1. **Write the definitions for the data terms slides and provide an example use-case for each term.**
2. Data Catalog: A data catalog is an organized metadata repository that enables users to quickly find, manage, and analyze data assets across several data sources inside their company. It is frequently built using AWS Glue, AWS Lake Formation, and Amazon Athena.

Test Case: Check to see if the data catalog fully covers the range of accessible data assets by automatically finding and indexing newly additional datasets to Amazon S3 buckets.

1. Data Anonymization/Data Masking: Encrypting sensitive information in datasets to safeguard individuals' privacy is known as data anonymization or data masking. This is usually accomplished by substituting fake or generic values for identifiable information, all the while maintaining the dataset's analytical or testing utility.

Test Case: Check that sensitive fields containing personally identifiable information (PII) are adequately masked or anonymized in sample datasets, ensuring data utility while respecting privacy.

1. PII Data: Any information that may be used to identify an individual, such as names, social security numbers, email addresses, or biometric data, is referred to as Personally Identifiable Information (PII) data and needs to be treated carefully to comply with privacy requirements.

Test Case: Verify that the encryption and decryption methods used to safeguard personally identifiable information (PII) in AWS storage services like Amazon S3 and Amazon RDS are reliable.

1. Data Democratization: In an organization, data democratization is the process of giving a wider variety of people access to data and analytics capabilities, enabling self-service analytics and enabling users to make data-driven decisions without requiring a high level of technical skill.

Test Case: Use user acceptability testing with business stakeholders to assess the usability and accessibility of self-service analytics platforms like Tableau Server or Amazon QuickSight.

1. Data Modelling: Creating an entity-relationship diagram or schema design to efficiently organize and represent data for storage and retrieval is known as data modeling. It entails creating the structure and relationships of data entities within a database or data warehouse.

Test Case: Compare entity-relationship diagrams (ERDs) and data models to business requirements and data source documentation to ensure accuracy and completeness.

1. Dashboards: Dashboards are graphic user interfaces that provide metrics, key performance indicators (KPIs), and other pertinent data in an easy-to-read and interactive manner. This enables users to analyze and track data trends, make well-informed decisions, and monitor advancement toward organizational objectives.

Test Case: Analyze performance data and simulate concurrent user access to validate dashboards created with Tableau Server or Amazon Quick Sight for responsiveness and load times.

1. Data Ecosystem: A data ecosystem includes data storage, processing, integration, analytics, and governance components, as well as an integrated collection of tools, services, and procedures for managing, evaluating, and extracting value from data throughout an enterprise.

Test Case: Validate end-to-end data flows across the ecosystem by tracking data migration from source systems to destination storage and analytics platforms, assuring data integrity and dependability.

1. Data Enrichment: To make data more relevant and high-quality for analysis and decision-making, data enrichment is adding new contextual information or features to preexisting datasets from outside sources, such as market trends, demographic data, or geographic data.

Test Case: Evaluate the completeness and precision of data enrichment procedures by contrasting enriched datasets with established reference data sources or ground truth values.

1. Data Exchange: The process of securely sharing and transferring data between various systems, applications, or organizations is known as data exchange. To promote smooth data interoperability and collaboration, APIs, data pipelines, or integration platforms are frequently used as facilitators.

Test Case: By transferring huge datasets between various AWS accounts or regions, you may test the reliability and performance of data sharing protocols like AWS Data sharing and AWS Direct Connect.

1. Data ExtractionThe process of extracting data from different sources—like databases, files, or APIs—and moving it to another location for additional processing, analysis, or archiving is known as data extraction. This process is frequently carried out with the use of ETL (Extract, Transform, Load) tools or services.

Test Case: Validate the correctness and completeness of data extraction operations by comparing derived datasets to source data records to ensure that all relevant data is obtained.

1. Data Governance: Data governance includes access control, data management, and regulatory compliance procedures as well as the policies, procedures, and controls that guarantee the availability, integrity, security, and compliance of data assets within an organization.

Test Case: Ensure that data governance standards are properly applied across AWS services such as Amazon S3, Amazon RDS, and Amazon Redshift, assuring regulatory compliance with organizational data policies.

1. Data Ingestion: The process of gathering, receiving, and loading data from different sources into a data processing or storage system like an analytics platform, data warehouse, or data lake is known as data ingestion. This is usually done with the use of automated pipelines or ingestion tools.

Test Case: By simulating different data volumes and ingestion rates, you can validate the scalability and performance of data ingestion pipelines and make sure they can handle peak loads without any data loss or performance deterioration.

