STUDY & COMPARISON BETWEEN

amazon REDSHIFT

VS

APACHE Spark™

with

amazon web services™ | S3

using

databricks

Presented by

Name	**Student Id no**	**Email id**
Aishwarya Gupta	18210298	Aishwarya.gupta3@mail.dcu.ie
Apurva Gawad	18210295	Apurva.gawad2@mail.dcu.ie
Gautam Shanbhag	18210455	Gautam.shanbhag2@mail.dcu.ie

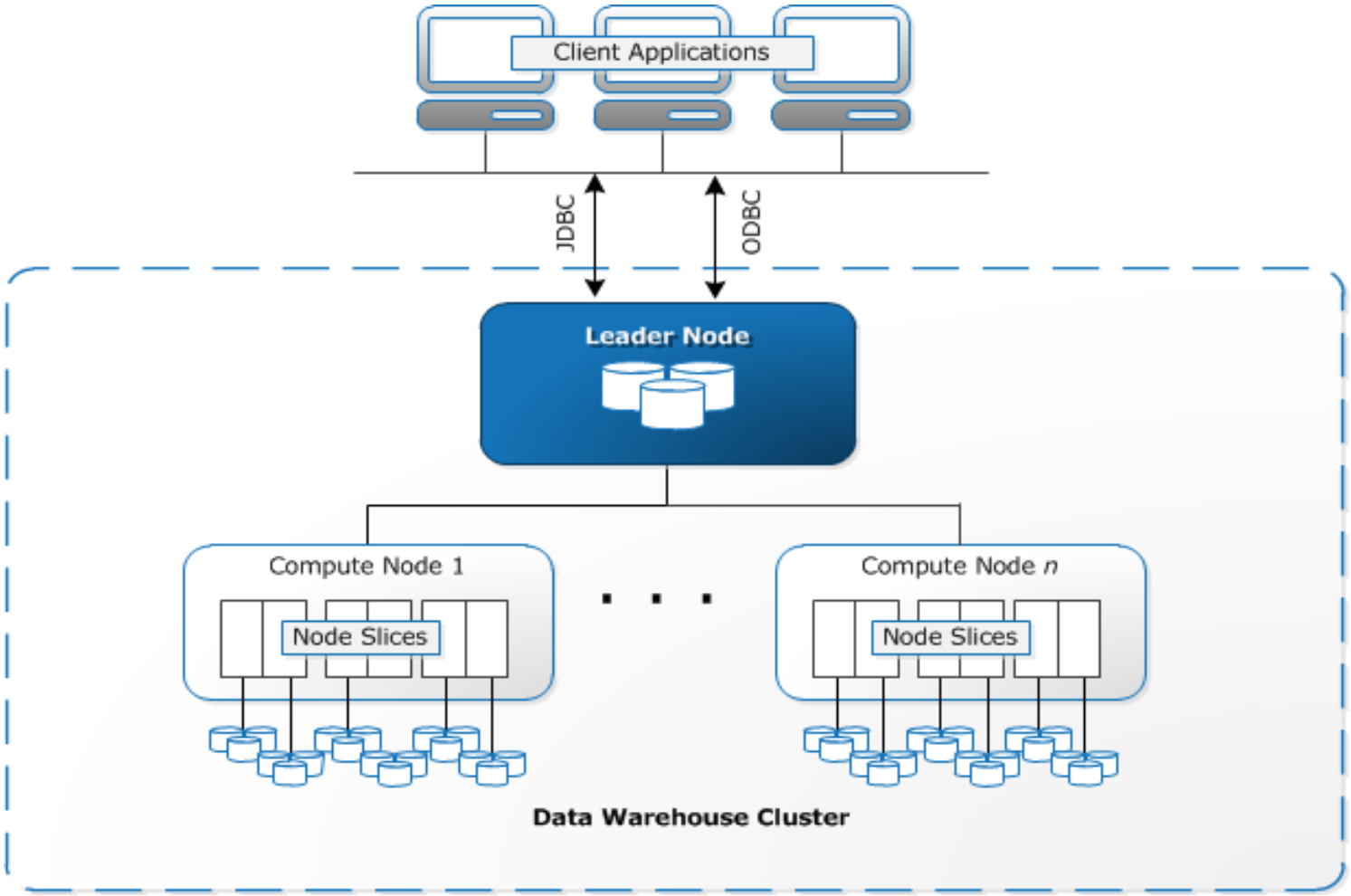
amazon
REDSHIFT

Overview

- Data warehouse cloud service from AWS
- Storage and analysis on large amount of data
- High performance and low costs
  - Massive Parallel Processing
  - Columnar data storage
- Security
  - End-to-end encryption
  - Network isolation
  - Audit and compliance
  - Access and Management
- Scalability
- Backup and Recovery

