ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

A company's ecommerce application is running on Amazon EC2 instances that are behind an Application Load Balancer (ALB). The instances are in an Auto Scaling group. Customers report that the website is occasionally down. When the website is down, it returns an HTTP 500 (server error) status code to customer browsers.

The Auto Scaling group's health check is configured for EC2 status checks, and the instances appear healthy.

Which solution will resolve the problem?

A. Replace the ALB with a Network Load Balancer.

B. Add Elastic Load Balancing (ELB) health checks to the Auto Scaling group.

C. Update the target group configuration on the ALB. Enable session affinity (sticky sessions).

D. Install the Amazon CloudWatch agent on all instances. Configure the agent to reboot the instances.

Answer: B

Explanation:

In this scenario, the EC2 instances pass their EC2 status checks, indicating that the operating system is responsive. However, the application hosted on the instance is failing intermittently, returning HTTP 500 errors. This demonstrates a discrepancy between the instance-level health and the application-level health.

According to AWS CloudOps best practices under Monitoring, Logging, Analysis, Remediation and Performance Optimization (SOA-C03 Domain 1), Auto Scaling groups should incorporate Elastic Load Balancing (ELB) health checks instead of relying solely on EC2 status checks. The ELB health check probes the application endpoint (for example, HTTP or HTTPS target group health checks), ensuring that the application itself is functioning correctly.

When an instance fails an ELB health check, Amazon EC2 Auto Scaling will automatically mark the instance as unhealthy and replace it with a new one, ensuring continuous availability and performance optimization.

Extract from AWS CloudOps (SOA-C03) Study Guide - Domain 1:

"Implement monitoring and health checks using ALB and EC2 Auto Scaling integration. Application Load Balancer health checks allow Auto Scaling to terminate and replace instances that fail application-level health checks, ensuring consistent application performance."

Extract from AWS Auto Scaling Documentation:

"When you enable the ELB health check type for your Auto Scaling group, Amazon EC2 Auto Scaling considers both EC2 status checks and Elastic Load Balancing health checks to determine instance health. If an instance fails the ELB health check, it is automatically replaced."

Therefore, the correct answer is B, as it ensures proper application-level monitoring and remediation using ALB-integrated ELB health checks—a core CloudOps operational practice for proactive incident response and availability assurance.

References (AWS CloudOps Verified Source Extracts):

AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide: Domain 1 - Monitoring, Logging, and Remediation.

AWS Auto Scaling User Guide: Health checks for Auto Scaling instances (Elastic Load Balancing integration).

AWS Well-Architected Framework - Operational Excellence and Reliability Pillars.

AWS Elastic Load Balancing Developer Guide - Target group health checks and monitoring.

Question: 2

A company hosts a critical legacy application on two Amazon EC2 instances that are in one Availability Zone. The instances run behind an Application Load Balancer (ALB). The company uses Amazon CloudWatch alarms to send Amazon Simple Notification Service (Amazon SNS) notifications when the ALB health checks detect an unhealthy instance. After a notification, the company's engineers manually restart the unhealthy instance. A CloudOps engineer must configure the application to be highly available and more resilient to failures. Which solution will meet these requirements?

A. Create an Amazon Machine Image (AMI) from a healthy instance. Launch additional instances from the AMI in the same Availability Zone. Add the new instances to the ALB target group.

B. Increase the size of each instance. Create an Amazon EventBridge rule. Configure the EventBridge rule to restart the instances if they enter a failed state.

C. Create an Amazon Machine Image (AMI) from a healthy instance. Launch an additional instance from the AMI in the same Availability Zone. Add the new instance to the ALB target group. Create an AWS Lambda function that runs when an instance is unhealthy. Configure the Lambda function to stop and restart the unhealthy instance.

D. Create an Amazon Machine Image (AMI) from a healthy instance. Create a launch template that uses the AMI. Create an Amazon EC2 Auto Scaling group that is deployed across multiple Availability Zones. Configure the Auto Scaling group to add instances to the ALB target group.

Answer: D

Explanation:

High availability requires removing single-AZ risk and eliminating manual recovery. The AWS Reliability best practices state to design for multi-AZ and automatic healing: Auto Scaling “helps maintain application

availability and allows you to automatically add or remove EC2 instances" (AWS Auto Scaling User Guide). The Reliability Pillar recommends to "distribute workloads across multiple Availability Zones" and to "automate recovery from failure" (AWS Well-Architected Framework - Reliability Pillar). Attaching the Auto Scaling group to an ALB target group enables health-based replacement: instances failing load balancer health checks are replaced and traffic is routed only to healthy targets. Using an AMI in a launch template ensures consistent, repeatable instance configuration (AWS EC2 Launch Templates). Options A and C keep all instances in a single Availability Zone and rely on manual or ad-hoc restarts, which do not meet high-availability or resiliency goals. Option B only scales vertically and adds a restart rule; it neither removes the single-AZ failure domain nor provides automated replacement. Therefore, creating a multi-AZ EC2 Auto Scaling group with a launch template and attaching it to the ALB target group (Option D) is the CloudOps-aligned solution for resilience and business continuity.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide: Domain 2 - Reliability and Business Continuity
- AWS Well-Architected Framework - Reliability Pillar
- Amazon EC2 Auto Scaling User Guide - Health checks and replacement
- Elastic Load Balancing User Guide - Target group health checks and ALB integration
- Amazon EC2 Launch Templates - Reproducible instance configuration

Question: 3

An Amazon EC2 instance is running an application that uses Amazon Simple Queue Service (Amazon SQS) queues. A CloudOps engineer must ensure that the application can read, write, and delete messages from the SQS queues.

Which solution will meet these requirements in the MOST secure manner?

A. Create an IAM user with an IAM policy that allows the sqs:SendMessage permission, the sqs:ReceiveMessage permission, and the sqs:DeleteMessage permission to the appropriate queues. Embed the IAM user's credentials in the application's configuration.

B. Create an IAM user with an IAM policy that allows the sqs:SendMessage permission, the sqs:ReceiveMessage permission, and the sqs:DeleteMessage permission to the appropriate queues. Export the IAM user's access key and secret access key as environment variables on the EC2 instance.

C. Create and associate an IAM role that allows EC2 instances to call AWS services. Attach an IAM policy to the role that allows sqs:* permissions to the appropriate queues.

D. Create and associate an IAM role that allows EC2 instances to call AWS services. Attach an IAM policy to the role that allows the sqs:SendMessage permission, the sqs:ReceiveMessage permission, and the

sqs:DeleteMessage permission to the appropriate queues.

