Content outline

This exam guide includes weightings, content domains, and task statements for the exam. This guide does not provide a comprehensive list of the content on the exam. However, additional context for each task statement is available to help you prepare for the exam.

The exam has the following content domains and weightings:

• Domain 1: Data Ingestion and Transformation (34% of scored content)

• Domain 2: Data Store Management (26% of scored content)

• Domain 3: Data Operations and Support (22% of scored content)

• Domain 4: Data Security and Governance (18% of scored content)

Domain 1: Data Ingestion and Transformation

Task Statement 1.1: Perform data ingestion.

Knowledge of:

• Throughput and latency characteristics for AWS services that ingest data

• Data ingestion patterns (for example, frequency and data history)

• Streaming data ingestion

• Batch data ingestion (for example, scheduled ingestion, event-driven ingestion)

• Replayability of data ingestion pipelines

• Stateful and stateless data transactions

Skills in:

• Reading data from streaming sources (for example, Amazon Kinesis, Amazon Managed Streaming for Apache Kafka [Amazon MSK], Amazon DynamoDB Streams, AWS Database Migration Service [AWS DMS], AWS Glue, Amazon Redshift)

• Reading data from batch sources (for example, Amazon S3, AWS Glue, Amazon EMR, AWS DMS, Amazon Redshift, AWS Lambda, Amazon AppFlow)

• Implementing appropriate configuration options for batch ingestion

• Consuming data APIs

• Setting up schedulers by using Amazon EventBridge, Apache Airflow, or time-based schedules for jobs and crawlers

• Setting up event triggers (for example, Amazon S3 Event Notifications, EventBridge)

• Calling a Lambda function from Amazon Kinesis

• Creating allowlists for IP addresses to allow connections to data sources

• Implementing throttling and overcoming rate limits (for example, DynamoDB, Amazon RDS, Kinesis)

• Managing fan-in and fan-out for streaming data distribution

Task Statement 1.2: Transform and process data.

Knowledge of:

• Creation of ETL pipelines based on business requirements

• Volume, velocity, and variety of data (for example, structured data, unstructured data) • Cloud computing and distributed computing

• How to use Apache Spark to process data

• Intermediate data staging locations

Skills in:

• Optimizing container usage for performance needs (for example, Amazon Elastic Kubernetes Service [Amazon EKS], Amazon Elastic Container Service [Amazon ECS])

• Connecting to different data sources (for example, Java Database Connectivity [JDBC], Open Database Connectivity [ODBC])

• Integrating data from multiple sources

• Optimizing costs while processing data

• Implementing data transformation services based on requirements (for example, Amazon EMR, AWS Glue, Lambda, Amazon Redshift)

• Transforming data between formats (for example, from .csv to Apache Parquet)

• Troubleshooting and debugging common transformation failures and performance issues

• Creating data APIs to make data available to other systems by using AWS services

Task Statement 1.3: Orchestrate data pipelines.

Knowledge of:

• How to integrate various AWS services to create ETL pipelines

• Event-driven architecture

• How to configure AWS services for data pipelines based on schedules or dependencies

• Serverless workflows

Skills in:

• Using orchestration services to build workflows for data ETL pipelines (for example, Lambda, EventBridge, Amazon Managed Workflows for Apache Airflow [Amazon MWAA], AWS Step Functions, AWS Glue workflows)

• Building data pipelines for performance, availability, scalability, resiliency, and fault tolerance

• Implementing and maintaining serverless workflows

• Using notification services to send alerts (for example, Amazon Simple Notification Service [Amazon SNS], Amazon Simple Queue Service [Amazon SQS])

Task Statement 1.4: Apply programming concepts.

Knowledge of:

• Continuous integration and continuous delivery (CI/CD) (implementation, testing, and deployment of data pipelines)

• SQL queries (for data source queries and data transformations)

• Infrastructure as code (IaC) for repeatable deployments (for example, AWS Cloud Development Kit [AWS CDK], AWS CloudFormation)

• Distributed computing

• Data structures and algorithms (for example, graph data structures and tree data structures)

• SQL query optimization

Skills in:

• Optimizing code to reduce runtime for data ingestion and transformation

• Configuring Lambda functions to meet concurrency and performance needs

• Performing SQL queries to transform data (for example, Amazon Redshift stored procedures)

• Structuring SQL queries to meet data pipeline requirements

• Using Git commands to perform actions such as creating, updating, cloning, and branching repositories

• Using the AWS Serverless Application Model (AWS SAM) to package and deploy serverless data pipelines (for example, Lambda functions, Step Functions, DynamoDB tables)

• Using and mounting storage volumes from within Lambda functions

Domain 2: Data Store Management

Task Statement 2.1: Choose a data store.