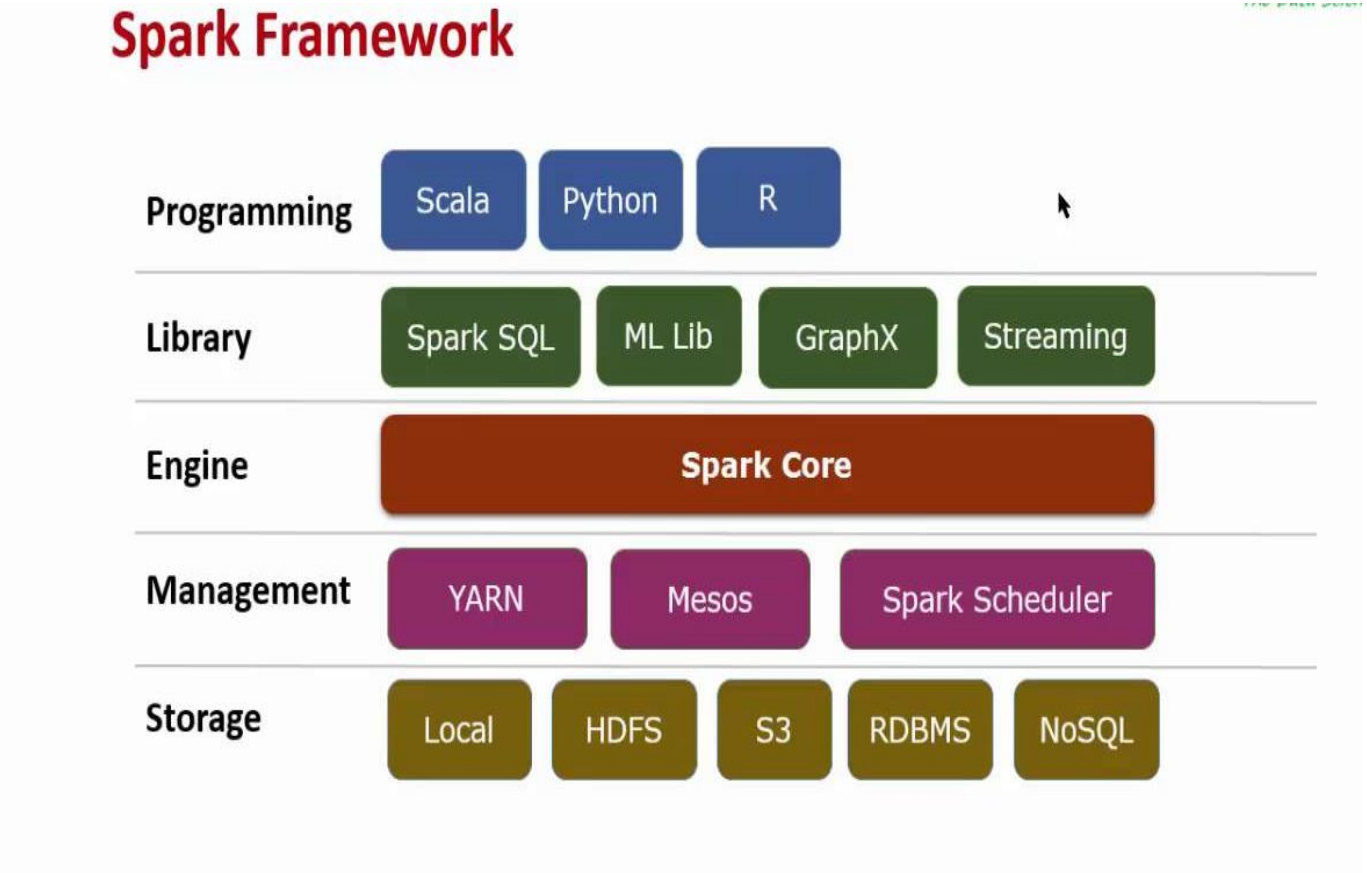
amazon REDSHIFT

Architecture

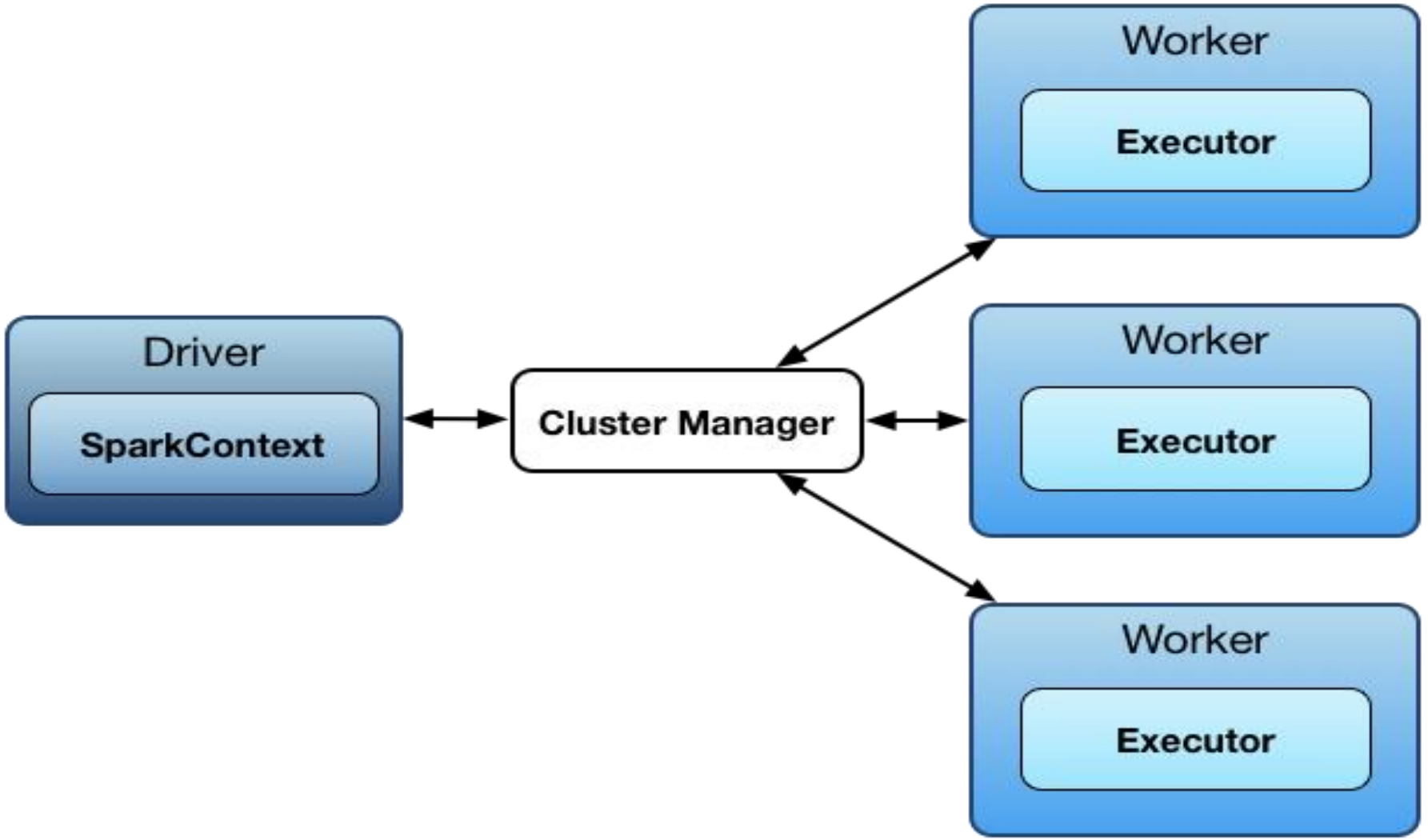


Overview

- Open source framework maintained by Apache
- Cluster computing
- Large scale data processing
- Master slave architecture



