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 a centralized metadata repository, often implemented using services like AWS Glue, AWS Lake Formation, and Amazon Athena, allowing users to efficiently discover, manage, and analyse data assets across various data sources within their organization.

Test Case: Verify that the data catalog automatically discovers and indexes new datasets added to Amazon S3 buckets, ensuring comprehensive coverage of available data assets.

1. Data Anonymization/Data Masking: Data anonymization or data masking involves obscuring sensitive information within datasets to protect the privacy of individuals, typically achieved by replacing identifiable data with fictitious or generalized values while preserving the dataset's utility for analysis or testing purposes.

Test Case: Validate that sensitive fields containing personally identifiable information (PII) are properly masked or anonymized in sample datasets, preserving data utility while protecting privacy.

1. PII Data: Personally Identifiable Information (PII) data refers to any information that can be used to identify an individual, such as names, social security numbers, email addresses, or biometric data, which must be handled with care to comply with privacy regulations.

Test Case: Validate the encryption and decryption mechanisms used to protect PII data at rest and in transit within AWS storage services such as Amazon S3 or Amazon RDS.

1. Data Democratization: Data democratization refers to the process of providing access to data and analytics capabilities to a broader range of users within an organization, enabling self-service analytics and empowering users to make data-driven decisions without extensive technical expertise.

Test Case: Evaluate the accessibility and usability of self-service analytics tools such as Amazon QuickSight or Tableau Server by conducting user acceptance testing with business stakeholders.

1. Data Modeling: Data modeling involves designing the structure and relationships of data entities within a database or data warehouse, typically using techniques such as entity-relationship diagrams or schema design to organize and represent data for efficient storage and retrieval.

Test Case: Validate the accuracy and completeness of entity-relationship diagrams (ERDs) and data models by comparing them against business requirements and data source documentation.

1. Dashboards: Dashboards are visual interfaces that display key performance indicators (KPIs), metrics, and other relevant data in a concise and interactive format, allowing users to monitor and analyze data trends, make informed decisions, and track progress towards organizational goals.

Test Case: Validate the responsiveness and load times of dashboards built using Amazon QuickSight or Tableau Server by simulating concurrent user access and analyzing performance metrics.

1. Data Ecosystem: A data ecosystem encompasses the interconnected set of tools, services, and processes for managing, analyzing, and deriving value from data across an organization, including data storage, processing, integration, analytics, and governance components.\

Test Case: Validate end-to-end data flows within the ecosystem by tracing data movement from source systems to target storage and analytics platforms, ensuring data integrity and reliability.

1. Data Enrichment: Data enrichment involves enhancing existing datasets with additional contextual information or attributes obtained from external sources, such as demographic data, geographic information, or market trends, to improve the quality and relevance of the data for analysis or decision-making.

Test Case: Verify the accuracy and completeness of data enrichment processes by comparing enriched datasets against known reference data sources or ground truth values.

1. Data Exchange: Data exchange involves the secure sharing and transfer of data between different systems, applications, or organizations, typically facilitated by APIs, data pipelines, or integration platforms to enable seamless data interoperability and collaboration.

Test Case: Validate the reliability and performance of data exchange mechanisms, such as AWS Data Exchange or AWS Direct Connect, by transferring large datasets between different AWS accounts or regions.

1. Data Extraction: Data extraction refers to the process of retrieving data from various sources, such as databases, files, or APIs, and transferring it to a destination for further processing, analysis, or storage, often performed using ETL (Extract, Transform, Load) tools or services.

Test Case: Validate the accuracy and completeness of data extraction processes by comparing extracted datasets against source data records and verifying that all relevant data is captured.

1. Data Governance: Data governance encompasses the policies, processes, and controls for ensuring the availability, integrity, security, and compliance of data assets across an organization, including data management, access control, and regulatory compliance measures.

Test Case: Validate that data governance policies are effectively enforced across AWS services such as Amazon S3, Amazon RDS, and Amazon Redshift, ensuring compliance with regulatory requirements and organizational data policies.

1. Data Ingestion: Data ingestion involves the process of collecting, receiving, and loading data from various sources into a data storage or processing system, such as a data lake, data warehouse, or analytics platform, typically performed using automated pipelines or ingestion tools.

Test Case: Validate the scalability and performance of data ingestion pipelines by simulating varying data volumes and ingestion rates, ensuring that pipelines can handle peak loads without data loss or degradation in performance.

