	Writing clear documentation for workflows and solutions
Adaptability	5	3	Adjusting to evolving project scopes, technology shifts, or unexpected challenges.
Decision-Making	4	4	Weighing trade-offs between costs, performance, and implementation complexity.
BigQuery	5	3	Data warehousing and analytics on large datasets in GCP.
Vertex AI	4	2	Building, training, and deploying ML models in GCP.
ETL Pipelines (Dataflow)	4	2	Creating data pipelines for batch and streaming workloads.
Python	4	5	Data analysis, automation, and integration with GCP SDKs.
Terraform	5	2	Provisioning and managing cloud resources as code.
Cloud IAM	5	2	Managing access control and security policies in GCP.
GDPR Compliance	5	2	Ensuring data protection and privacy standards are met.
BigQuery ML	4	2	Training and evaluating ML models directly in BigQuery.
Docker	4	3	Containerizing applications for deployment in GCP.
Dataplex	5	2	Managing and governing data lakes in GCP, ensuring data quality and compliance.
Looker	5	2	Creating visualizations and dashboards to communicate insights effectively.

Table A.1: Soft/Hard Skills Assessment Table

A.1.4 Strategy to Improve Skills

To bridge the identified skill gaps and enhance both soft and hard skills, the following strategies have been defined:

Soft Skills

- **Active Planning and Self-Reflection:** Effective planning techniques, such as setting clear goals and using productivity tools, will be employed to improve time management and overall efficiency. Periodic self-reflection will help evaluate progress and adjust strategies as needed.
- **Collaborative Engagement:** Participating in group discussions, knowledge-sharing sessions, and feedback meetings will foster better collaboration. Engaging with professional communities and forums will further enhance teamwork and networking capabilities.

Hard Skills

- **Structured Learning and Practice:** A combination of online courses, certifications, and hands-on labs will be utilized to build technical expertise. Practical projects and experimentation with new tools will solidify theoretical knowledge and provide real-world experience.
- **Experimentation and Iteration:** Regular experimentation with new technologies and iterative refinement of solutions will deepen understanding. Complex challenges will be approached incrementally, starting with foundational concepts and progressively incorporating advanced techniques.

A.1.5 Project Management

Project management is an essential component of any successful research endeavor, especially when undertaking a master thesis project. This chapter provides a comprehensive framework for managing the complexities of the thesis, ensuring that objectives are met within the allocated timeframe and resources.

The chapter is organized into several subsections, each addressing a key aspect of project management as applied to the master thesis:

- **Main Elements**: This subsection identifies the foundational components of the project, including objectives, stakeholders, and constraints.
- **Scope - Work Breakdown Structure (WBS)**: Focuses on defining the scope of the project and decomposing it into manageable tasks.
- **Project Schedule - Gantt**: Explores the development and use of a Gantt chart to visualize and track progress against deadlines.
- **Milestones**: Highlights critical junctures in the project that signify significant progress and achievement.
- **Deliverables**: Defines the tangible outcomes expected from the project, aligning them with its goals.

- **Monitoring and Controlling Procedures**: Discusses strategies for tracking performance, ensuring quality, and adapting to changes.
- **Risk Identification and Management**: Examines potential risks to the project's success and outlines mitigation strategies.

By addressing these elements, this chapter aims to equip the reader with a structured approach to managing the thesis effectively, fostering both academic rigor and practical efficiency.

A.1.6 Main Elements

This subsection outlines the foundational elements of the project, which are essential for its successful execution. These elements include the primary stakeholders, anticipated benefits, constraints, assumptions, high-level objectives, and dependencies.

Main Stakeholders: The key stakeholders for this project are:

- **Company Supervisor:** Offers practical insights, ensures the project aligns with the company's objectives, and facilitates access to required resources.
- **Academic Supervisor:** Provides academic guidance, ensures the project aligns with the thesis requirements, and offers regular feedback.

Benefits: The project is expected to yield the following benefits:

- Personal growth through hands-on experience with cloud platforms and their infrastructure.
- Gaining valuable industry experience in data engineering.
- Learning more about cloud platforms such as Google Cloud Platform (GCP).
- Producing a high-quality thesis that contributes both academically and professionally.