1. Data Joins: Data joins, which are frequently carried out in SQL queries or data processing processes, are the process of merging linked datasets based on shared keys or attributes to produce a unified picture of the data, enabling analysis and insights production across numerous sources.

Test Case: Compare query results to expected output based on established join conditions to verify the accuracy of join operations conducted on datasets stored in Amazon Redshift or Amazon Athena.

1. Data Lineage: Data lineage provides insight into the creation, usage, and modification of data across various systems, processes, and analytical workflows by tracking the beginnings, transformations, and migrations of data throughout its existence.

Test Case: Make sure that data lineage accurately and completely traces the path of data from its source to its destination by verifying the information obtained from AWS services like AWS Glue and AWS Lake Formation.

1. Data MeshBy treating data as a product, empowering cross-functional teams to oversee their own data domains and offering self-serve data infrastructure and tools to promote data democratization and agility, data mesh is an architectural approach that promotes decentralizing data ownership and administration.

Test Case: Individual data domains' autonomy and scalability inside the data mesh architecture should be validated by evaluating their ability to manage and evolve their data assets independently.

Data terms:

1. Data Portability: The term "data portability" describes the ease with which data can be moved between various AWS services, regions, or storage systems, guaranteeing flexibility and agility in the management of data assets.

Test case: Export a dataset to Google Cloud Storage from Amazon S3, then confirm that the import went through without any corruption or loss of data.

2. Data Replication: Data replication is the process of duplicating data from one storage place or service to another, generally for disaster recovery, high availability, or data distribution across many locations or settings.

Test case: Configure data replication between various Amazon RDS instances in different regions, then make sure the duplicated data is consistent.

3. Data PrivacyProtecting sensitive data and making sure that data protection laws are followed require the implementation of suitable security measures, encryption methods, access controls, and data governance procedures to prevent unauthorized access to or disclosure of data.

Test case: Use AWS Key Management Service (KMS) to encrypt data stored in Amazon S3 server-side and confirm that the right key is used to decrypt the data.

4. Data Consistency: The term "data consistency" describes the accuracy and dependability of data across several systems or replicas, making sure that all copies of the data are current and synced to preserve data integrity and dependability for users and applications.

Test case: Compare data snapshots to ensure consistency of replicated data across Amazon RDS multi-AZ installations.

5. Data Quality: Data validation, cleansing, enrichment, and monitoring procedures are frequently used to assure high-quality data for analytics and decision-making. Data quality is defined as the degree of accuracy, completeness, consistency, and dependability of data stored and processed inside the AWS ecosystem.

Test case: Make sure that only high-quality data is put into Amazon Redshift by validating data validation rules and constraints during data ingestion.

6. Data Silo: When data is kept or handled in separate or dispersed systems or settings, it creates silos, which hinder data exchange and cooperation inside an organization and cause inefficiencies and duplication.

Test case: Analyze data flow diagrams and do data inventory evaluations to find any existing data silos within the company.

7. Data Validation: Data validation is the process of examining and verifying the accuracy, integrity, and compliance of data with predetermined rules, standards, or requirements. This is often done using automated validation routines or manual review processes to assure data quality and reliability.

Test case: Verify the correctness, consistency, and completeness of the data against pre-established validation standards by implementing automated data validation tests during data import operations.

8. Data Wrangling: The process of cleaning, parsing, aggregating, and structuring data using tools and services like AWS Glue, Data Pipeline, or Amazon EMR is known as "data wrangling." It involves converting raw or unstructured data into a format that can be used for analysis or consumption.

Test case: 1: Conduct user acceptance testing with data analysts and business users to assess the usability and efficacy of data wrangling tools like AWS Glue DataBrew or Amazon SageMaker Data Wrangler.

9. Database Schema: A database schema outlines the tables, fields, constraints, and indexes that control how data is stored, accessed, and altered. It also describes the relationships between the many data items that make up a database.

Test case: Verify the accuracy of database schema design by contrasting it with industry standards and best practices for data modeling.

10. Data Stewardship: Assigning accountability and duty for managing and safeguarding an organization's data assets, such as creating data governance guidelines, monitoring data use and access, and making sure that data privacy laws and industry standards are followed, is known as data stewardship.

Test case: Establish and implement roles and duties for data stewardship inside the company to guarantee that data security, quality, and compliance are clearly accountable for.

11. EDI Data Standards: The formats, protocols, and syntax for electronically exchanging structured data between various business systems or trading partners are defined by Electronic Data Interchange (EDI) data standards. This allows for the seamless integration and interoperability of data across supply chains and corporate processes.