Answer: D

Explanation:

The most secure pattern is to use an IAM role for Amazon EC2 with the minimum required permissions. AWS guidance states: “Use roles for applications that run on Amazon EC2 instances” and “grant least privilege by allowing only the actions required to perform a task.” By attaching a role to the instance, short-lived credentials are automatically provided through the instance metadata service; this removes the need to create long-term access keys or embed secrets. Granting only sqs:SendMessage, sqs:ReceiveMessage, and sqs:DeleteMessage against the specific SQS queues enforces least privilege and aligns with CloudOps security controls. Options A and B rely on IAM user access keys, which contravene best practices for workloads on EC2 and increase credentialmanagement risk. Option C uses a role but grants sqs:*, violating least-privilege principles. Therefore, Option D meets the security requirement with scoped, temporary credentials and precise permissions.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Security & Compliance
- IAM Best Practices - “Use roles instead of long-term access keys,” “Grant least privilege”
- IAM Roles for Amazon EC2 - Temporary credentials for applications on EC2
- Amazon SQS - Identity and access management for Amazon SQS

Question: 4

A company runs an application that logs user data to an Amazon CloudWatch Logs log group. The company discovers that personal information the application has logged is visible in plain text in the CloudWatch logs.

The company needs a solution to redact personal information in the logs by default. Unredacted information must be available only to the company's security team. Which solution will meet these requirements?

A. Create an Amazon S3 bucket. Create an export task from appropriate log groups in CloudWatch. Export the logs to the S3 bucket. Configure an Amazon Macie scan to discover personal data in the S3 bucket. Invoke an AWS Lambda function to move identified personal data to a second S3 bucket. Update the S3 bucket policies to grant only the security team access to both buckets.

B. Create a customer managed AWS KMS key. Configure the KMS key policy to allow only the security team to perform decrypt operations. Associate the KMS key with the application log group.

C. Create an Amazon CloudWatch data protection policy for the application log group. Configure data identifiers for the types of personal information that the application logs. Ensure that the security team has permission to call the unmask API operation on the application log group.

D. Create an OpenSearch domain. Create an AWS Glue workflow that runs a Detect PII transform job and streams the output to the OpenSearch domain. Configure the CloudWatch log group to stream the logs to AWS

Glue. Modify the OpenSearch domain access policy to allow only the security team to access the domain.

Answer: C

Explanation:

CloudWatch Logs data protection provides native redaction/masking of sensitive data at ingestion and query. AWS documentation states it can "detect and protect sensitive data in logs" using data identifiers, and that authorized users can "use the unmask action to view the original data." Creating a data protection policy on the log group masks PII by default for all viewers, satisfying the requirement to redact personal information. Granting only the security team permission to invoke the unmask API operation ensures that unredacted content is restricted. Option B (KMS) encrypts at rest but does not redact fields; encryption alone does not prevent plaintext visibility to authorized readers. Options A and D add complexity and latency, move data out of CloudWatch, and do not provide default inline redaction/unmask controls in CloudWatch itself. Therefore, the CloudOps- aligned, managed solution is to use CloudWatch Logs data protection with appropriate data identifiers and unmask permissions limited to the security team.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Monitoring & Logging
- Amazon CloudWatch Logs - Data Protection (masking/redaction with data identifiers)
- CloudWatch Logs - Permissions for masking and unmasking sensitive data
- AWS Well-Architected Framework - Security and Operational Excellence (sensitive data handling)

Question: 5

A multinational company uses an organization in AWS Organizations to manage over 200 member accounts across multiple AWS Regions. The company must ensure that all AWS resources meet specific security requirements.

The company must not deploy any EC2 instances in the ap-southeast-2 Region. The company must completely block root user actions in all member accounts. The company must prevent any user from deleting AWS CloudTrail logs, including administrators. The company requires a centrally managed solution that the company can automatically apply to all existing and future accounts. Which solution will meet these requirements?

A. Create AWS Config rules with remediation actions in each account to detect policy violations. Implement IAM permissions boundaries for the account root users.

B. Enable AWS Security Hub across the organization. Create custom security standards to enforce the security requirements. Use AWS CloudFormation StackSets to deploy the standards to all the accounts in the organization. Set up Security Hub automated remediation actions.

C. Use AWS Control Tower for account governance. Configure Region deny controls. Use Service Control Policies (SCPs) to restrict root user access.

D. Configure AWS Firewall Manager with security policies to meet the security requirements. Use an AWS Config aggregator with organization-wide conformance packs to detect security policy violations.

Answer: C

Explanation:

AWS CloudOps governance best practices emphasize centralized account management and preventive guardrails. AWS Control Tower integrates directly with AWS Organizations and provides "Region deny controls" and "Service Control Policies (SCPs)" that apply automatically to all existing and newly created member accounts. SCPs are organization-wide guardrails that define the maximum permissions for accounts. They can explicitly deny actions such as launching EC2 instances in a specific Region, or block root user access.

To prevent CloudTrail log deletion, SCPs can also include denies on cloudtrail:DeleteTrail and s3:DeleteObject actions targeting the CloudTrail log S3 bucket. These SCPs ensure that no user, including administrators, can violate the compliance requirements.

AWS documentation under the Security and Compliance domain for CloudOps states:

"Use AWS Control Tower to establish a secure, compliant, multi-account environment with preventive guardrails through service control policies and detective controls through AWS Config."

This approach meets all stated needs: centralized enforcement, automatic propagation to new accounts, region-based restrictions, and immutable audit logs. Options A, B, and D either detect violations reactively or lack complete enforcement and automation across future accounts.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 4: Security and Compliance
- AWS Control Tower - Preventive and Detective Guardrails
- AWS Organizations - Service Control Policies (SCPs)
- AWS Well-Architected Framework - Security Pillar (Governance and Centralized Controls)

Question: 6

A company's AWS accounts are in an organization in AWS Organizations. The organization has all features enabled. The accounts use Amazon EC2 instances to host applications. The company manages the EC2 instances manually by using the AWS Management Console. The company applies updates to the EC2 instances by using an SSH connection to each EC2 instance.

The company needs a solution that uses AWS Systems Manager to manage all the organization's current and future EC2 instances. The latest version of Systems Manager Agent (SSM Agent) is running on the EC2 instances.

Which solution will meet these requirements?

A. Configure a home AWS Region in Systems Manager Quick Setup in the organization's management account. Deploy the Systems Manager Default Host Management Configuration Quick Setup from the management account.