Constraints: The project is subject to the following constraints:

- Adhering to cloud policies and security guidelines.
- Working within the cost and budget constraints assigned to the project.
- Meeting project deadlines and university submission timelines.

Assumptions: The success of the project relies on several assumptions:

- Access to cloud platforms and their resources, primarily Google Cloud Platform (GCP).
- Weekly meetings with supervisors to assess progress and address challenges.
- Timely feedback from stakeholders to ensure smooth progress.
- Existence of use cases

High-Level Objectives: The project aims to achieve the following high-level objectives:

- Deepen knowledge of cloud platforms, particularly in data engineering contexts.
- Enhance skills in managing and analyzing data using cloud infrastructure.
- Gain expertise that bridges academic learning and practical applications in the field.

Dependencies: The project depends on several external factors, including:

- Collaborations with other stakeholders, such as team members or external partners, if required.
- Availability of necessary tools and resources on the selected cloud platforms.
- Timely input and support from the company and university supervisors.

This comprehensive overview of the main elements ensures a clear understanding of the project's framework and the foundational aspects that guide its execution.

A.1.7 Scope - WBS

The Scope - WBS subsection outlines the Work Breakdown Structure (Work Breakdown Structure (WBS)), detailing the hierarchical decomposition of project deliverables and tasks necessary to achieve the project's objectives. Figure A.8 demonstrates this.

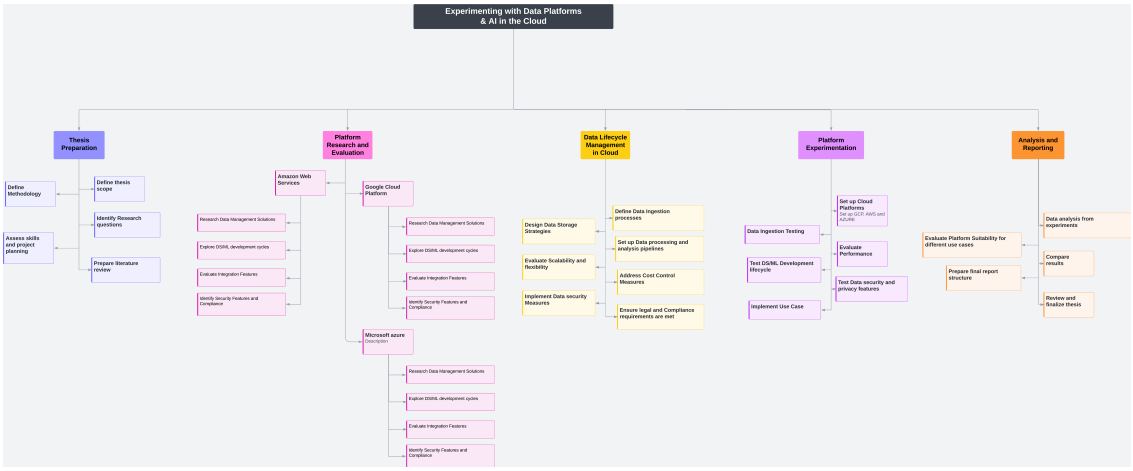


Figure A.8: Informal diagram of WBS

The WBS will be broken down into it's 5 individual stages and explained sequentially. Figure A.9 outlines the groundwork needed to start the project:

- **Define scope:** Set boundaries for what the project will and will not cover.
- **Methodology:** Develop a structured approach for conducting the research.
- **Identify research questions:** Pinpoint specific questions that the project aims to answer.
- **Assess skills and project planning:** Evaluate the team's abilities and plan tasks accordingly.
- **Prepare literature review:** Review existing research to build a strong foundation for the thesis.