Knowledge of:

• Storage platforms and their characteristics

• Storage services and configurations for specific performance demands

• Data storage formats (for example, .csv, .txt, Parquet)

• How to align data storage with data migration requirements

• How to determine the appropriate storage solution for specific access patterns

• How to manage locks to prevent access to data (for example, Amazon Redshift, Amazon RDS)

Skills in:

• Implementing the appropriate storage services for specific cost and performance requirements (for example, Amazon Redshift, Amazon EMR, AWS Lake Formation, Amazon RDS, DynamoDB, Amazon Kinesis Data Streams, Amazon MSK)

• Configuring the appropriate storage services for specific access patterns and requirements (for example, Amazon Redshift, Amazon EMR, Lake Formation, Amazon RDS, DynamoDB)

• Applying storage services to appropriate use cases (for example, Amazon S3)

• Integrating migration tools into data processing systems (for example, AWS Transfer Family)

• Implementing data migration or remote access methods (for example, Amazon Redshift federated queries, Amazon Redshift materialized views, Amazon Redshift Spectrum)

Task Statement 2.2: Understand data cataloging systems.

Knowledge of:

• How to create a data catalog

• Data classification based on requirements

• Components of metadata and data catalogs

Skills in:

• Using data catalogs to consume data from the data's source

• Building and referencing a data catalog (for example, AWS Glue Data Catalog, Apache Hive metastore)

• Discovering schemas and using AWS Glue crawlers to populate data catalogs

• Synchronizing partitions with a data catalog

• Creating new source or target connections for cataloging (for example, AWS Glue)

Task Statement 2.3: Manage the lifecycle of data.

Knowledge of:

• Appropriate storage solutions to address hot and cold data requirements

• How to optimize the cost of storage based on the data lifecycle

• How to delete data to meet business and legal requirements

• Data retention policies and archiving strategies

• How to protect data with appropriate resiliency and availability

Skills in:

• Performing load and unload operations to move data between Amazon S3 and Amazon Redshift

• Managing S3 Lifecycle policies to change the storage tier of S3 data

• Expiring data when it reaches a specific age by using S3 Lifecycle policies

• Managing S3 versioning and DynamoDB TTL

Task Statement 2.4: Design data models and schema evolution.

Knowledge of:

• Data modeling concepts

• How to ensure accuracy and trustworthiness of data by using data lineage

• Best practices for indexing, partitioning strategies, compression, and other data optimization techniques

• How to model structured, semi-structured, and unstructured data

• Schema evolution techniques

Skills in:

• Designing schemas for Amazon Redshift, DynamoDB, and Lake Formation

• Addressing changes to the characteristics of data

• Performing schema conversion (for example, by using the AWS Schema Conversion Tool [AWS SCT] and AWS DMS Schema Conversion)

• Establishing data lineage by using AWS tools (for example, Amazon SageMaker ML Lineage Tracking)

Domain 3: Data Operations and Support

Task Statement 3.1: Automate data processing by using AWS services.

Knowledge of:

• How to maintain and troubleshoot data processing for repeatable business outcomes

• API calls for data processing

• Which services accept scripting (for example, Amazon EMR, Amazon Redshift, AWS Glue)

Skills in:

• Orchestrating data pipelines (for example, Amazon MWAA, Step Functions)

• Troubleshooting Amazon managed workflows

• Calling SDKs to access Amazon features from code

• Using the features of AWS services to process data (for example, Amazon EMR, Amazon Redshift, AWS Glue)

• Consuming and maintaining data APIs

• Preparing data transformation (for example, AWS Glue DataBrew)

• Querying data (for example, Amazon Athena)

• Using Lambda to automate data processing

• Managing events and schedulers (for example, EventBridge)

Task Statement 3.2: Analyze data by using AWS services.

Knowledge of:

• Tradeoffs between provisioned services and serverless services

• SQL queries (for example, SELECT statements with multiple qualifiers or JOIN clauses)

• How to visualize data for analysis

• When and how to apply cleansing techniques

• Data aggregation, rolling average, grouping, and pivoting

Skills in:

• Visualizing data by using AWS services and tools (for example, AWS Glue DataBrew, Amazon QuickSight)

• Verifying and cleaning data (for example, Lambda, Athena, QuickSight, Jupyter Notebooks, Amazon SageMaker Data Wrangler)

• Using Athena to query data or to create views

• Using Athena notebooks that use Apache Spark to explore data

Task Statement 3.3: Maintain and monitor data pipelines.

