BIRLA INSTITITUE OF TECHNOLOGY & SCIENCE, PILANI (RAJASTHAN)

April 2024

SE ZG628T - DISSERTATION

Cloud-First Approach: Engineering a Solution for Efficient On-Premises Data Migration to Cloud Platforms

By

SHELKE AKSHAY NANDKUMAR

2022MT93331

Agenda

- Introduction
  - Background
  - Problem Definition
  - Project Objective
- Overview of Architectural Design
- Implementation & Working
  - Cloud Resources
  - Code Snippets
  - Github code repository
  - PowerBI Dashboards
  - Demo video of working solution
- Future Scope and Limitations

Introduction

- Background:

- Cloud computing's evolution has revolutionized IT infrastructure, offering scalable resources on-demand via the internet.
- Organizations migrate databases to the cloud for cost reduction, scalability, flexibility, security enhancement, and disaster recovery benefits.
- Challenges include data security, compliance, compatibility, and downtime mitigation during migration.
- Mitigation strategies involve thorough planning, data encryption, network optimization, and leveraging cloud migration tools.
- Best practices include infrastructure assessment, cloud model selection, provider evaluation, detailed migration planning, and rigorous testing for data integrity and application functionality.

Introduction

- Problem Definition :

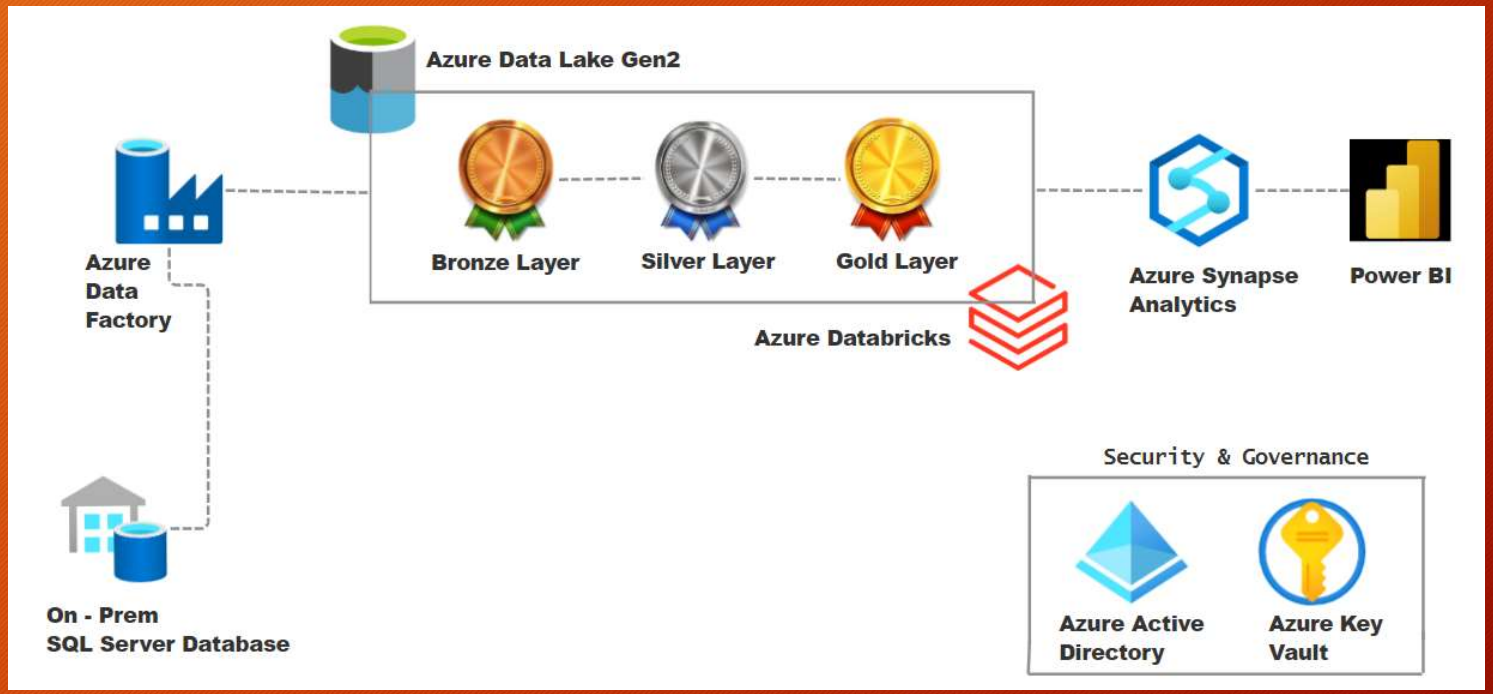
- The problem entails the need to efficiently and securely migrate on-premises databases to the cloud, balancing the benefits of scalability and flexibility with challenges such as data security, compliance, compatibility, and potential downtime, thereby ensuring seamless operations and preserving data integrity throughout the migration process.
- In this dissertation report I have created end-to-end Azure cloud based solution for data migration from on-premises to cloud infrastructure.

