The following services will be discussed as part of this converged course.

Task Statement 1: Design secure access to AWS resources.

Learning objectives:

- Applying AWS security best practices to IAM users and root users (for example, multi-factor authentication [MFA])
- Designing a flexible authorization model that includes IAM users, groups, roles, and policies
- Designing a role-based access control strategy (for example, AWS Security Token Service [AWS STS], role switching, cross-account access)
- Designing a security strategy for multiple AWS accounts (for example, AWS Control Tower, service control policies [SCPs])
- Determining the appropriate use of resource policies for AWS services
- Determining when to federate a directory service with IAM roles

Task Statement 2: Design secure workloads and applications.

Learning objectives:

- Designing VPC architectures with security components (for example, security groups, route tables, network ACLs, NAT gateways)
- Determining network segmentation strategies (for example, using public subnets and private subnets)
- Integrating AWS services to secure applications (for example, AWS Shield, AWS WAF, AWS SSO, AWS Secrets Manager)
- Securing external network connections to and from the AWS Cloud (for example, VPN, AWS Direct Connect)

Task Statement 3: Determine appropriate data security controls.

Learning objectives:

- Aligning AWS technologies to meet compliance requirements
- Encrypting data at rest (for example, AWS Key Management Service [AWS KMS])
- Encrypting data in transit (for example, AWS Certificate Manager [ACM] using TLS)
- Implementing access policies for encryption keys
- Implementing data backups and replications
- Implementing policies for data access, lifecycle, and protection
- Rotating encryption keys and renewing certificates

Task Statement 4: Design scalable and loosely coupled architectures.

Learning objectives:

- Designing event-driven, microservice, and/or multi-tier architectures based on requirements
- Determining scaling strategies for components used in an architecture design

- Determining the AWS services required to achieve loose coupling based on requirements
- Determining when to use containers
- Determining when to use serverless technologies and patterns
- Recommending appropriate compute, storage, networking, and database technologies based on requirements
- Using purpose-built AWS services for workloads

Task Statement 5: Design highly available and/or fault-tolerant architectures.

Learning objectives:

- Determining automation strategies to ensure infrastructure integrity
- Determining the AWS services required to provide a highly available and/or fault-tolerant architecture across AWS Regions or Availability Zones
- Identifying metrics based on business requirements to deliver a highly available solution
- Implementing designs to mitigate single points of failure
- Implementing strategies to ensure the durability and availability of data (for example, backups) Selecting an appropriate DR strategy to meet business requirements
- Using AWS services that improve the reliability of legacy applications and applications not built for the cloud (for example, when application changes are not possible)
- Using purpose-built AWS services for workloads

Task Statement 6: Determine high-performing and/or scalable storage solutions

Learning objectives:

- Determining storage services and configurations that meet performance demands
- Determining storage services that can scale to accommodate future needs

Task Statement 7: Design high-performing and elastic compute solutions.

Learning objectives:

- Decoupling workloads so that components can scale independently
- Identifying metrics and conditions to perform scaling actions
- Selecting the appropriate compute options and features (for example, EC2 instance types) to meet business requirements
- Selecting the appropriate resource type and size (for example, the amount of Lambda memory) to meet business requirements

Task Statement 8: Determine high-performing database solutions.

Learning objectives:

- Configuring read replicas to meet business requirements
- Designing database architectures
- Determining an appropriate database engine (for example, MySQL compared with PostgreSQL

- Determining an appropriate database type (for example, Amazon Aurora, Amazon DynamoDB
- Integrating caching to meet business requirements

Task Statement 9: Determine high-performing and/or scalable network architectures.

Knowledge of:

- Edge networking services with appropriate use cases (for example, Amazon CloudFront, AWS Global Accelerator)
- How to design network architecture (for example, subnet tiers, routing, IP addressing)
- Load balancing concepts (for example, Application Load Balancer)
- Network connection options (for example, AWS VPN, Direct Connect, AWS PrivateLink) Learning objectives:
- Creating a network topology for various architectures (for example, global, hybrid, multi-tier)
- Determining network configurations that can scale to accommodate future needs
- Determining the appropriate placement of resources to meet business requirements
- Selecting the appropriate load balancing strategy

Task Statement 10: Determine high-performing data ingestion and transformation solutions.

Learning objectives:

- Building and securing data lakes
- Designing data streaming architectures
- Designing data transfer solutions
- Implementing visualization strategies
- Selecting appropriate compute options for data processing (for example, Amazon EMR)
- Selecting appropriate configurations for ingestion
- Transforming data between formats (for example, .csv to .parquet)

Task Statement 11: Design cost-optimized storage solutions.

Learning objectives:

- Designing appropriate storage strategies (for example, batch uploads to Amazon S3 compared with individual uploads)
- Determining the correct storage size for a workload
- Determining the lowest cost method of transferring data for a workload to AWS storage
- Determining when storage auto scaling is required
- Managing S3 object lifecycles
- Selecting the appropriate backup and/or archival solution
- Selecting the appropriate service for data migration to storage services
- Selecting the appropriate storage tier
- Selecting the correct data lifecycle for storage
- Selecting the most cost-effective storage service for a workload

Task Statement 12: Design cost-optimized compute solutions.

Learning objectives:

- Determining an appropriate load balancing strategy (for example, Application Load Balancer [Layer 7] compared with Network Load Balancer [Layer 4] compared with Gateway Load Balancer)
- Determining appropriate scaling methods and strategies for elastic workloads (for example, horizontal compared with vertical, EC2 hibernation)
- Determining cost-effective AWS compute services with appropriate use cases (for example, Lambda, Amazon EC2, Fargate)
- Determining the required availability for different classes of workloads (for example, production workloads, non-production workloads)
- Selecting the appropriate instance family for a workload
- Selecting the appropriate instance size for a workload

Task Statement 13: Design cost-optimized database solutions.

Learning objectives:

- Designing appropriate backup and retention policies (for example, snapshot frequency)
- Determining an appropriate database engine (for example, MySQL compared with PostgreSQL)
- Determining cost-effective AWS database services with appropriate use cases (for example, DynamoDB compared with Amazon RDS, serverless)
- Determining cost-effective AWS database types (for example, time series format, columnar format)
- Migrating database schemas and data to different locations and/or different database engines

Task Statement 14: Design cost-optimized network architectures.

Learning objectives:

- Configuring appropriate NAT gateway types for a network (for example, a single shared NAT gateway compared with NAT gateways for each Availability Zone)
- Configuring appropriate network connections (for example, Direct Connect compared with VPN compared with internet)
- Configuring appropriate network routes to minimize network transfer costs (for example, Region to Region, Availability Zone to Availability Zone, private to public, Global Accelerator, VPC endpoints)
- Determining strategic needs for content delivery networks (CDNs) and edge caching
- Reviewing existing workloads for network optimizations
- Selecting an appropriate throttling strategy
-
- Selecting the appropriate bandwidth allocation for a network device (for example, a single VPN compared with multiple VPNs, Direct Connect speed.