

AWS Domain 1: Design for Organizational Complexity

1.1 Determine cross-account authentication and access strategy for complex organizations.

- Analyze the organizational structure
- Evaluate the current authentication infrastructure
- Analyze the AWS resources at an account level
- Determine an auditing strategy for authentication and access

1.2 Determine how to design networks for complex organizations.

- Outline an IP addressing strategy for VPCs
- Determine DNS strategy
- Classify network traffic and security
- Determine connectivity needs for hybrid environments
- Determine a way to audit network traffic

1.3 Determine how to design a multi-account AWS environment for complex organizations.

- Determine how to use AWS Organizations
- Implement the most appropriate account structure for proper cost allocation, agility, and security
- Recommend a central audit and event notification strategy
- Decide on an access strategy

AWS Domain 2: Design for New Solutions

2.1 Determine security requirements and controls when designing and implementing a solution.

- Implement infrastructure as code
- Determine prevention controls for large-scale web applications
- Determine roles and responsibilities of applications
- Determine a secure method to manage credentials for the solutions/applications
- Enable detection controls and security services for large-scale applications
- Enforce host and network security boundaries
- Enable encryption in transit and at rest

2.2 Determine a solution design and implementation strategy to meet reliability requirements.

- Design a highly available application environment
- Determine advanced techniques to detect for failure and service recoverability
- Determine processes and components to monitor and recover from regional service disruptions with regional failover

2.3 Determine a solution design to ensure business continuity.

- Architect an automated, cost-effective back-up solution that supports business continuity across multiple AWS Regions
- Determine an architecture that provides application and infrastructure availability in the event of a service disruption

2.4 Determine a solution design to meet performance objectives.

- Design internet-scale application architectures
- Design an architecture for performance according to business objectives
- Apply design patterns to meet business objectives with caches, buffering, and replicas

2.5 Determine a deployment strategy to meet business requirements when designing and implementing a solution.

- Determine resource provisioning strategy to meet business objectives
- Determine a migration process to change the version of a service
- Determine services to meet deployment strategy
- Determine patch management strategy

AWS Domain 3: Migration Planning

3.1 Select existing workloads and processes for potential migration to the cloud.

- Complete an application migration assessment
- Classify applications according to the six Rs (re-host, re-platform, re-purchase, refactor, retire, and retain)

3.2 Select migration tools and/or services for new and migrated solutions based on detailed AWS knowledge.

- Select an appropriate database transfer mechanism
- Select an appropriate data transfer service
- Select an appropriate data transfer target
- Select an appropriate server migration mechanism
- Apply the appropriate security methods to the migration tools

3.3 Determine a new cloud architecture for an existing solution.

- Evaluate business applications and determine the target cloud architecture
- Break down the functionality of applications into services
- Determine target database platforms

3.4 Determine a strategy for migrating existing on-premises workloads to the cloud.

- Determine the desired prioritization strategy of the organization
- Analyze data volume and rate of change to determine a data transfer strategy
- Evaluate cutover strategies
- Assess internal and external compliance requirements for a successful migration

AWS Domain 4: Cost Control

4.1 Select a cost-effective pricing model for a solution.

- Purchase resources based on usage requirements
- Identify when to use different storage tiers

4.2 Determine which controls to design and implement that will ensure cost optimization.

- Determine an AWS-generated cost allocation tags strategy that allows mapping cost to business units
- Determine a mechanism to monitor when underutilized resources are present
- Determine a way to manage commonly deployed resources to achieve governance
- Define a way to plan costs that do not exceed the budget amount

4.3 Identify opportunities to reduce cost in an existing architecture.

- Distinguish opportunities to use AWS Managed Services
- Determine which services are most cost-effective in meeting business objectives

AWS Domain 5: Continuous Improvement for Existing Solutions

5.1 Troubleshoot solutions architectures.

- Assess an existing application architecture for deficiencies
- Analyze application and infrastructure logs
- Test possible solutions in non-production environment

5.2 Determine a strategy to improve an existing solution for operational excellence.

- Determine the most appropriate logging and monitoring strategy
- Recommend the appropriate AWS offering(s) to enable configuration management automation

5.3 Determine a strategy to improve the reliability of an existing solution.

- Evaluate existing architecture to determine areas that are not sufficiently reliable
- Remediate single points of failure
- Enable data replication, self-healing, and elastic features and services
- Test the reliability of the new solution

5.4 Determine a strategy to improve the performance of an existing solution.

- Reconcile current performance metrics against performance targets
- Identify and examine performance bottlenecks
- Recommend and test potential remediation solutions

5.5 Determine a strategy to improve the security of an existing solution.

- Evaluate AWS Secrets Manager strategy
- Audit the environment for security vulnerabilities
- Enable manual and/or automated responses to the detection of vulnerabilities

5.6 Determine how to improve the deployment of an existing solution.

- Evaluate appropriate tooling to enable infrastructure as code
- Evaluate current deployment processes for improvement opportunities
- Test automated deployment and rollback strategies