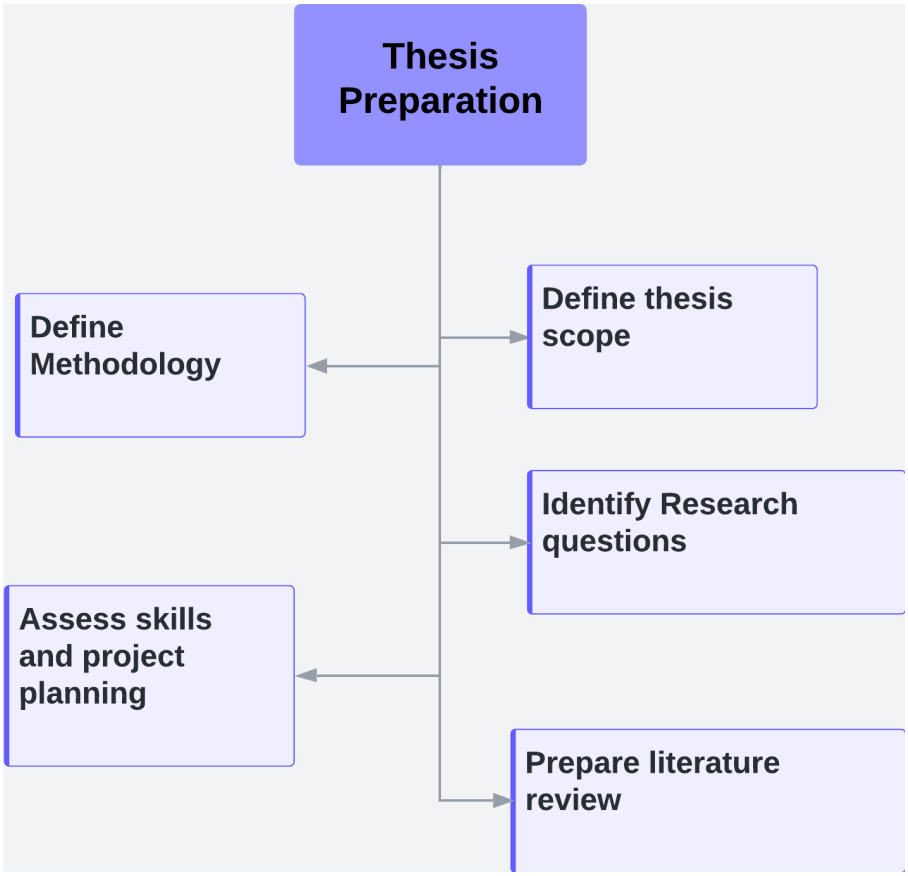


Figure A.9: Informal diagram of WBS-preparation

Figure A.10 is dedicated to understanding the capabilities and features of various cloud platforms:

- **Research data management solutions:** Explore tools and services for organizing, storing, and retrieving data.
- **Explore DS/ML development cycles:** Investigate the process of building and deploying data science and machine learning solutions.
- **Evaluate integration features:** Check the ease of integrating different tools and services within the platforms.
- **Identify security features and compliance:** Analyze the platforms' mechanisms for ensuring data privacy and regulatory compliance.

