Data Warehousing with IBM Cloud Db2 Warehouse

Phase 3: Development Part 1

In this part we will begin building our project.
Start building the data warehouse using IBM Cloud Db2 Warehouse.
Define the schema and structure of the data warehouse tables.Identify data sources (e.g., CSV files, databases) and design a strategy to integrate them into the data warehouse.

Creating a data warehouse in IBM Db2 on IBM Cloud involves several steps, including provisioning the necessary resources, configuring the database, and loading data. Below is a simplified step-by-step guide to get you started.

Processes

1. **Sign Up and Log In to IBM Cloud:**
   If you haven't already, sign up for an IBM Cloud account and log in.

2. Provision Db2 Service:

- From the IBM Cloud dashboard, navigate to the catalog.
- Search for "Db2" and select the Db2 service that suits your needs. You may choose Db2 on Cloud or Db2 Warehouse on Cloud based on your specific requirements.

3.Create an Instance:

- Click "Create" to set up a new instance of the Db2 service.
- Follow the on-screen instructions to configure the service instance, such as selecting the plan, naming the instance, specifying the region, and setting resource group options.

4.Access the Db2 Console:

- ○ Once your Db2 instance is provisioned, go to the Db2 service in your IBM Cloud dashboard.
- ○ Click "Open Console" to access the Db2 web console.

5.Create a Database:

- ○ In the Db2 web console, click "Create Database" to define a new database for your data warehouse.

- Specify the database name, the storage type, and any other configuration settings.

6.Define Tables and Schema:

- Define the schema of your data warehouse by creating tables and specifying their structures. You can do this using SQL commands or the web-based interface.

7.Load Data:

- To populate your data warehouse, you can load data from various sources. Db2 provides tools to load data from CSV files or other databases.
- You can use SQL statements or the Db2 data import tools to load data into your tables.

8.Query and Analyze Data:

- With your data loaded, you can start running SQL queries to analyze your data. Use the Db2 console's SQL editor or connect to Db2 from a client tool to run queries.

9.Performance Optimization:

- As your data warehouse grows, you may need to optimize its performance. This can involve creating indexes, optimizing queries, and managing system resources.

10.Security and Access Control:

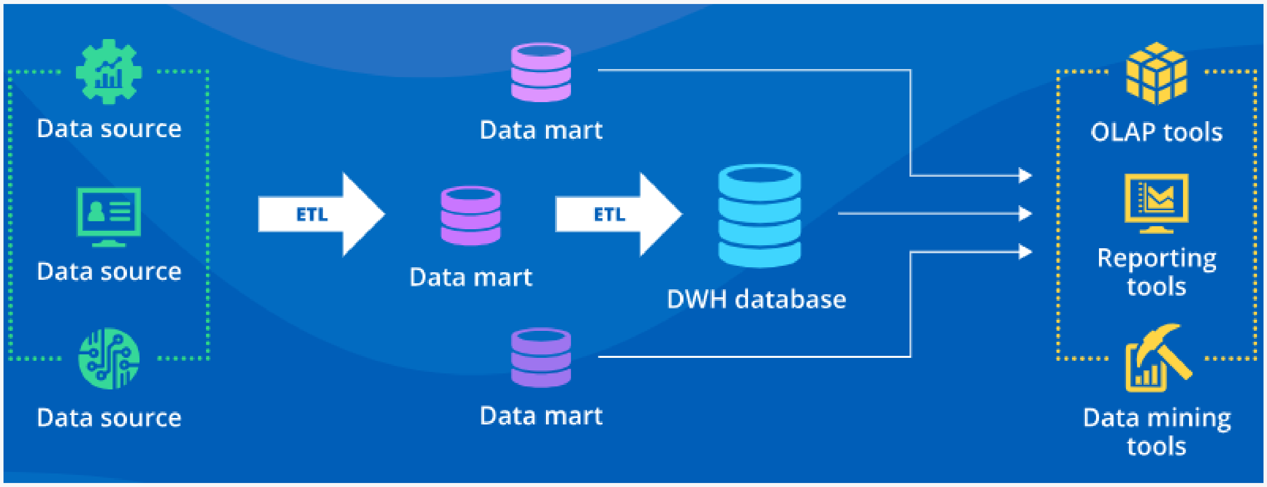
- Ensure that you have proper security measures in place, such as access control, authentication, and encryption, to protect your data.

