Text

Description automatically generated

Group Project Report  
Fundamentals of Cloud Computing  
MIS-612

Building Data Lake, Data Warehouse and Creating Interactive Dashboard on AWS

**Team-1**

Eduardo Dominguez

Imran Ahmad

Tanicha Ongalah

Tiffany Nguyen

Vaishnavi Mandadi

Venkata Avvari

Introduction:

We are working for Cloud Solutions Factory ICT as Business Analyst and Data Engineer. CSF-ICT specializes in data analysis and provides various reporting solutions to its customers along with data science capability. CSF-ICT is working with a new customer that is going to provide data set that needs analysis and insight. Our team is brought in to work on this new project where data will come in various formats from different methods. We were asked to build a data lake, data warehouse and provide dashboards to the customer.

Team members Role:

|  |  |
| --- | --- |
| Name | Role |
| Eduardo Domingue | Business Analyst |
| Imran Ahmad | Developer |
| Tanicha Ongalah | Developer |
| Tiffany Nguyen | Developer |
| Vaishnavi Mandadi | Business Analyst |
| Venkata Avvari | Business Analyst |

Presentation outline:

1. Purpose of each services in different options and what they do

2. Azure diagram and purpose of each service

3. List of different actions/tasks to accomplish the build

4. Challenges / lessons learned

5. Demo: a) S3 files, b) Lambda code, c) Glue, d) Quicksight, e) Redshift, f) Athena

## Services:

To Build Our Project we tried different AWS and third-party cloud provider services. The Following services we tried to build our project:

### AWS

|  |  |
| --- | --- |
| Name | Purpose |
| Athena | Analyze data in S3 using SQL queries |
| CloudTrail | Record of actions on AWS account for security/compliance |
| CloudWatch | Monitor and manage resources and applications in the cloud |
| Cognito | User sign-up, sign-in, and access control service |
| Cost Explorer | Tool to track AWS usage and estimate expenses |
| EMR | Hadoop and big data processing made easy in the cloud |
| Glue Crawler | Tool to automatically discover data for processing |
| Glue Job | Automated way to process data, no coding required |
| IAM | Controls who can access what in AWS |
| Kinesis | Real-time data streaming and processing |
| Lambda | Run code without provisioning servers or infrastructure |
| QuickSight | Data visualization tool for analyzing business data visually |
| Redshift | Cloud data warehousing for big data analytics |
| S3 | Cloud storage for files, accessible from anywhere |
| Sagemaker | Tools to build, train and deploy machine learning models |
| SNS | Instant messaging service for notifications and alerts |
| X-Ray | Debugging tool for identifying performance issues in applications |

### Third-party

|  |  |  |
| --- | --- | --- |
| Name | Purpose | Why is it better than \_\_\_? |
| Apache Kafka | Stream processing software for handling big data | Better than Kinesis because...   1. Open source 2. Broader ecosystem 3. Lower cost 4. Higher throughput 5. Multi-cloud support |
| Matillion | Data integration software for cloud-based analytics | Better than Glue, because...   1. User-friendly interface 2. Pre-built connectors 3. Scalability 4. Cost-effective pricing model 5. Fast time-to-value |
| Nifi | Data integration and distribution platform for big data | Better than Lambda because...   1. Visual data flow 2. Real-time processing 3. Scalability 4. Built-in data transformation 5. Data provenance |
| Power BI | Data visualization tool for business insights and reporting | Better than Quicksight because...   1. Better visualization options 2. More advanced data modeling 3. Greater flexibility in data sources 4. Easier report customization 5. More robust sharing options |
| Snowflake | Cloud-based data warehousing and analytics platform | Better than Redshift because...   1. Separation of storage/compute 2. Near-zero maintenance 3. Pay-per-second pricing 4. Better performance at scale 5. Native semi-structured data support |

## Architecture:

After exploring all these services, we built our project with suitable services to optimize cost and better performance and security. The architecture of our project is as following:

A picture containing text, diagram, plan, screenshot

Description automatically generated

Challenges/lessons learned:

* 1. There are many use cases for one service. Finding the use case that we want was hard but not impossible.
  2. When creating a query in Athena, the date column in the parquet file was not working. It might have been because of the data type, but we could not figure it out in time.
  3. We were unable to implemente the copy command in Redshift after several attempts.
  4. Understanding how to connect all of the services to each other was one challenge. Setting them up in everyone’s accounts was another challenge.
  5. When sharing codes to set up a Lambda or a Glue Job, we often ran into indentation errors due to formatting.