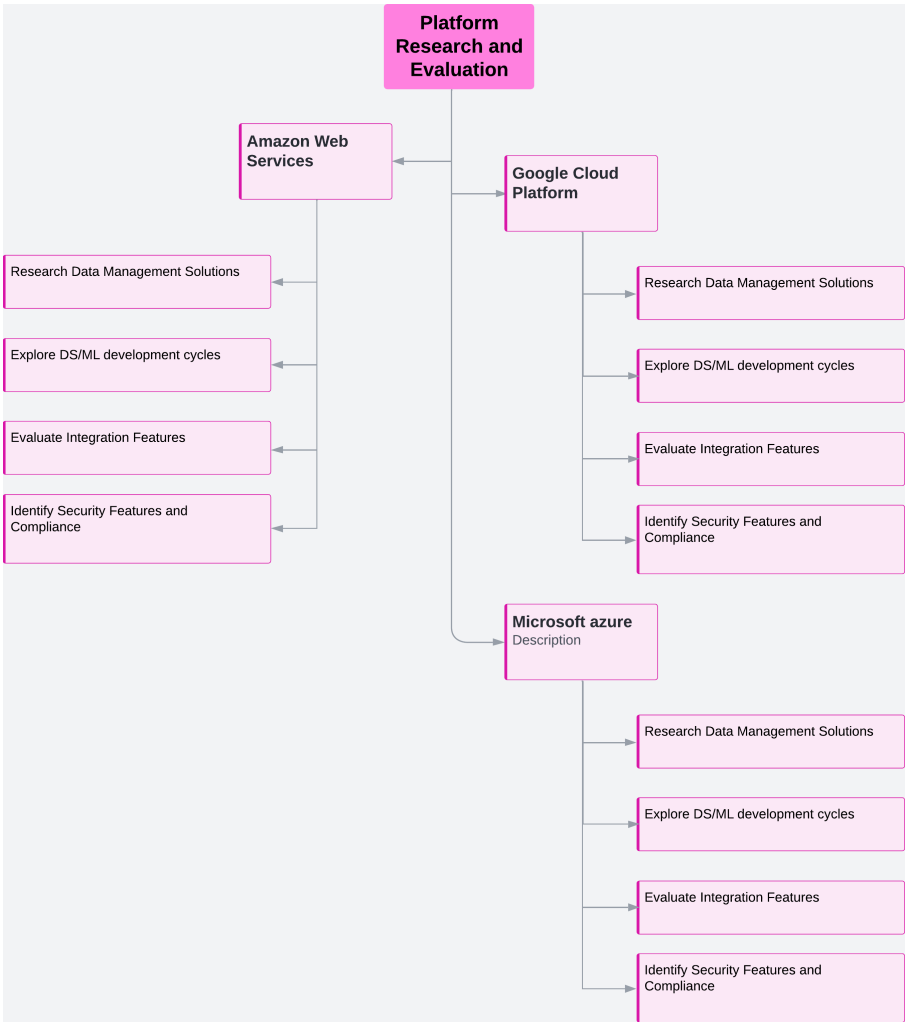


Figure A.10: Informal diagram of WBS-research

Figure A.11 addresses how to design and maintain the backend infrastructure and ensure compliance.

- **Design data storage strategies:** Define optimal methods for storing data, ensuring scalability and reliability.
- **Define data ingestion processes:** Establish protocols for importing and processing incoming data.
- **Evaluate scalability and flexibility:** Test how well the system adapts to changing workloads and requirements.
- **Implement data security measures:** Protect the system from unauthorized access and ensure data integrity.
- **Set up data processing and analysis pipelines:** Create workflows to clean, analyze, and visualize data.
- **Address cost control measures:** Monitor expenses and implement strategies to reduce costs.

- Ensure legal and compliance requirements are met: Verify that all activities adhere to relevant regulations and industry standards.