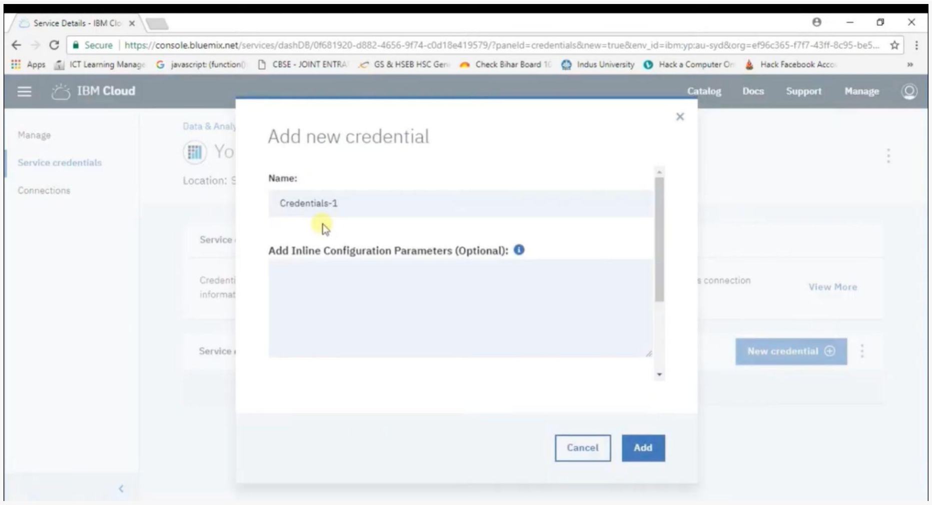
11.Regular Maintenance:

- Regularly monitor and maintain your data warehouse to ensure it continues to meet your performance and data quality requirements. This includes data backups, updates, and system tuning.

These steps should guide you through the process of creating a data warehouse in IBM Db2 on IBM Cloud.

Kimball's approach – creating data marts first and then developing a data warehouse database incrementally from independent data marts.



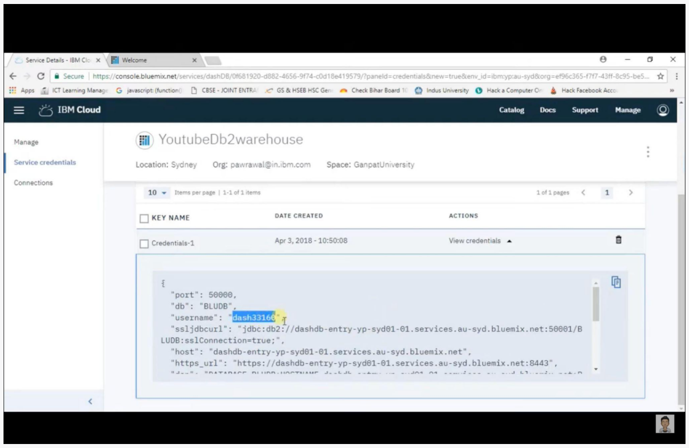


Add new credential

Name:

Credentials-1

Add Inline Configuration Parameters (Optional):

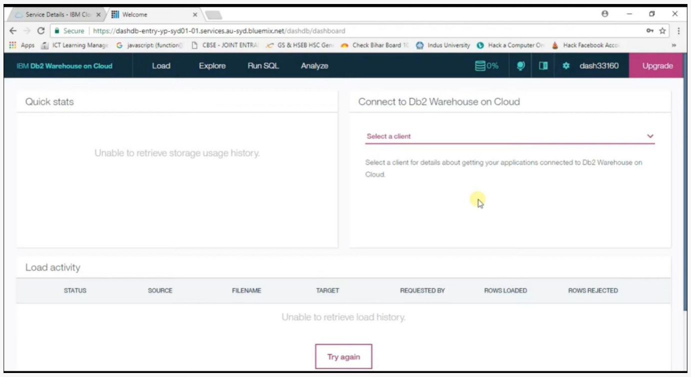


YoutubeDb2warehouse

Location: Sydney Org: pawrawal@in.ibm.com Space: GanpatUniversity

KEY NAME	DATE CREATED	ACTIONS
Credentials-1	Apr 3, 2018 - 10:50:08	View credentials

```
{
  "port": 50000,
  "db": "BLUDB",
  "username": "dash33160",
  "ssljdbcurl": "jdbc:db2://dashdb-entry-yp-syd01-01.services.au-syd.bluemix.net:50001/B
LUDB:sslConnection=true;",
  "host": "dashdb-entry-yp-syd01-01.services.au-syd.bluemix.net",
  "https_url": "https://dashdb-entry-yp-syd01-01.services.au-syd.bluemix.net:8443",
```

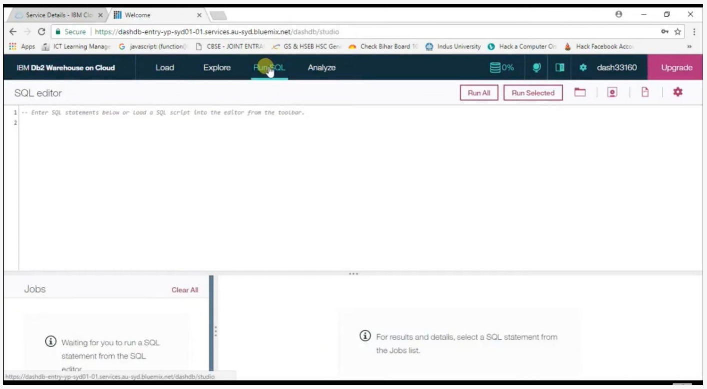


Unable to retrieve storage usage history.

Select a client for details about getting your applications connected to Db2 Warehouse on Cloud.