Introduction

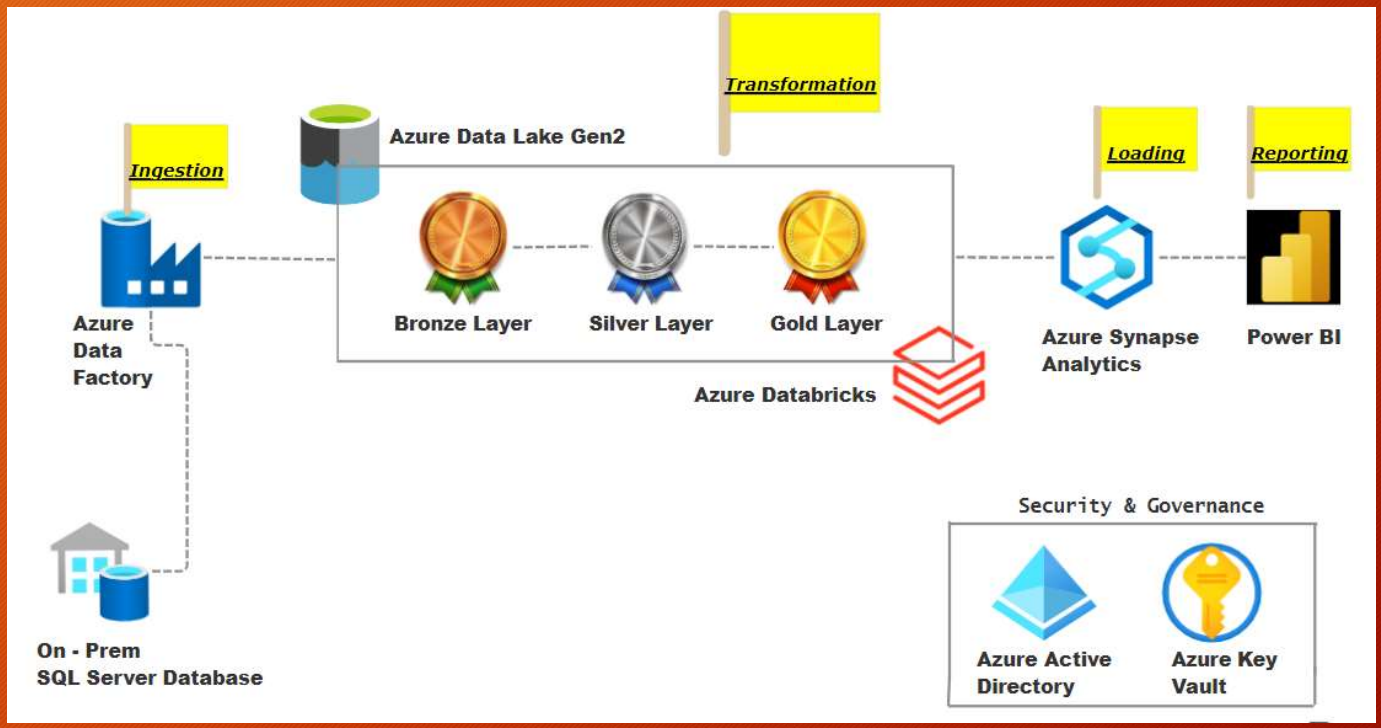
- Project Objective :

- Assess the current on-premises database infrastructure.
- Select the appropriate cloud service model and provider.
- Develop a comprehensive migration plan addressing security, compliance, and downtime.
- Execute the migration process while ensuring data integrity and minimal disruption.
- Validate the functionality of migrated databases in the cloud environment.