B. Configure a home AWS Region in Systems Manager Quick Setup in the organization's management account. Create a Systems Manager Run Command that attaches the AmazonSSMServiceRolePolicy IAM policy to every IAM role that the EC2 instances use. Invoke the command in every account in the organization.

C. Create an AWS CloudFormation stack set that contains a Systems Manager parameter to define the Default Host Management Configuration role. Use the organization's management account to deploy the stack set to every account in the organization.

D. Create an AWS CloudFormation stack set that contains an EC2 instance profile with the AmazonSSMManagedEC2InstanceDefaultPolicy IAM policy attached. Use the organization's management account to deploy the stack set to every account in the organization.

_____ Answer: A

Explanation:

AWS CloudOps automation best practices recommend using AWS Systems Manager Quick Setup for organization-wide management and configuration of EC2 instances. The Default Host Management Configuration Quick Setup automatically enables Systems Manager capabilities such as Patch Manager, Inventory, Session Manager, and Automation across all managed instances within the organization.

When deployed from the management account, Quick Setup automatically integrates with AWS Organizations to propagate configuration and permissions to existing and future accounts. This meets the requirement for organization-wide management with no manual configuration or SSH access. AWS documentation notes:

"You can use Quick Setup in the management account of an organization in AWS Organizations to configure Systems Manager capabilities for all accounts and Regions. Quick Setup automatically keeps configurations up to date."

Options B, C, and D require custom deployments or manual IAM updates, lacking centralized automation. Therefore, Option A fully satisfies CloudOps standards for automated provisioning and ongoing management of EC2 instances across an organization.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 3: Deployment,

Provisioning and Automation

- AWS Systems Manager - Quick Setup and Default Host Management Configuration
- AWS Organizations Integration with Systems Manager
- AWS Well-Architected Framework - Operational Excellence Pillar

Question: 7

A CloudOps engineer creates an AWS CloudFormation template to define an application stack that can be deployed in multiple AWS Regions. The CloudOps engineer also creates an Amazon CloudWatch dashboard by using the AWS Management Console. Each deployment of the application requires its own CloudWatch dashboard.

How can the CloudOps engineer automate the creation of the CloudWatch dashboard each time the application is deployed?

A. Create a script by using the AWS CLI to run the aws cloudformation put-dashboard command with the name of the dashboard. Run the command each time a new CloudFormation stack is created.

B. Export the existing CloudWatch dashboard as JSON. Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Include the exported JSON in the resource's DashboardBody property.

C. Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Use the intrinsic Ref function to reference the ID of the existing CloudWatch dashboard.

D. Update the CloudFormation template to define an AWS::CloudWatch::Dashboard resource. Specify the name of the existing dashboard in the DashboardName property.

Answer: B

Explanation:

According to CloudOps automation and monitoring best practices, CloudWatch dashboards should be provisioned as infrastructure-as-code (IaC) resources using AWS CloudFormation to ensure consistency, repeatability, and version control. AWS CloudFormation supports the AWS::CloudWatch::Dashboard resource, where the DashboardBody property accepts a JSON object describing widgets, metrics, and layout.

By exporting the existing dashboard configuration as JSON and embedding it into the CloudFormation template, every deployment of the application automatically creates its corresponding dashboard. This method aligns with the CloudOps requirement for automated deployment and operational visibility within the same stack lifecycle.

AWS documentation explicitly states:

"Use the AWS::CloudWatch::Dashboard resource to create a dashboard from your template. You can include the same JSON you use to define a dashboard in the console."

Option A requires manual execution. Options C and D incorrectly reference or reuse existing dashboards, failing to produce unique, deployment-specific dashboards.

References (AWS CloudOps Documents / Study Guide):

• AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 1: Monitoring and Logging

• AWS CloudFormation User Guide - Resource Type: AWS::CloudWatch::Dashboard

• AWS Well-Architected Framework - Operational Excellence Pillar

• Amazon CloudWatch - Automating Dashboards with Infrastructure as Code

Question: 8

A CloudOps engineer needs to ensure that AWS resources across multiple AWS accounts are tagged consistently. The company uses an organization in AWS Organizations to centrally manage the accounts. The company wants to implement cost allocation tags to accurately track the costs that are allocated to each business unit.

Which solution will meet these requirements with the LEAST operational overhead?

A. Use Organizations tag policies to enforce mandatory tagging on all resources. Enable cost allocation tags in the AWS Billing and Cost Management console.

B. Configure AWS CloudTrail events to invoke an AWS Lambda function to detect untagged resources and to automatically assign tags based on predefined rules.

C. Use AWS Config to evaluate tagging compliance. Use AWS Budgets to apply tags for cost allocation.

D. Use AWS Service Catalog to provision only pre-tagged resources. Use AWS Trusted Advisor to enforce tagging across the organization.

Answer: A

Explanation:

Tagging is essential for governance, cost management, and automation in CloudOps operations. The AWS Organizations tag policies feature allows centralized definition and enforcement of required tag keys and

accepted values across all accounts in an organization. According to the AWS CloudOps study guide under Deployment, Provisioning, and Automation, tag policies enable automatic validation of tags, ensuring consistency with minimal manual overhead.

Once tagging consistency is enforced, enabling cost allocation tags in the AWS Billing and Cost Management console allows accurate cost distribution per business unit. AWS documentation states:

"Use AWS Organizations tag policies to standardize tags across accounts. You can activate cost allocation tags in the Billing console to track and allocate costs."

Option B introduces unnecessary complexity with Lambda automation. Option C detects but does not enforce tagging. Option D limits flexibility to Service Catalog resources only. Therefore, Option A provides a centrally managed, automated, and low-overhead solution that meets CloudOps tagging and cost-tracking requirements.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 3: Deployment, Provisioning and Automation
- AWS Organizations - Tag Policies
- AWS Billing and Cost Management - Cost Allocation Tags
- AWS Well-Architected Framework - Operational Excellence and Cost Optimization Pillars

Question: 9

A user working in the Amazon EC2 console increased the size of an Amazon Elastic Block Store (Amazon EBS) volume attached to an Amazon EC2 Windows instance. The change is not reflected in the file system.

What should a CloudOps engineer do to resolve this issue?

A. Extend the file system with operating system-level tools to use the new storage capacity.

B. Reattach the EBS volume to the EC2 instance.

C. Reboot the EC2 instance that is attached to the EBS volume.

D. Take a snapshot of the EBS volume. Replace the original volume with a volume that is created from the snapshot.

Answer: A

Explanation:

When an Amazon EBS volume is resized, the new storage capacity is immediately available to the attached EC2 instance. However, EBS does not automatically extend the file system. The CloudOps engineer must manually extend the file system within the operating system to utilize the additional space.