Test case: By parsing and comparing EDI messages to the appropriate standards, you can verify compliance with EDI data standards like ANSI X12 or EDIFACT.

12. ObservabilityUsing metrics, logs, traces, and other telemetry data to identify issues, troubleshoot issues, and optimize resource utilization and user experience, observability refers to the capacity to watch over, measure, and comprehend the behavior and performance of distributed systems, applications, and services.

Test case: By creating a simulated workload and examining logs and metrics for performance, faults, and resource usage, you may verify the efficacy of monitoring and logging setups.

13. Streaming Data: AWS Lambda and Amazon Kinesis are two examples of streaming data services that process, analyze, and act upon continuous and real-time data streams produced by devices, sensors, applications, or online transactions in near real-time to support use cases like real-time analytics, monitoring, and alerting.

Test case: Ingest and process live data streams from IoT devices, sensors, or application logs to validate streaming data pipelines' real-time processing capabilities.

14. Data Lake: Large volumes of structured, semi-structured, and unstructured data are stored in centralized repositories called "data lakes" in their original formats. These lakes offer scalable processing and storage capacities for machine learning, data analytics, and other data-driven applications.

Test case: Verify whether data lake storage options like Amazon S3 are elastic and scalable enough to handle substantial amounts of both structured and unstructured data.

15. Lakehouse Architecture: By combining the advantages of data lakes and data warehouses, lakehouse architecture allows businesses to store and process both structured and unstructured data on a single platform. This is made possible by the use of AWS Glue, Amazon Redshift, and Apache Spark, which enhance performance, affordability, and manageability for a variety of analytics workloads and use cases.

Test case: Check that queries are responding in accordance with real-time data.

**Differentiate between Monolith vs Micro-service Architecture.**

|  |  |
| --- | --- |
| Monolith | Micro-service Architecture. |
| A single code base supporting several interconnected features. | Software components that operate independently and exchange data with one another via APIs. |
| Starts out requiring less planning but becomes more difficult to comprehend and manage. | Initially requires more infrastructure and preparation, but it becomes easier to maintain and administer with time. |
| The complete application is launched as a unified unit. | Since each microservice is a separate software component, each one needs to be deployed individually using containers. |
| Follow the code path in the identical setting. | Needs sophisticated debugging tools in order to track the data transfer between various microservices. |
| Because they affect the entire code base, small changes come with more dangers. | Modifying a single microservice won't affect the application as a whole. |
| Even if demand for only some functional parts of the application increases, you still need to scale the entire application. | Scaling costs can be reduced overall since individual microservices can be scaled as needed. |
| Low upfront investment at the expense of increased continuing and maintenance costs. | More time and money will be needed to establish the necessary infrastructure and improve team performance. But long-term financial savings, upkeep, and flexibility. |

1. **Write about the following AWS services.**

1. **S3 and S3 Glacier**: While S3 Glacier offers safe and reasonably priced archive storage for long-term data retention, Amazon S3 enables scalable object storage for a range of data types.

2. **Redshift, Amazon RDS, and DynamoDB:** Different database needs and use cases are served by different services; Redshift is a fully managed data warehousing solution, RDS provides managed relational databases, and DynamoDB is a fully managed NoSQL database service.

3. **EC2 and Lightsail:** EC2 delivers scalable virtual servers in the cloud with complete control over server instances, whereas Lightsail is a simpler, easier-to-use service for deploying and managing virtual private servers (VPS) using pre-configured templates.

4. **Lambda:** Lambda enables serverless computing and event-driven architecture by enabling code execution without the need for server provisioning or management.

5. **Amazon SNS:** A completely managed messaging solution, Amazon Simple Notification Service allows messages to be delivered to distributed systems via push notifications, SMS, email, and other endpoints.

6**. DynamoDB:** For applications needing low-latency data access, DynamoDB is a fully managed NoSQL database service that offers single-digit millisecond performance at any scale.

**7. CloudWatch and CloudTrail:** While CloudTrail records AWS API calls and related events to allow governance, compliance, and audit trail logging, CloudWatch monitors and observability for AWS resources and applications.

**8. Sagemaker**: Machine learning models may be created, trained, and implemented at scale by developers and data scientists using Amazon SageMaker, a fully managed service.

**9. Step Functions:** Using visual workflows to automate application operations and coordinate numerous AWS services, AWS Step Functions is a serverless orchestration technology that enables you to organize various AWS services into serverless workflows.