Overview of Architectural Design



Overview of Architectural Design



Implementation & Working

- Cloud Resources

- Azure Data Factory
- Azure Data Lake Storage (Gen2)
- Azure Databricks
- Azure Synapse Analytics
- Azure Key vault
- Azure Active Directory
- Microsoft PowerBI

Implementation & Working

- Code Snippets

```
from pyspark.sql.functions import from_utc_timestamp, date_format
from pyspark.sql.types import TimestampType

for i in table_name:
    path = "/mnt/bronze/SalesLT/" + i + "/" + i + ".parquet"
    df = spark.read.format("parquet").load(path)
    column = df.columns

    for col in column:
        if "Date" in col or "date" in col:
            df = df.withColumn(col, date_format(from_utc_timestamp(df[col].cast(TimestampType()), "UTC"), "yyyy-MM-dd"))

    output_path = "/mnt/silver/SalesLT/" + i + "/"
    df.write.format("delta").mode("overwrite").save(output_path)
```

Implementation & Working

- Code Snippets

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, regexp_replace

for name in table_name:
    path = "/mnt/silver/SalesLT/" + name
    print(path)
    df = spark.read.format("delta").load(path)

    # Get the list of column names
    column_names = df.columns

    for old_col_name in column_names:
        # Convert column name from ColumnName to Column_Name format
        new_col_name = "".join(["_" + char if char.isupper() and not old_col_name[i-1].isupper() else char for i, char in enumerate(old_col_name)]).
        lstrip("_")

        # Change the column name using withColumnRenamed and regexp_replace
        df = df.withColumnRenamed(old_col_name, new_col_name)