AWS documentation for EC2 and EBS specifies:

"After you increase the size of an EBS volume, use file system-specific tools to extend the file system so that the operating system can use the new storage capacity."

On Windows instances, this can be achieved through Disk Management or diskpart commands. On Linux systems, utilities such as growpart and resize2fs are used.

Options B and C do not modify file system metadata and are ineffective. Option D unnecessarily replaces the volume, which adds risk and downtime. Thus, Option A aligns with the Monitoring and Performance Optimization practices of AWS CloudOps by properly extending the file system to recognize the new capacity.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 1
- Amazon EBS - Modifying EBS Volumes
- Amazon EC2 User Guide - Extending a File System After Resizing a Volume
- AWS Well-Architected Framework - Performance Efficiency Pillar

Question: 10

Optimization]

A company has a workload that is sending log data to Amazon CloudWatch Logs. One of the fields includes a measure of application latency. A CloudOps engineer needs to monitor the p90 statistic of this field over time.

What should the CloudOps engineer do to meet this requirement?

A. Create an Amazon CloudWatch Contributor Insights rule on the log data.

B. Create a metric filter on the log data.

C. Create a subscription filter on the log data.

D. Create an Amazon CloudWatch Application Insights rule for the workload.

Answer: B

Explanation:

To analyze and visualize custom statistics such as the p90 latency (90th percentile), a CloudWatch metric must be generated from the log data. The correct method is to create a metric filter that extracts the latency value from each log event and publishes it as a CloudWatch metric. Once the metric is published, percentile statistics (p90, p95, etc.) can be displayed in CloudWatch dashboards or alarms.

AWS documentation states:

"You can use metric filters to extract numerical fields from log events and publish them as metrics in CloudWatch. CloudWatch supports percentile statistics such as p90 and p95 for these metrics."

Contributor Insights (Option A) is for analyzing frequent contributors, not numeric distributions. Subscription filters (Option C) are used for log streaming, and Application Insights (Option D) provides monitoring of application health but not custom p90 statistics. Hence, Option B is the CloudOps-aligned, minimal-overhead solution for percentile latency monitoring.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 1: Monitoring and Logging
- Amazon CloudWatch Logs - Metric Filters
- AWS Well-Architected Framework - Operational Excellence Pillar

Question: 11

A company is running an application on premises and wants to use AWS for data backup. All of the data must be available locally. The backup application can write only to block-based storage that is compatible with the Portable Operating System Interface (POSIX).

Which backup solution will meet these requirements?

A. Configure the backup software to use Amazon S3 as the target for the data backups.

B. Configure the backup software to use Amazon S3 Glacier Flexible Retrieval as the target for the data backups.

C. Use AWS Storage Gateway, and configure it to use gateway-cached volumes.

D. Use AWS Storage Gateway, and configure it to use gateway-stored volumes.

Answer: D

Explanation:

The Storage Gateway service enables hybrid cloud backup by presenting local block storage that synchronizes with AWS cloud storage. For scenarios where all data must remain available locally while still backed up to

AWS, the correct mode is gateway-stored volumes.

AWS documentation defines:

"Use stored volumes if you want to keep all your data locally while asynchronously backing up point-in-time snapshots to Amazon S3 for durable storage."

These volumes expose an iSCSI interface compatible with POSIX file systems, allowing direct use by on-premises backup software.

Gateway-cached volumes (Option C) store primary data in AWS with limited local cache, violating the "all data must be available locally" requirement. Options A and B are object-based storage solutions, not compatible with POSIX or block-based backup applications.

Therefore, Option D fully satisfies CloudOps reliability and continuity best practices by ensuring local availability, cloud durability, and POSIX compatibility for backups.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 2: Reliability and Business Continuity
- AWS Storage Gateway User Guide - Stored Volumes Overview
- AWS Well-Architected Framework - Reliability Pillar
- AWS Hybrid Cloud Storage Best Practices

Question: 12

A CloudOps engineer needs to control access to groups of Amazon EC2 instances using AWS Systems Manager Session Manager. Specific tags on the EC2 instances have already been added.

Which additional actions should the CloudOps engineer take to control access? (Select TWO.)

A. Attach an IAM policy to the users or groups that require access to the EC2 instances.

B. Attach an IAM role to control access to the EC2 instances.

C. Create a placement group for the EC2 instances and add a specific tag.

D. Create a service account and attach it to the EC2 instances that need to be controlled.

E. Create an IAM policy that grants access to any EC2 instances with a tag specified in the Condition element.

Answer: A, E

Explanation:

AWS Systems Manager Session Manager allows secure, auditable instance access without SSH keys or inbound ports. To control access based on instance tags, CloudOps best practices require two configurations:

Attach an IAM policy to users or groups granting ssm:StartSession, ssm:DescribeInstanceInformation, and ssm:DescribeSessions.

Include a Condition element in the IAM policy referencing instance tags, such as Condition: {"StringEquals": {"ssm:resourceTag/Environment": "Production"}}.

This ensures users can start sessions only with instances that have matching tags, providing fine-grained access control.

AWS CloudOps documentation under Security and Compliance states:

"Use IAM policies with resource tags in the Condition element to restrict which managed instances users can access using Session Manager."

Options B and D incorrectly suggest attaching roles or service accounts that are not relevant to user-level access control. Option C (placement groups) pertains to networking and performance, not access management. Therefore, A and E together provide tag-based, least-privilege access as required.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 4: Security and Compliance
- AWS Systems Manager User Guide - Controlling Access to Session Manager Using Tags
- AWS IAM Policy Reference - Condition Keys for AWS Systems Manager
- AWS Well-Architected Framework - Security Pillar

Question: 13 _____

A global gaming company is preparing to launch a new game on AWS. The game runs in multiple AWS Regions on a fleet of Amazon EC2 instances. The instances are in an Auto Scaling group behind an Application Load Balancer (ALB) in each Region. The company plans to use Amazon Route 53 for DNS services. The DNS configuration must direct users to the Region that is closest to them and must provide automated failover.

Which combination of steps should a CloudOps engineer take to configure Route 53 to meet these requirements? (Select TWO.)

A. Create Amazon CloudWatch alarms that monitor the health of the ALB in each Region. Configure Route 53 DNS failover by using a health check that monitors the alarms.

B. Create Amazon CloudWatch alarms that monitor the health of the EC2 instances in each Region. Configure Route 53 DNS failover by using a health check that monitors the alarms.