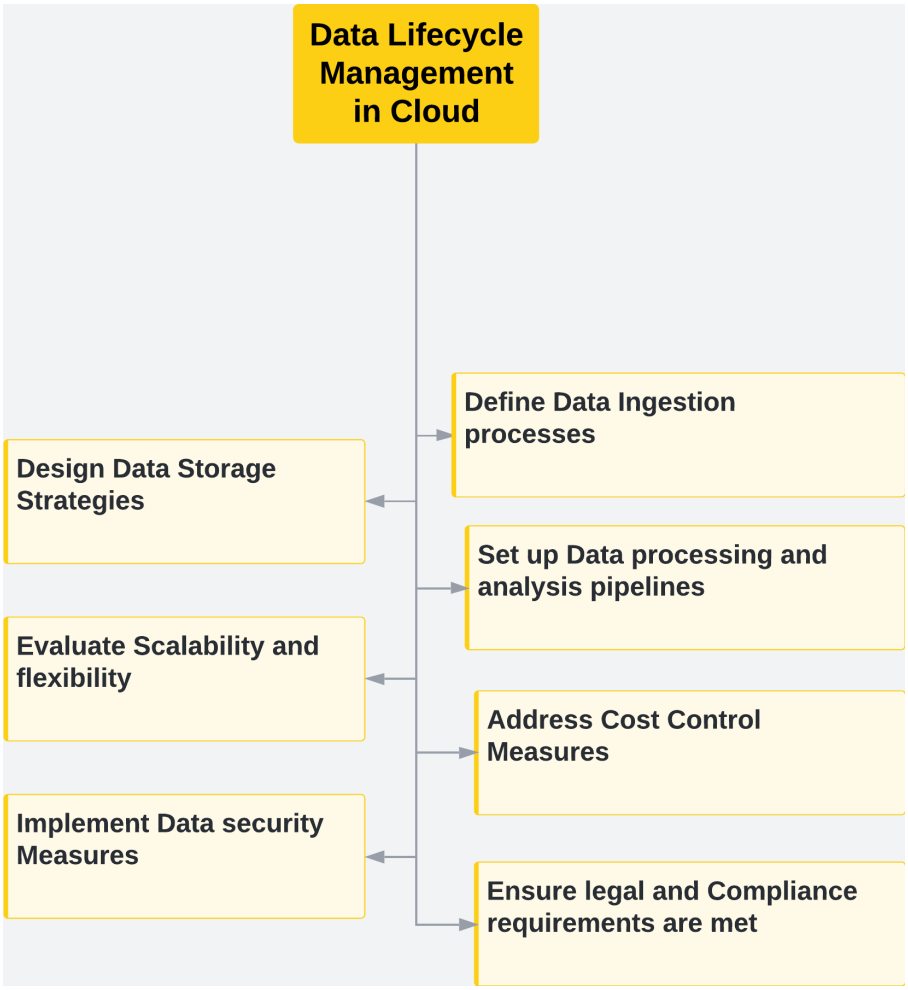


Figure A.11: Informal diagram of WBS-management

Figure A.12 focuses on setting up and testing cloud platforms and their capabilities.

- **Set up cloud platforms:** Deploy infrastructure on cloud services such as GCP, AWS, and Azure.
- **Data ingestion testing:** Verify the ability of each platform to process and handle data efficiently.
- **Evaluate performance:** Measure the speed, reliability, and resource usage of the platforms under different workloads.
- **Test DS/ML development lifecycle:** Assess the ease of developing, training, and deploying machine learning models.
- **Implement use case:** Apply a real-world scenario to validate the platform's capabilities.
- **Test data security and privacy features:** Examine the mechanisms in place to safeguard sensitive information.