STATUS	SOURCE	FILENAME	TARGET	REQUESTED BY	ROWS LOADED	ROWS REJECTED

Unable to retrieve load history.



```
-- Enter SQL statements below or load a SQL script into the editor from the toolbar.
```

Waiting for you to run a SQL statement from the SQL editor

For results and details, select a SQL statement from the Jobs list.

Defining the schema and structure of your data warehouse tables and identifying data sources are crucial steps in building an effective data warehouse. Here's a detailed process to help you define the schema, structure, and integration strategy for your data warehouse:

1.Understand Business Requirements:

- Before defining the schema, understand the business requirements and objectives of your data warehouse. What kind of data do you need to store, analyze, and report on? What are the key performance indicators (KPIs) and metrics?

2. Define the Schema:

- Determine the tables and entities you need in your data warehouse. This involves identifying the different dimensions and facts that are relevant to your business. Commonly used schemas include star schema and snowflake schema.
- Design dimension tables to store descriptive attributes, and fact tables to store measures and metrics.
- Establish relationships between tables, such as foreign keys in dimension tables linking to primary keys in fact tables.

3. Identify Data Sources:

- Identify the sources of data that you need to integrate into your data warehouse. This can include:
    - Relational databases (e.g., Oracle, SQL Server, MySQL)
    - Flat files (e.g., CSV, Excel)
    - External APIs
    - Streaming data sources
    - Other data warehouses

4. Data Extraction:

- Determine how you will extract data from these sources. This can involve techniques like batch ETL (Extract, Transform, Load) processes, real-time data streaming, or API calls.
- Choose tools or scripts that can help automate data extraction from various sources.

5. Data Transformation:

- Data from different sources may have varying formats and structures. Transform the data as needed to ensure it aligns with your defined schema. This may include data cleaning, restructuring, and aggregations.
- Use ETL tools or scripts to perform these transformations.

6. Data Loading:

- Load the transformed data into your data warehouse. Most data warehouses provide tools and utilities to facilitate this process. For example, you might use SQL commands, data import/export utilities, or ETL tools.

7. Data Quality and Validation:

- Implement data quality checks and validation procedures to ensure the accuracy and consistency of the data being loaded.
- Monitor for data anomalies and inconsistencies.

8. Metadata Management:

- Maintain metadata about the source of each data element, transformations applied, and any relevant business rules. This helps with data lineage and troubleshooting.

9. Performance Considerations:

- Optimize data loading and query performance by creating appropriate indexes, partitions, and caching strategies.
- Monitor and tune your data warehouse to ensure it performs efficiently.

10. Data Security and Access Control:

- Implement security measures to protect your data. Define access controls and encryption methods to secure sensitive information.

11. Documentation:

- Document the schema, data integration processes, and transformation rules for future reference and for the benefit of other team members.

12. Regular Maintenance and Updates:

- Data warehouses are not static; they evolve with your business needs. Plan for regular data updates, maintenance, and schema changes.

13. Testing and Validation:

- Before deploying the data warehouse for production use, thoroughly test it to ensure it meets the business requirements and functions correctly.

14. Backup and Recovery:

- Implement backup and recovery procedures to safeguard your data warehouse against potential data loss or system failures.

Conclusion:

- In the rapidly evolving landscape of data-driven decision-making, the development of a data warehouse is a pivotal step towards empowering organizations with actionable insights and analytics. As we've explored in this document, the creation of a data warehouse involves a meticulous process that begins with understanding the business objectives and culminates in the integration of diverse data sources into a structured and accessible repository. Here are some key takeaways:
- Business-Driven Approach: The journey of building a data warehouse commences with a profound understanding of the organization's business requirements. By aligning the schema and structure with the specific needs of the business, the data warehouse becomes a valuable asset for decision-makers.
- Integration is Key: In a world awash with data, the capability to extract, transform, and load data from various sources is central to a data warehouse's effectiveness. The integration strategy must be tailored to accommodate diverse data sources, ensuring that data quality and consistency are maintained.
- Data Transformation and Quality: Data transformation is not just about making data fit the schema but also involves enhancing data quality. Addressing issues of data accuracy, completeness, and consistency is imperative to make the data warehouse a reliable resource.
- Performance and Security: To make the data warehouse perform optimally, various aspects like indexing, partitioning, and caching need careful consideration. Additionally, robust security measures must be in place to safeguard sensitive data and maintain data integrity.
- Documentation and Collaboration: Clear and comprehensive documentation of schema, processes, and transformations is essential for the continued success of the data warehouse. Collaboration with all relevant stakeholders is crucial, as it ensures that the data warehouse remains aligned with evolving business needs.

- Continuous Improvement: A data warehouse is not a static entity; it's a living, evolving resource. Regular maintenance, updates, testing, and performance tuning are essential to keep the data warehouse effective and up-to-date.
- Testing and Validation: Prior to deploying the data warehouse for production use, rigorous testing and validation are required to ensure that it meets the defined business requirements and operates correctly.
- Backup and Recovery: Implementing robust backup and recovery procedures is vital to safeguard data against potential losses or system failures.