C. Configure Route 53 DNS failover by using a health check that monitors the private IP address of an EC2 instance in each Region.

D. Configure Route 53 geoproximity routing. Specify the Regions that are used for the infrastructure.

E. Configure Route 53 simple routing. Specify the continent, country, and state or province that are used for the infrastructure.

Answer: A, D

Explanation:

The combination of geoproximity routing and DNS failover health checks provides global low-latency routing with high availability.

Geoproximity routing in Route 53 routes users to the AWS Region closest to their geographic location, optimizing latency. For automatic failover, Route 53 health checks can monitor CloudWatch alarms tied to the health of the ALB in each Region. When a Region becomes unhealthy, Route 53 reroutes traffic to the next available Region automatically.

AWS documentation states:

"Use geoproximity routing to direct users to resources based on geographic location, and configure health checks to provide DNS failover for high availability."

Option B incorrectly monitors EC2 instances directly, which is not efficient at scale. Option C uses private IPs, which cannot be globally health-checked. Option E (simple routing) does not support geographic or failover routing. Hence, A and D together meet both the proximity and failover requirements.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 5: Networking and Content Delivery
- Amazon Route 53 Developer Guide - Geoproximity Routing and DNS Failover
- AWS Well-Architected Framework - Reliability Pillar
- Amazon CloudWatch Alarms - Integration with Route 53 Health Checks

Question: 14

A company requires the rotation of administrative credentials for production workloads on a regular basis. A CloudOps engineer must implement this policy for an Amazon RDS DB instance's master user password.

Which solution will meet this requirement with the LEAST operational effort?

A. Create an AWS Lambda function to change the RDS master user password. Create an Amazon EventBridge scheduled rule to invoke the Lambda function.

B. Create a new SecureString parameter in AWS Systems Manager Parameter Store. Encrypt the parameter with an AWS Key Management Service (AWS KMS) key. Configure automatic rotation.

C. Create a new String parameter in AWS Systems Manager Parameter Store. Configure automatic rotation.

D. Create a new RDS database secret in AWS Secrets Manager. Apply the secret to the RDS DB instance. Configure automatic rotation.

Answer: D

Explanation:

AWS Secrets Manager natively supports credential management and automatic rotation for Amazon RDS master user passwords. When a secret is associated with an RDS instance, Secrets Manager automatically updates the password both in the secret and on the database, without downtime or manual scripting.

AWS documentation confirms:

"AWS Secrets Manager can automatically rotate the master user password for Amazon RDS databases. Rotation is fully managed and integrated, requiring no custom code or maintenance."

Option A introduces unnecessary Lambda automation. Option B and C use Parameter Store, which does not provide direct RDS password rotation. Therefore, Option D achieves secure, automatic credential rotation with least operational effort, fully aligned with CloudOps security automation principles.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 4: Security and Compliance
- AWS Secrets Manager - Rotating Secrets for Amazon RDS
- AWS Well-Architected Framework - Security Pillar
- Amazon RDS User Guide - Managing Master User Passwords

Question: 15

A company has a microservice that runs on a set of Amazon EC2 instances. The EC2 instances run behind an Application Load Balancer (ALB).

A CloudOps engineer must use Amazon Route 53 to create a record that maps the ALB URL to example.com.

Which type of record will meet this requirement?

A. An A record

B. An AAAA record

C. An alias record

D. A CNAME record

Answer: C

Explanation:

An alias record is the recommended Route 53 record type to map domain names (e.g., example.com) to AWS-managed resources such as an Application Load Balancer. Alias records are extension types of A or AAAA records that support AWS resources directly, providing automatic DNS integration and no additional query costs.

AWS documentation states:

"Use alias records to map your domain or subdomain to an AWS resource such as an Application Load Balancer, CloudFront distribution, or S3 website endpoint."

A and AAAA records are used for static IP addresses, not load balancers. CNAME records cannot be used at the root domain (e.g., example.com). Thus, Option C is correct as it meets CloudOps networking best practices for scalable, managed DNS resolution to ALBs.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Domain 5: Networking and Content Delivery
- Amazon Route 53 Developer Guide - Alias Records
- AWS Well-Architected Framework - Reliability and Performance Efficiency Pillars
- Elastic Load Balancing - Integrating with Route 53

Question: 16

Application A runs on Amazon EC2 instances behind a Network Load Balancer (NLB). The EC2 instances are in an Auto Scaling group and are in the same subnet that is associated with the NLB. Other applications from an on-premises environment cannot communicate with Application A on port 8080.

To troubleshoot the issue, a CloudOps engineer analyzes the flow logs. The flow logs include the following records:

ACCEPT from 192.168.0.13:59003 172.31.16.139:8080

REJECT from 172.31.16.139:8080 192.168.0.13:59003

What is the reason for the rejected traffic?

A. The security group of the EC2 instances has no Allow rule for the traffic from the NLB.

B. The security group of the NLB has no Allow rule for the traffic from the on-premises environment.

C. The ACL of the on-premises environment does not allow traffic to the AWS environment.

D. The network ACL that is associated with the subnet does not allow outbound traffic for the ephemeral port range.

Answer: D

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

VPC Flow Logs show the request arriving and being ACCEPTed on dstport 8080 and the corresponding response being REJECTed on the return path to the client's ephemeral port (59003). AWS networking guidance states that security groups are stateful (return traffic is automatically allowed) while network ACLs are stateless and require explicit inbound and outbound rules for both directions. CloudOps operational guidance for VPC networking further notes that when you allow an inbound request (for example, TCP 8080) through a subnet's network ACL, you must also allow the outbound ephemeral port range (typically 1024-65535) for the response traffic; otherwise, the return packets are dropped and appear as REJECT in flow logs. The observed pattern—request accepted to 8080, response rejected to 59003—matches a missing outbound ephemeral-range allow on the subnet's NACL. Therefore, the cause is the subnet NACL, not security groups or on-premises ACLs. The remediation is to add an outbound ALLOW rule on the NACL for the appropriate ephemeral TCP port range back to the on-premises CIDR (and the corresponding inbound rule if asymmetric).

References (AWS CloudOps documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Networking and Content

Delivery

- Amazon VPC - Network ACLs (stateless behavior and rule requirements)
- Amazon VPC - Security Groups (stateful return traffic)
- VPC Flow Logs - Record fields, ACCEPT/REJECT analysis

Question: 17

A company runs a website on Amazon EC2 instances. Users can upload images to an Amazon S3 bucket and publish the images to the website. The company wants to deploy a serverless image- processing application that uses an AWS Lambda function to resize the uploaded images.