Architecture

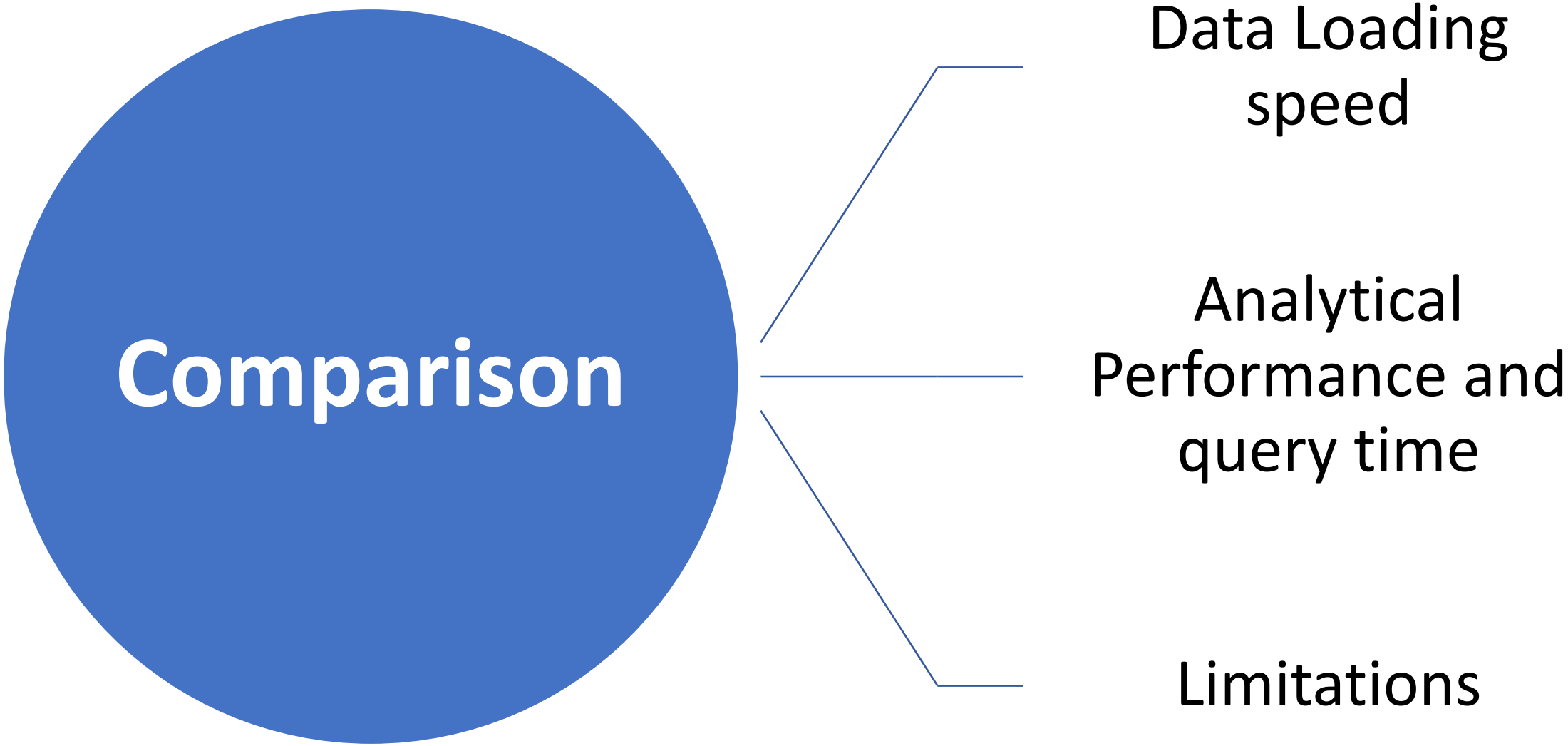


Configuration

- AWS account setup and IAM role configuration.
- Redshift Cluster creation and Initialisation.
- S3 bucket creation.
- Databricks Cluster creation and initialisation with Spark Framework

Implementation Plan

- Black Friday retail store dataset
- Languages used
- Loading data from S3
- Data Processing
- Upload data result to S3


Comparison
Data Loading speed
Analytical Performance and query time
Limitations

Experimental Comparison

Comparison factor	Amazon Redshift	Apache Spark
Data Load time from S3	6s 663ms (50s 238 ms *)	2s 620 ms
Fetching entire data set	12 s 557ms (21 s 257ms *)	20 ms
Aggregate function execution	393 ms (8s 193 ms *)	80 ms
Nested query execution	406 ms (30s 629 ms *)	100 ms
Deletion of values based on condition	357 ms (2s 183 ms *)	110 ms
Drop table	153 ms (162 ms *)	10 ms
Alter command	185 ms (3s 607ms *)	20 ms
Update command	1s 290 ms (2s 691ms *)	30 ms
Data unload to S3	1s 578ms (7s 298 ms *)	36s 760 ms

* RedShift performs Result Caching, making it faster on repeated operations

Conceptual Comparison

Comparison factor	Amazon Redshift	Apache Spark
Category	Analytical database	Data processing engine
Supported languages	SQL-Like language	Extensible via pre built libraries in Java, Python, R, Scala
Interoperability	Compatible with standard JDBC, ODBC drivers Integrated with most BI, ETL tools	Runs Everywhere. Can access diverse data sources. Runs on Hadoop, Mesos, Kubernetes, standalone or in the cloud.
Distribution model	Commercial/managed	Opensource
Storage Support	Inbuilt Storage	External Storage Engine Required
Live app Database	Not supported	Supports streaming data
Concurrency	Efficient multi user support	Performance degrades as concurrency increases
Typical application	Data warehouse	applications

Conclusion

When to use Redshift ?

- Storage and analysis of large amount of data
- Running analytical queries against ever growing data
- Benefit of AWS compatibility for Security, hardware management

When to use Spark ?

- Faster processing of huge volume of data
- To process streaming data
- To build machine learning applications

References

- Online reference - https://docs.aws.amazon.com/redshift
- Online reference - https://dbengines.com/en/system/Amazon+Redshift%3BSpark+SQL
- Online reference - https://spark.apache.org/
- Image reference - https://www.youtube.com/watch?v=ZTFGwQaXJm8
- Online reference - https://aws.amazon.com/s3/
- Online reference - https://spark.apache.org/docs/latest/sql-programming-guide.html
- Online reference - https://aws.amazon.com/what-is-cloud-computing/