Knowledge of:

• How to log application data

• Best practices for performance tuning

• How to log access to AWS services

• Amazon Macie, AWS CloudTrail, and Amazon CloudWatch

Skills in:

• Extracting logs for audits

• Deploying logging and monitoring solutions to facilitate auditing and traceability

• Using notifications during monitoring to send alerts

• Troubleshooting performance issues

• Using CloudTrail to track API calls

• Troubleshooting and maintaining pipelines (for example, AWS Glue, Amazon EMR)

• Using Amazon CloudWatch Logs to log application data (with a focus on configuration and automation)

• Analyzing logs with AWS services (for example, Athena, Amazon EMR, Amazon OpenSearch Service, CloudWatch Logs Insights, big data application logs)

Task Statement 3.4: Ensure data quality.

Knowledge of:

• Data sampling techniques

• How to implement data skew mechanisms

• Data validation (data completeness, consistency, accuracy, and integrity)

• Data profiling

Skills in:

• Running data quality checks while processing the data (for example, checking for empty fields)

• Defining data quality rules (for example, AWS Glue DataBrew)

• Investigating data consistency (for example, AWS Glue DataBrew)

Domain 4: Data Security and Governance

Task Statement 4.1: Apply authentication mechanisms.

Knowledge of:

• VPC security networking concepts

• Differences between managed services and unmanaged services

• Authentication methods (password-based, certificate-based, and role-based)

• Differences between AWS managed policies and customer managed policies

Skills in:

• Updating VPC security groups

• Creating and updating IAM groups, roles, endpoints, and services

• Creating and rotating credentials for password management (for example, AWS Secrets Manager)

• Setting up IAM roles for access (for example, Lambda, Amazon API Gateway, AWS CLI, CloudFormation)

• Applying IAM policies to roles, endpoints, and services (for example, S3 Access Points, AWS PrivateLink)

Task Statement 4.2: Apply authorization mechanisms.

Knowledge of:

• Authorization methods (role-based, policy-based, tag-based, and attributebased)

• Principle of least privilege as it applies to AWS security

• Role-based access control and expected access patterns

• Methods to protect data from unauthorized access across services

Skills in:

• Creating custom IAM policies when a managed policy does not meet the needs

• Storing application and database credentials (for example, Secrets Manager, AWS Systems Manager Parameter Store)

• Providing database users, groups, and roles access and authority in a database (for example, for Amazon Redshift)

• Managing permissions through Lake Formation (for Amazon Redshift, Amazon EMR, Athena, and Amazon S3)

Task Statement 4.3: Ensure data encryption and masking.

Knowledge of:

• Data encryption options available in AWS analytics services (for example, Amazon Redshift, Amazon EMR, AWS Glue)

• Differences between client-side encryption and server-side encryption

• Protection of sensitive data

• Data anonymization, masking, and key salting

Skills in:

• Applying data masking and anonymization according to compliance laws or company policies

• Using encryption keys to encrypt or decrypt data (for example, AWS Key Management Service [AWS KMS])

• Configuring encryption across AWS account boundaries

• Enabling encryption in transit for data.

Task Statement 4.4: Prepare logs for audit.

Knowledge of:

• How to log application data

• How to log access to AWS services

• Centralized AWS logs

Skills in:

• Using CloudTrail to track API calls

• Using CloudWatch Logs to store application logs

• Using AWS CloudTrail Lake for centralized logging queries

• Analyzing logs by using AWS services (for example, Athena, CloudWatch Logs Insights, Amazon OpenSearch Service)

• Integrating various AWS services to perform logging (for example, Amazon EMR in cases of large volumes of log data)

Task Statement 4.5: Understand data privacy and governance.

Knowledge of:

• How to protect personally identifiable information (PII)

• Data sovereignty

Skills in:

• Granting permissions for data sharing (for example, data sharing for Amazon Redshift)

• Implementing PII identification (for example, Macie with Lake Formation)

• Implementing data privacy strategies to prevent backups or replications of data to disallowed AWS Regions

• Managing configuration changes that have occurred in an account (for example, AWS Config)