The company's development team has created the Lambda function. A CloudOps engineer must implement a

solution to invoke the Lambda function when users upload new images to the S3 bucket.

Which solution will meet this requirement?

A. Configure an Amazon Simple Notification Service (Amazon SNS) topic to invoke the Lambda function when a user uploads a new image to the S3 bucket.

B. Configure an Amazon CloudWatch alarm to invoke the Lambda function when a user uploads a new image to the S3 bucket.

C. Configure S3 Event Notifications to invoke the Lambda function when a user uploads a new image to the S3 bucket.

D. Configure an Amazon Simple Queue Service (Amazon SQS) queue to invoke the Lambda function when a user uploads a new image to the S3 bucket.

Answer: C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

Use Amazon S3 Event Notifications with AWS Lambda to trigger image processing on object creation. S3 natively supports invoking Lambda for events such as s3:ObjectCreated:*, providing a serverless, low-latency pipeline without managing additional services. AWS operational guidance states that "Amazon S3 can directly invoke a Lambda function in response to object-created events," allowing you to pass event metadata (bucket/key) to the function for resizing and writing results back to S3. This approach minimizes operational overhead, scales automatically with upload volume, and integrates with standard retry semantics. SNS or SQS can be added for fan-out or buffering patterns, but they are not required when the requirement is simply "invoke the Lambda function on upload." CloudWatch alarms do not detect individual S3 object uploads and cannot directly satisfy per-object triggers. Therefore, configuring S3 Lambda event notifications meets the requirement most directly and aligns with CloudOps best practices for event-driven, serverless automation.

References (AWS CloudOps Documents / Study Guide):

- Using AWS Lambda with Amazon S3 (Lambda Developer Guide)
- Amazon S3 Event Notifications (S3 User Guide)
- AWS Well-Architected - Serverless Applications (Operational Excellence)

Question: 18

A company hosts a production MySQL database on an Amazon Aurora single-node DB cluster. The database is queried heavily for reporting purposes. The DB cluster is experiencing periods of performance degradation because of high CPU utilization and maximum connections errors. A CloudOps engineer needs to improve the stability of the database.

Which solution will meet these requirements?

A. Create an Aurora Replica node. Create an Auto Scaling policy to scale replicas based on CPU utilization. Ensure that all reporting requests use the read-only connection string.

B. Create a second Aurora MySQL single-node DB cluster in a second Availability Zone. Ensure that all reporting requests use the connection string for this additional node.

C. Create an AWS Lambda function that caches reporting requests. Ensure that all reporting requests call the Lambda function.

D. Create a multi-node Amazon ElastiCache cluster. Ensure that all reporting requests use the ElastiCache cluster. Use the database if the data is not in the cache.

_____ Answer: A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

Amazon Aurora supports up to 15 Aurora Replicas that share the same storage volume and provide read scaling and improved availability. Official guidance states that replicas “ofload read traffic from the writer” and that you should direct read-only workloads to the reader endpoint, reducing CPU pressure and connection counts on the primary. Aurora also supports Replica Auto Scaling through Application Auto Scaling policies using metrics such as CPU utilization or connections to add or remove replicas automatically. This design addresses both high CPU and maximum connections by moving reporting traffic to read replicas while keeping a single write primary for OLTP. Option B creates a separate cluster with independent storage, increasing operational overhead and data synchronization complexity. Options C and D introduce application-layer caching changes that may not guarantee data freshness or relieve the write node directly. Therefore, adding read replicas and routing reporting to the reader endpoint, with auto scaling based on load, is the least intrusive, CloudOps-aligned way to stabilize performance.

References (AWS CloudOps Documents / Study Guide):

- Amazon Aurora - Replicas and Reader Endpoint (Aurora User Guide)
- Aurora Replica Auto Scaling (Aurora & Application Auto Scaling Guides)
- AWS Well-Architected Framework - Reliability & Performance Efficiency

Question: 19

A CloudOps engineer configures an application to run on Amazon EC2 instances behind an Application Load Balancer (ALB) in a simple scaling Auto Scaling group with the default settings. The Auto Scaling group is configured to use the RequestCountPerTarget metric for scaling. The CloudOps engineer notices that the RequestCountPerTarget metric exceeded the specified limit twice in 180 seconds.

How will the number of EC2 instances in this Auto Scaling group be affected in this scenario?

A. The Auto Scaling group will launch an additional EC2 instance every time the RequestCountPerTarget metric exceeds the predefined limit.

B. The Auto Scaling group will launch one EC2 instance and will wait for the default cooldown period before launching another instance.

C. The Auto Scaling group will send an alert to the ALB to rebalance the traffic and not add new EC2 instances until the load is normalized.

D. The Auto Scaling group will try to distribute the traffic among all EC2 instances before launching another instance.

Answer: B

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

With simple scaling policies, an Auto Scaling group performs one scaling activity when the alarm condition is met, then observes a default cooldown period (300 seconds) before another scaling activity of the same type can begin. CloudOps guidance explains that cooldown prevents rapid successive scale-outs by allowing time for the newly launched instance(s) to register with the load balancer and impact the metric. Even if the alarm breaches multiple times during the cooldown window, the group waits until the cooldown completes before evaluating and acting again. In this case, although RequestCountPerTarget exceeded the threshold twice within 180 seconds, the group will launch a single instance and then wait for cooldown before any additional scale-out can occur. Options A, C, and D do not reflect the behavior of simple scaling with cooldowns; A describes step/target-tracking-like behavior, and C/D are not Auto Scaling mechanics.

References (AWS CloudOps Documents / Study Guide):

- Amazon EC2 Auto Scaling - Simple Scaling Policies and Cooldown (User Guide)
- Elastic Load Balancing Metrics - ALB RequestCountPerTarget (CloudWatch Metrics)
- AWS Well-Architected Framework - Performance Efficiency & Operational Excellence

Question: 20

A company uses Amazon ElastiCache (Redis OSS) to cache application data. A CloudOps engineer must implement a solution to increase the resilience of the cache. The solution also must minimize the recovery time objective (RTO).

Which solution will meet these requirements?

A. Replace ElastiCache (Redis OSS) with ElastiCache (Memcached).

B. Create an Amazon EventBridge rule to initiate a backup every hour. Restore the backup when necessary.

C. Create a read replica in a second Availability Zone. Enable Multi-AZ for the ElastiCache (Redis OSS) replication group.