    output_path = "/mnt/gold/SalesLT/" + name + "/"
    df.write.format("delta").mode("overwrite").save(output_path)
```

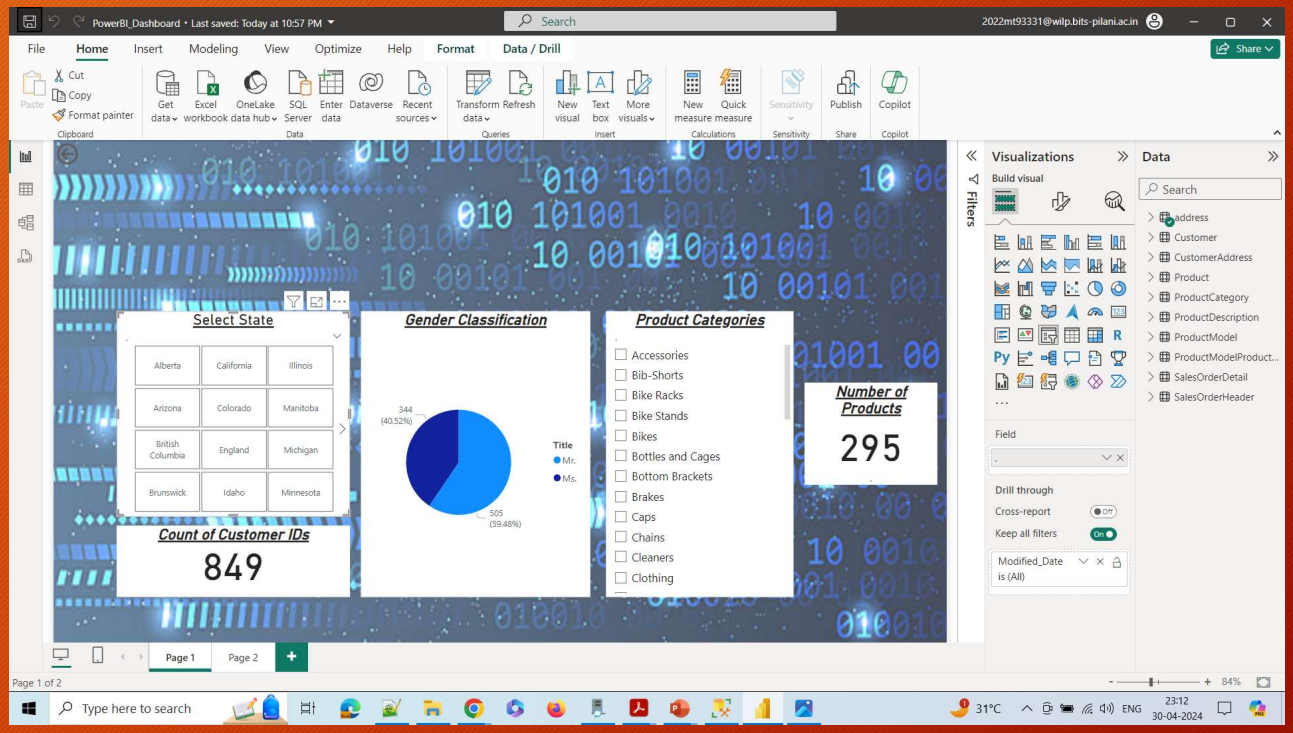
Implementation & Working

- Github code repository

- Link : https://github.com/ShelkeAkshay-2022mt93331/dissertation

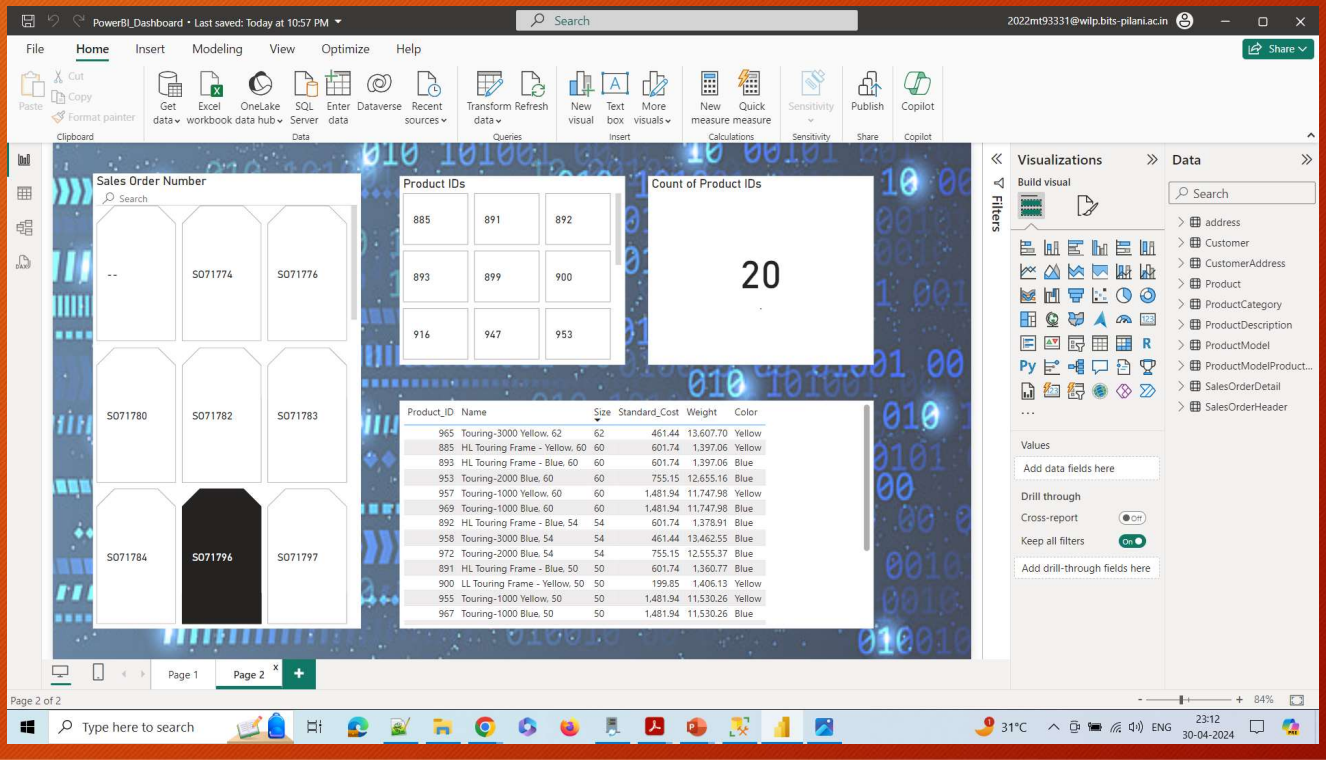
Implementation & Working

- PowerBI Dashboards



Implementation & Working

- PowerBI Dashboards



Implementation & Working

- Demo video of working solution

- Google Drive Link : https://drive.google.com/drive/folders/1qEwP5DDKlXF6vuxXq-6B8rs8iA8jmczB?usp=drive_link

- Google Drive Video Link : https://drive.google.com/file/d/1F5PeEJjorfrCm4q3jTkJCXk6tYbLfKPZ/view?usp=sharing

Future Scope & Limitations

- Future Scope :

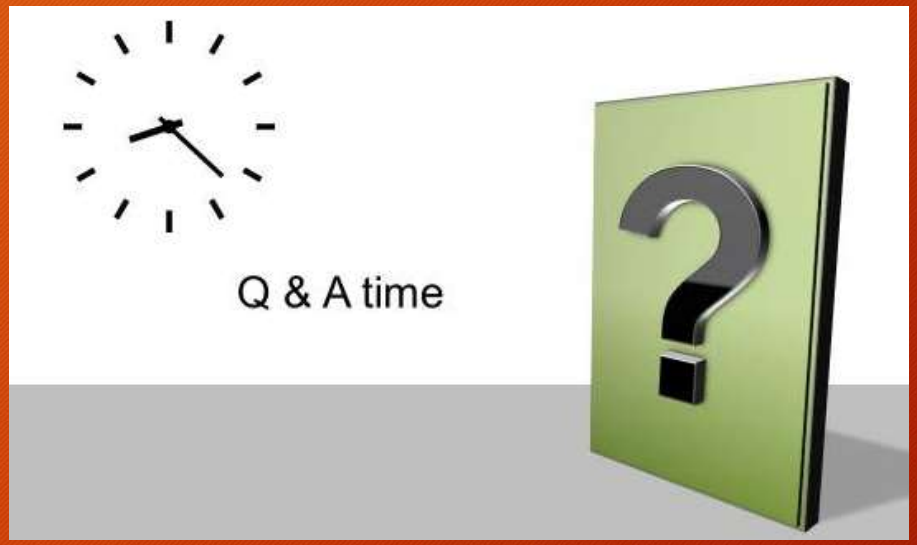
- Extending the solution to support real-time data streaming and processing for low-latency analytics.
- Incorporating advanced machine learning and AI capabilities for predictive analytics and automated decision-making.
- Enhancing the solution to handle multi-cloud environments and hybrid architectures.
- Implementing advanced data governance and lineage tracking mechanisms for improved compliance and auditing.
- Exploring serverless computing options for increased scalability and cost optimization.

Future Scope & Limitations

- Limitations :

- The solution is primarily focused on data migration and may require additional components for advanced use cases like real-time analytics or IoT data processing.
- While the solution demonstrates data migration from on-premises SQL Server, additional connectors and adaptations may be required for other data sources or formats.
- The project scope is limited to the Microsoft Azure ecosystem, and additional work may be required for integrating with other cloud platforms or on-premises systems.
- Advanced data governance and lineage tracking features may require further development and integration with external tools or services.
- The solution does not cover aspects of performance optimization, cost optimization, or autoscaling, which may be relevant for large-scale deployments.

Q & A



Thank You !