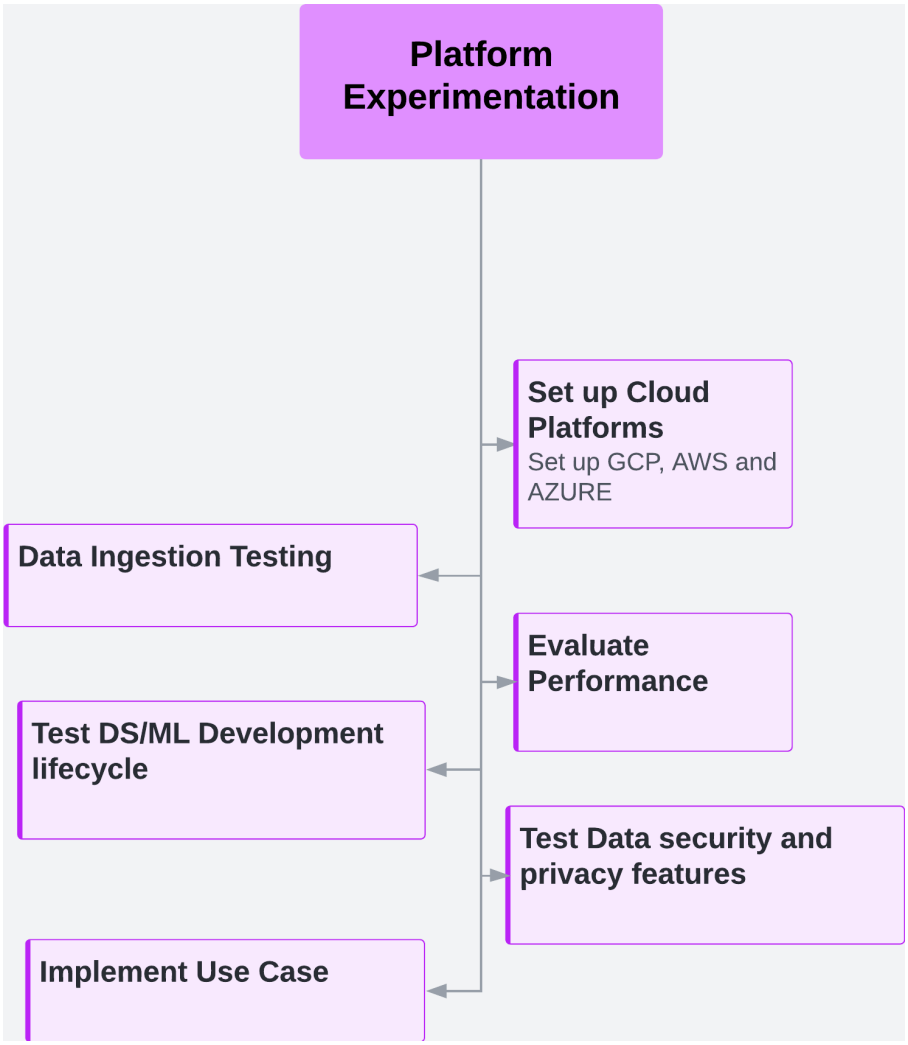


Figure A.12: Informal diagram of WBS-experimentation

Figure A.13 focuses on tasks related to the final stages of the project, ensuring outcomes are clearly documented and delivered effectively.

- **Data analysis from experiments:** Interpret the results gathered from experimentation to derive actionable insights.
- **Evaluate platform suitability for different use cases:** Compare and assess cloud platforms (like AWS, Azure, and GCP) against specific requirements and use cases.
- **Compare results:** Analyze and contrast the findings from different platforms or methods.
- **Prepare final report structure:** Organize the final deliverable in a logical and comprehensive manner.
- **Review and finalize thesis:** Revise the thesis document to ensure accuracy, clarity, and professionalism.

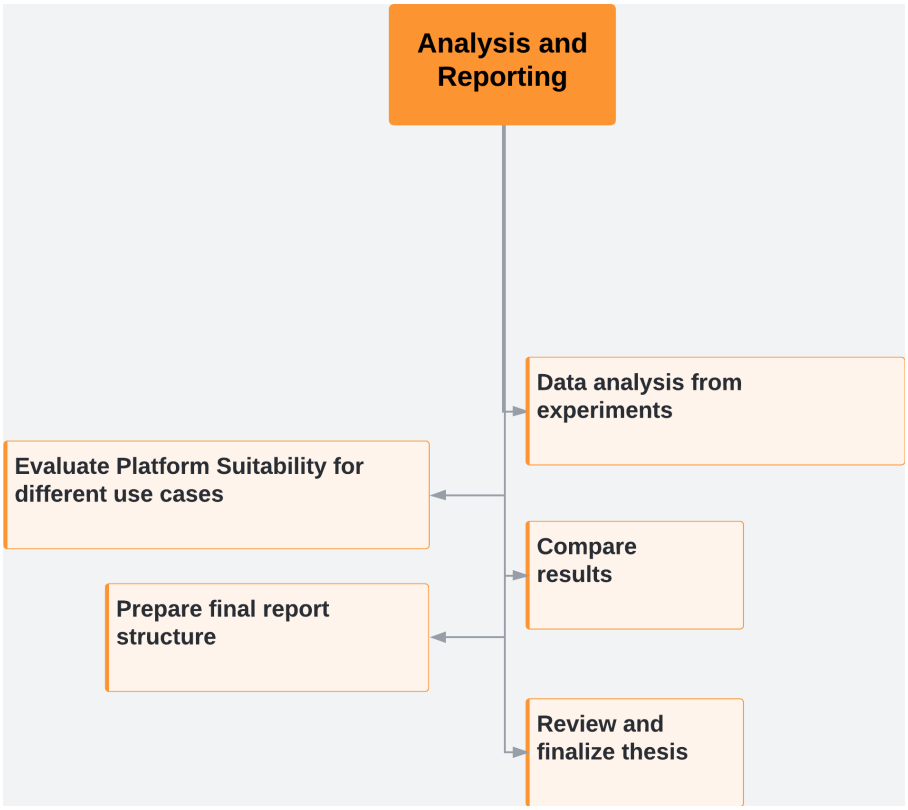


Figure A.13: Informal diagram of WBS-delivery

A.1.8 Project Schedule - Gantt

The project schedule is visualized using a Gantt chart, which outlines the timeline, key milestones, and dependencies for each phase of the work breakdown structure (WBS), ensuring a clear and structured approach to project execution. Figure A.14 demonstrates the project plannning. A complete version of the gantt diagram can be found in thid document's A, it comprises of 5 diferent images.