D. Enable automatic backups. Restore the backups when necessary.

Answer: C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

For high availability and fast failover, ElastiCache for Redis supports replication groups with Multi-AZ and automatic failover. CloudOps guidance states that a primary node can be paired with one or more replicas across multiple Availability Zones; if the primary fails, Redis automatically promotes a replica to primary in seconds, thereby minimizing RTO. This architecture maintains in-memory data continuity without waiting for backup restore operations. Backups (Options B and D) provide durability but require restore and re-warm procedures that increase RTO and may impact application latency. Switching engines (Option A) to Memcached does not provide Redis replication/failover semantics and would not inherently improve resilience for this use case. Therefore, creating a read replica in a different AZ and enabling Multi-AZ with automatic failover is the prescribed CloudOps pattern to increase resilience and achieve the lowest practical RTO for Redis caches.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Reliability and Business Continuity
- Amazon ElastiCache for Redis - Replication Groups, Multi-AZ, and Automatic Failover
- AWS Well-Architected Framework - Reliability Pillar

Question: 21

An AWS CloudFormation template creates an Amazon RDS instance. This template is used to build up development environments as needed and then delete the stack when the environment is no longer required. The RDS-persisted data must be retained for further use, even after the CloudFormation stack is deleted.

How can this be achieved in a reliable and efficient way?

A. Write a script to continue backing up the RDS instance every five minutes.

B. Create an AWS Lambda function to take a snapshot of the RDS instance, and manually invoke the function before deleting the stack.

C. Use the Snapshot Deletion Policy in the CloudFormation template definition of the RDS instance.

D. Create a new CloudFormation template to perform backups of the RDS instance, and run this template

before deleting the stack.

Answer: C

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

AWS CloudFormation supports the DeletionPolicy attribute to control what happens to a resource when a stack is deleted. For Amazon RDS DB instances, setting DeletionPolicy: Snapshot instructs CloudFormation to retain a final DB snapshot automatically at stack deletion. CloudOps best practice recommends using this native mechanism for data retention and auditability, avoiding manual scripts or out-of-band processes. Options A, B, and D introduce operational overhead and potential human error. With DeletionPolicy set to Snapshot, the environment can be repeatedly created and torn down while preserving data states for later restoration with minimal manual steps. This aligns with IaC principles—declarative, repeatable, and reliable—and supports efficient lifecycle management of ephemeral development stacks.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Deployment, Provisioning and Automation
- AWS CloudFormation User Guide - DeletionPolicy Attribute (Snapshot for RDS)
- AWS Well-Architected Framework - Operational Excellence Pillar

Question: 22

A company has a VPC that contains a public subnet and a private subnet. The company deploys an Amazon EC2 instance that uses an Amazon Linux Amazon Machine Image (AMI) and has the AWS Systems Manager Agent (SSM Agent) installed in the private subnet. The EC2 instance is in a security group that allows only outbound traffic.

A CloudOps engineer needs to give a group of privileged administrators the ability to connect to the instance through SSH without exposing the instance to the internet.

Which solution will meet this requirement?

A. Create an EC2 Instance Connect endpoint in the private subnet. Update the security group to allow inbound SSH traffic. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.

B. Create a Systems Manager endpoint in the private subnet. Update the security group to allow SSH traffic from the private network where the Systems Manager endpoint is connected. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.

C. Create an EC2 Instance Connect endpoint in the public subnet. Update the security group to allow SSH

traffic from the private network. Create an IAM group for privileged administrators. Assign the PowerUserAccess managed policy to the IAM group.

D. Create a Systems Manager endpoint in the public subnet. Create an IAM role that has the AmazonSSMManagedInstanceCore permission for the EC2 instance. Create an IAM group for privileged administrators. Assign the AmazonEC2ReadOnlyAccess IAM policy to the IAM group.

Answer: A

Explanation:

Comprehensive and Detailed Explanation From Exact Extract of AWS CloudOps Doocuments:

EC2 Instance Connect Endpoint (EIC Endpoint) enables SSH to instances in private subnets without public IPs and without needing to traverse the public internet. CloudOps guidance explains that you deploy the endpoint in the same VPC/subnet as the targets, then allow inbound SSH on the instance security group from the endpoint's security group. Access is governed by IAM—administrators must have Instance Connect permissions; while the example uses a broad policy, the key mechanism is EIC in the private subnet plus SG rules scoped to the endpoint. Systems Manager Session Manager can provide shell access without SSH, but the requirement explicitly states "connect through SSH," making EIC the purpose-built solution. Options B and D misuse Systems Manager for SSH and propose unnecessary SG changes or incorrect endpoint placement; Option C places the endpoint in a public subnet, which is not required for private SSH access. Therefore, creating an EC2 Instance Connect endpoint in the private subnet and updating SGs accordingly meets the requirement while keeping the instance non-internet-exposed.

References (AWS CloudOps Documents / Study Guide):

- AWS Certified CloudOps Engineer - Associate (SOA-C03) Exam Guide - Security and Compliance
- Amazon EC2 - Instance Connect Endpoint (Private SSH Access)
- AWS Well-Architected Framework - Security Pillar (Least Privilege Network Access)

Question: 23

A global company runs a critical primary workload in the us-east-1 Region. The company wants to ensure business continuity with minimal downtime in case of a workload failure. The company wants to replicate the workload to a second AWS Region.

A CloudOps engineer needs a solution that achieves a recovery time objective (RTO) of less than 10 minutes and a zero recovery point objective (RPO) to meet service level agreements.

Which solution will meet these requirements?

A. Implement a pilot light architecture that provides real-time data replication in the second Region. Configure Amazon Route 53 health checks and automated DNS failover.

B. Implement a warm standby architecture that provides regular data replication in a second Region.

Configure Amazon Route 53 health checks and automated DNS failover.

C. Implement an active-active architecture that provides real-time data replication across two Regions. Use Amazon Route 53 health checks and a weighted routing policy.

D. Implement a custom script to generate a regular backup of the data and store it in an S3 bucket that is in a second Region. Use the backup to launch the application in the second Region in the event of a workload failure.

_____ Answer: C

Explanation:

According to the AWS Cloud Operations and Disaster Recovery documentation, the active-active multi-Region architecture provides the lowest possible RTO and RPO among all disaster recovery strategies. In this approach, workloads are deployed and actively running in multiple AWS Regions simultaneously. All data is continuously replicated in real time between Regions using fully managed replication services, ensuring zero data loss (zero RPO).

Because both Regions are active and capable of handling requests, failover between them is instantaneous, meeting the RTO of less than 10 minutes. Amazon Route 53 is used with weighted or latency-based routing policies and health checks to automatically route traffic away from an impaired Region to the healthy Region without manual intervention.