1. Data Joins: Data joins refer to the operation of combining related datasets based on common keys or attributes to create a unified view of the data, enabling analysis and insights generation across multiple sources, often performed in SQL queries or data processing workflows.

Test Case: Validate the correctness of join operations performed on datasets stored in Amazon Redshift or Amazon Athena by comparing query results against expected output based on predefined join conditions.

1. Data Lineage: Data lineage involves tracking the origins, transformations, and movements of data throughout its lifecycle, providing visibility into how data is created, used, and modified across different systems, processes, and analytical workflows.

Test Case: Validate the accuracy and completeness of data lineage information captured by AWS services such as AWS Glue or AWS Lake Formation, ensuring that data lineage traces the end-to-end flow of data from source to destination.

1. Data Mesh: Data mesh is an architectural approach that advocates for decentralizing data ownership and management by treating data as a product, enabling cross-functional teams to manage their own data domains and providing self-serve data infrastructure and tools to enable data democratization and agility.

Test Case: Validate the autonomy and scalability of individual data domains within the data mesh architecture by assessing their ability to independently manage and evolve their data assets.

Data terms:

1. Data Portability: Data portability refers to the ability to easily move data between different storage systems, services, or regions within the AWS ecosystem, ensuring flexibility and agility in managing data assets.

Test case: Export a dataset from Amazon S3 to Google Cloud Storage and verify successful import without data loss or corruption.

2. Data Replication: Data replication involves copying data from one storage location or service to another, typically for purposes such as disaster recovery, high availability, or data distribution across multiple regions or environments.

Test case: Set up data replication between Amazon RDS instances in different regions and verify consistency of replicated data.

3. Data Privacy: Data privacy involves protecting sensitive information and ensuring compliance with data protection regulations by implementing appropriate security measures, encryption techniques, access controls, and data governance practices to safeguard data against unauthorized access or disclosure.

Test case: Encrypt data stored in Amazon S3 using server-side encryption with AWS Key Management Service (KMS) and verify data decryption using the correct key.

4. Data Consistency: Data consistency refers to the reliability and accuracy of data across different systems or replicas, ensuring that all copies of the data are synchronized and up-to-date to maintain data integrity and reliability for applications and users.

Test case: Verify the consistency of replicated data across multi-AZ deployments of Amazon RDS by comparing data snapshots.

5. Data Quality: Data quality refers to the level of accuracy, completeness, consistency, and reliability of data stored and processed within the AWS ecosystem, often addressed through data validation, cleansing, enrichment, and monitoring processes to ensure high-quality data for analytics and decision-making.

Test case: Validate data validation rules and constraints during data ingestion into Amazon Redshift to ensure that only high-quality data is loaded.

6. Data Silo: A data silo refers to a situation where data is stored or managed in isolated or fragmented systems or environments, leading to inefficiencies, duplication, and barriers to data sharing and collaboration across an organization.

Test case: Identify existing data silos within the organization by conducting data inventory assessments and analyzing data flow diagrams.

7. Data Validation: Data validation involves the process of checking and verifying the accuracy, integrity, and compliance of data against predefined rules, standards, or requirements, typically performed using automated validation routines or manual review processes to ensure data quality and reliability.

Test case: Implement automated data validation checks during data ingestion processes to verify data completeness, accuracy, and consistency against predefined validation rules.

8. Data Wrangling: Data wrangling refers to the process of preparing and transforming raw or unstructured data into a usable format for analysis or consumption, involving tasks such as cleansing, parsing, aggregating, and structuring data using tools and services like AWS Glue, Data Pipeline, or Amazon EMR.

Test case: 1: Evaluate the usability and effectiveness of data wrangling tools such as AWS Glue DataBrew or Amazon SageMaker Data Wrangler by conducting user acceptance testing with data analysts and business users.

9. Database Schema: A database schema defines the structure, organization, and relationships of data elements within a database, specifying the tables, fields, constraints, and indexes that govern how data is stored, accessed, and manipulated.

Test case: Validate the correctness of database schema design by comparing it against data modeling best practices and industry standards.

10. Data Stewardship: Data stewardship involves assigning responsibility and accountability for managing and protecting data assets within an organization, including defining data governance policies, overseeing data usage and access, and ensuring compliance with data privacy regulations and industry standards.

Test case: Define and enforce data stewardship roles and responsibilities within the organization to ensure clear accountability for data quality, security, and compliance.

11. EDI Data Standards: Electronic Data Interchange (EDI) data standards define formats, protocols, and syntax for exchanging structured data electronically between different business systems or trading partners, facilitating seamless integration and interoperability of data across supply chains and business processes.