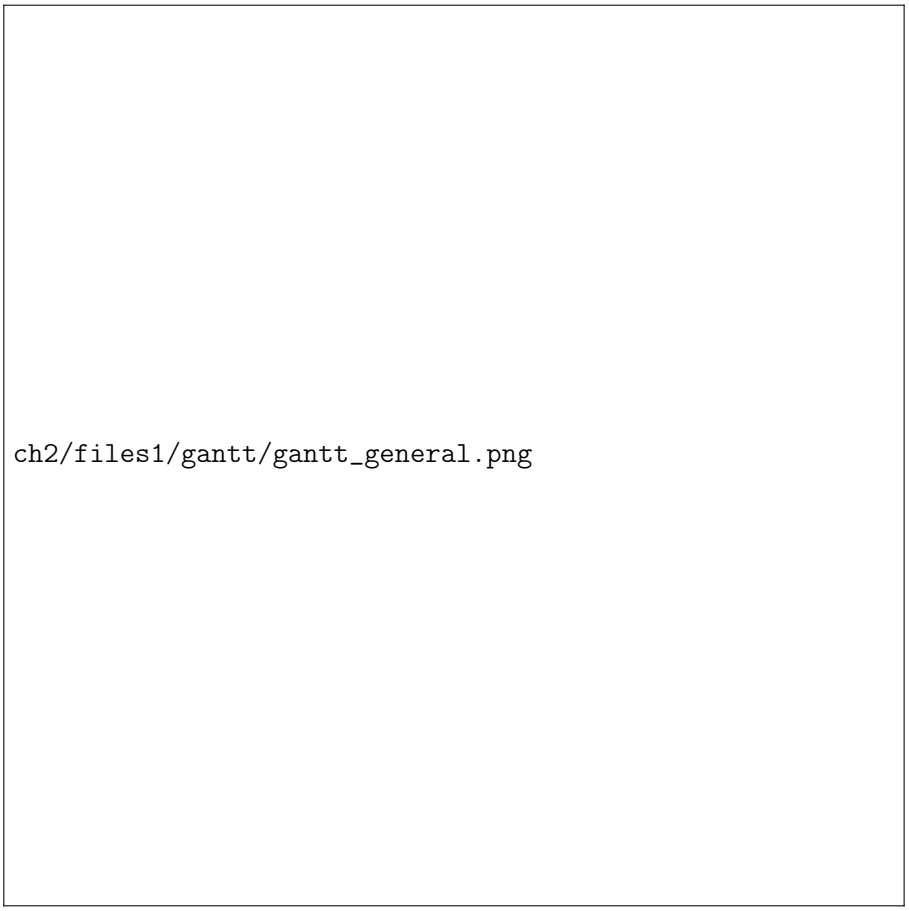


Figure A.14: General view of gantt diagram

A.1.9 Milestones

The project milestones represent significant achievements and checkpoints that guide the progress of the work. These include:

- Gaining foundational knowledge in cloud computing aspects related to data and AI.
- Completing the thesis document for submission to PREPD and DIMEI.
- Performing the High-Level Design (HLD) for the system architecture.
- Performing the Low-Level Design (LLD) for implementation details.
- Successfully implementing the intended use case within the system.

A.1.10 Deliverables

The project deliverables constitute tangible outputs that encapsulate the work and outcomes. These include:

- First version of the thesis document.
- Second and final version of the thesis document.
- Final project delivery, comprising the codebase, system documentation, and related artifacts.

A.1.11 Monitoring and Controlling Procedures

This subsection outlines the procedures that will be used to monitor and control the progress and quality of the project. These procedures ensure that the project remains on track and meets its objectives effectively.

- **Regular Meetings:** Scheduled meetings with both the academic supervisor and the company supervisor will be conducted to review progress, discuss challenges, and align expectations.
- **Use of Jira:** Jira will be employed as the primary tool for task tracking, progress monitoring, and issue management, ensuring visibility and accountability throughout the project lifecycle.
- **Work Reviews:** Regular reviews of work will be performed to assess quality, evaluate progress against milestones, and identify areas for improvement.

A.1.12 Risk Identification and Management

This section outlines the approach to identifying, assessing, and managing risks throughout the project. Effective risk management ensures that potential challenges are addressed proactively, minimizing their impact on the project's objectives and outcomes.

All identified risks will be documented and assessed using an Excel table. The table will contain the following columns to ensure comprehensive analysis and response planning:

- **Risk ID:** A unique identifier assigned to each risk for easy reference.
- **Description:** A brief description of the risk, summarizing the potential issue or challenge.
- **Cause:** The underlying reason or condition that could lead to the occurrence of the risk.
- **Effect:** The potential impact the risk may have on the project, should it materialize.
- **Risk Owner:** The individual or team responsible for monitoring and addressing the risk.
- **Probability (1-5):** The likelihood of the risk occurring, rated on a scale from 1 (very low) to 5 (very high).
- **Impact (1-5):** The severity of the risk's effect on the project, rated on a scale from 1 (minor) to 5 (critical).
- **PI Score:** The combined score derived from multiplying Probability and Impact, representing the overall risk level.
- **Expected Result, No Action:** The anticipated outcome if no mitigating actions are taken to address the risk.
- **Risk Response Type:** The strategy selected to address the risk, such as Avoid, Mitigate, Transfer, or Accept.
- **Response Description:** A detailed explanation of the actions to be taken to manage the risk effectively.

This structured approach enables the project team to assess risks quantitatively and define appropriate responses to mitigate their potential impact. A complete image of the risks can be found in this document's A.1

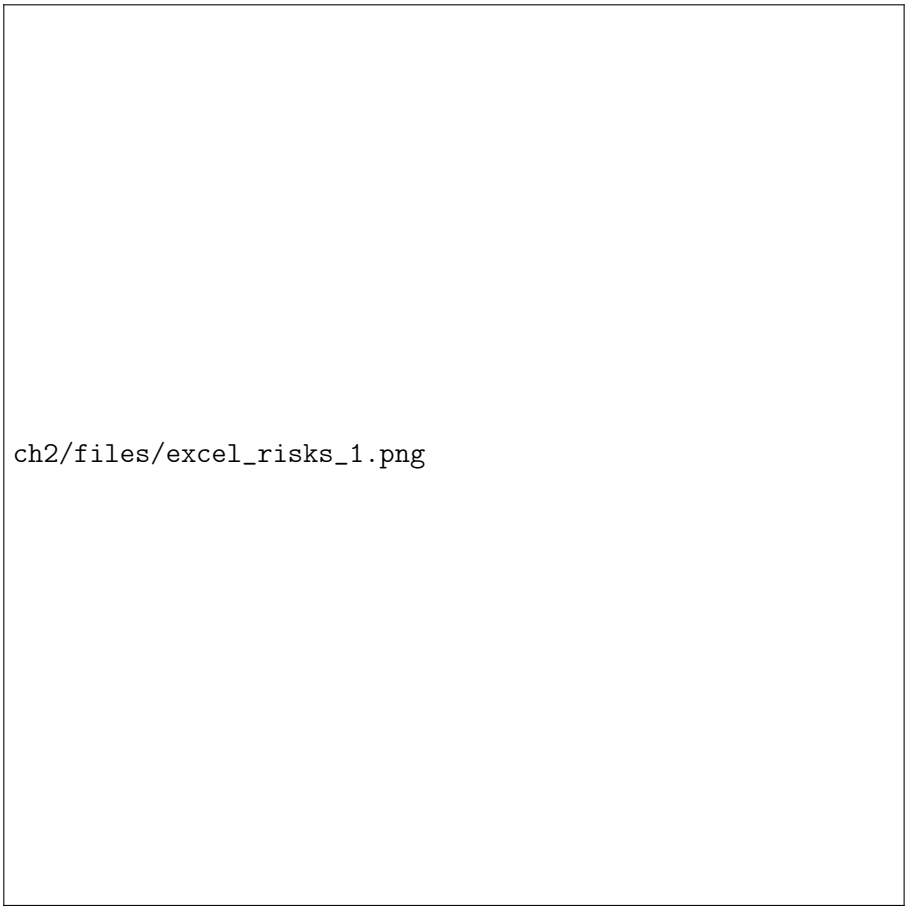


Figure A.15: Table of risks 1/2

Figure A.16: Table of risks 2/2