Test case: Validate compliance with EDI data standards such as ANSI X12 or EDIFACT by parsing and validating EDI messages against the respective standards.

12. Observability: Observability refers to the ability to monitor, measure, and understand the behavior and performance of distributed systems, applications, and services, using metrics, logs, traces, and other telemetry data to identify issues, troubleshoot problems, and optimize resource utilization and user experience.

Test case: Validate the effectiveness of monitoring and logging configurations by generating simulated workload and analyzing logs and metrics for performance, errors, and resource utilization.

13. Streaming Data: Streaming data refers to continuous and real-time data streams generated by devices, sensors, applications, or online transactions, which are processed, analyzed, and acted upon in near-real-time using streaming data services such as Amazon Kinesis or AWS Lambda to enable use cases such as real-time analytics, monitoring, and alerting.

Test case: Validate the real-time processing capabilities of streaming data pipelines by ingesting and processing live data streams from IoT devices, sensors, or application logs.

14. Data Lake: A data lake is a centralized repository that stores large volumes of structured, semi-structured, and unstructured data in its native format, providing scalable storage and processing capabilities for data analytics, machine learning, and other data-driven applications.

Test case: Validate the scalability and elasticity of data lake storage solutions such as Amazon S3 to accommodate large volumes of structured and unstructured data.

1. Lakehouse Architecture: Lakehouse architecture combines the features and benefits of data lakes and data warehouses, enabling organizations to store and analyze both raw and structured data in a unified platform, leveraging services like AWS Glue, Amazon Redshift, and Apache Spark to support diverse analytics workloads and use cases with improved performance, cost-effectiveness, and ease of management.

Test case: Verify responsiveness of queries against real-time data.

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

|  |  |  |
| --- | --- | --- |
| Category | Monolith | Micro-service Architecture. |
| Design | Single code base with multiple interdependent functions. | Independent software components with autonomous functionality that communicate with each other using APIs. |
| Development | Requires less planning at the start, but gets increasingly complex to understand and maintain. | Requires more planning and infrastructure at the start, but gets easier to manage and maintain over time. |
| Deployment | Entire application deployed as a single entity. | Every microservice is an independent software entity that requires individual containerized deployment. |
| Debugging | Trace the code path in the same environment. | Requires advanced debugging tools to trace the data exchange between multiple microservices. |
| Modification | Small changes introduce greater risks as they impact the entire code base. | You can modify individual microservices without impacting the entire application. |
| Scale | You have to scale the entire application, even if only certain functional areas experience an increase in demand. | You can scale individual microservices as required, which saves overall scaling costs. |
| Investment | Low upfront investment at the cost of increased ongoing and maintenance efforts. | Additional time and cost investment to set up the required infrastructure and build team competency. However, long-term cost savings, maintenance, and adaptability. |

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

1. **S3 and S3 Glacier**: Amazon S3 provides scalable object storage for a variety of data types, while S3 Glacier offers secure and cost-effective archival storage for long-term data retention.

2. **Redshift, Amazon RDS, and DynamoDB:** Redshift is a fully managed data warehousing service, RDS offers managed relational databases, and DynamoDB is a fully managed NoSQL database service, each catering to different database needs and use cases.

3. **EC2 and Lightsail:** EC2 provides scalable virtual servers in the cloud with full control over server instances, while Lightsail offers a simpler, easy-to-use service for launching and managing virtual private servers (VPS) with pre-configured templates.

4. **Lambda:** Lambda allows you to run code without provisioning or managing servers, enabling serverless computing and event-driven architecture.

5. **Amazon SNS:** Amazon Simple Notification Service is a fully managed messaging service that enables message delivery through push notifications, SMS, email, and other endpoints to distributed systems.

6**. DynamoDB:** DynamoDB is a fully managed NoSQL database service that delivers single-digit millisecond performance at any scale, making it ideal for applications requiring low-latency data access.

**7. CloudWatch and CloudTrail:** CloudWatch provides monitoring and observability for AWS resources and applications, while CloudTrail enables governance, compliance, and audit trail logging by recording AWS API calls and related events.

**8. Sagemaker**: Amazon SageMaker is a fully managed service that enables developers and data scientists to build, train, and deploy machine learning models at scale.

**9. Step Functions:** AWS Step Functions is a serverless orchestration service that allows you to coordinate multiple AWS services into serverless workflows using visual workflows to automate application workflows and coordinate multiple AWS services.