In contrast:

Pilot Light Architecture maintains only a minimal copy of the environment in the secondary Region. It requires time to scale up infrastructure during a disaster, resulting in longer RTO and potential data loss (non-zero RPO).

Warm Standby Architecture keeps partially running infrastructure in the secondary Region. Although faster than pilot light, it still requires scaling and synchronization, resulting in higher RTO and RPO compared to active-active.

Backup and Restore (option D) relies on periodic backups and restores data when needed. This approach has the highest RTO and RPO, unsuitable for mission-critical workloads demanding high availability and zero data loss.

Therefore, based on AWS-recommended disaster recovery strategies outlined in the AWS Cloud Operations and Disaster Recovery Guide, the Active-Active Multi-Region architecture (Option C) is the only approach that guarantees RTO <10 minutes and RPO = 0, achieving continuous availability and business continuity across Regions.

Reference: AWS Cloud Operations and Disaster Recovery Whitepaper - Section: Disaster Recovery Strategies - Multi-Site (Active-Active) Approach; AWS CloudOps Best Practices for Reliability and Business Continuity.

Question: 24

Optimization]

A CloudOps engineer is using AWS Compute Optimizer to generate recommendations for a fleet of Amazon EC2 instances. Some of the instances use newly released instance types, while other instances use older instance types.

After the analysis is complete, the CloudOps engineer notices that some of the EC2 instances are missing from the Compute Optimizer dashboard.

What is the likely cause of this issue?

A. The missing instances have insufficient historical Amazon CloudWatch metric data for analysis.

B. Compute Optimizer does not support the instance types of the missing instances.

C. Compute Optimizer already considers the missing instances to be optimized.

D. The missing instances are running a Windows operating system.

Answer: B

Explanation:

According to the AWS Cloud Operations and Compute Optimizer documentation, Compute Optimizer provides right-sizing recommendations by analyzing Amazon CloudWatch metrics and instance configuration data. However, AWS explicitly notes that only supported instance types are included in Compute Optimizer analyses. If an EC2 instance type is newly released or not yet supported by Compute Optimizer, it will not appear in the Compute Optimizer dashboard until official support is added.

The documentation explains that “Compute Optimizer analyses only supported resource types and instance families. Instances using unsupported or newly launched instance types will not appear in the Compute Optimizer console.” This ensures the service provides accurate recommendations based on sufficient performance history and benchmark data.

While CloudWatch metrics are required for analysis, the complete absence of instances from the dashboard — rather than “insufficient metric data” notifications — indicates unsupported instance types. Compute Optimizer would normally still display those with limited metrics but would flag them as “insufficient data,” not remove them entirely.

Therefore, the most accurate cause of missing instances in this case is that Compute Optimizer does not support the newly released instance types, making option B correct.

Reference: AWS Cloud Operations & Compute Optimizer Guide - Section: Supported Resources and Limitations in Compute Optimizer

Question: 25

A company uses an organization in AWS Organizations to manage multiple AWS accounts. The company needs to send specific events from all the accounts in the organization to a new receiver account, where an AWS Lambda function will process the events.

A CloudOps engineer configures Amazon EventBridge to route events to a target event bus in the us- west-2 Region in the receiver account. The CloudOps engineer creates rules in both the sender and receiver accounts that match the specified events. The rules do not specify an account parameter in the event pattern. IAM roles are created in the sender accounts to allow PutEvents actions on the target event bus.

However, the first test events from the us-east-1 Region are not processed by the Lambda function in the receiving account.

What is the likely reason the events are not processed?

A. Interface VPC endpoints for EventBridge are required in the sender accounts and receiver accounts.

B. The target Lambda function is in a different AWS Region, which is not supported by EventBridge.

C. The resource-based policy on the target event bus must be modified to allow PutEvents API calls from the sender accounts.

D. The rule in the receiving account must specify {"account": ["sender-account-id"]} in its event pattern and must include the receiving account ID.

Answer: C

Explanation:

Per the AWS Cloud Operations and EventBridge documentation, when events are sent across AWS accounts — particularly from multiple accounts in an AWS Organization — the target event bus in the receiver account must include a resource-based policy that explicitly allows events:PutEvents API calls from the sender accounts or the organization ID.

Even if the sender accounts have IAM permissions to call PutEvents, the receiving event bus must trust those accounts via a resource policy. Without this configuration, EventBridge automatically rejects incoming cross-account events, and those events never reach the target Lambda function for processing.

AWS guidance states that “Cross-account event delivery requires a resource-based policy on the event bus that grants permissions to the source accounts or organization.” The policy can include either individual AWS account IDs or the organization's root ID.

In this scenario, because the events originate from multiple accounts and there is no resource policy on the target event bus to authorize those sender accounts, the events are not delivered.

Therefore, the correct cause is C - the resource-based policy on the target event bus must be modified to

allow PutEvents API calls from the sender accounts.

Reference: AWS Cloud Operations - EventBridge Cross-Account Event Delivery Section, Permissions for Event Bus Targets and Organizational Event Routing

Question: 26

A CloudOps engineer needs to track the costs of data transfer between AWS Regions. The CloudOps engineer must implement a solution to send alerts to an email distribution list when transfer costs reach 75% of a specific threshold.

What should the CloudOps engineer do to meet these requirements?

A. Create an AWS Cost and Usage Report. Analyze the results in Amazon Athena. Configure an alarm to publish a message to an Amazon Simple Notification Service (Amazon SNS) topic when costs reach 75% of the threshold. Subscribe the email distribution list to the topic.

B. Create an Amazon CloudWatch billing alarm to detect when costs reach 75% of the threshold. Configure the alarm to publish a message to an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the email distribution list to the topic.

C. Use AWS Budgets to create a cost budget for data transfer costs. Set an alert at 75% of the budgeted amount. Configure the budget to send a notification to the email distribution list when costs reach 75% of the threshold.

D. Set up a VPC flow log. Set up a subscription filter to an AWS Lambda function to analyze data transfer. Configure the Lambda function to send a notification to the email distribution list when costs reach 75% of the threshold.

_____ Answer: C

Explanation:

According to the AWS Cloud Operations and Cost Management documentation, AWS Budgets is the recommended service to track and alert on cost thresholds across all AWS accounts and resources. It allows

users to define cost, usage, or reservation budgets, and configure notifications to trigger when usage or cost reaches defined percentages of the budgeted value (e.g., 75%, 90%, 100%).

The AWS Budgets system integrates natively with Amazon Simple Notification Service (SNS) to deliver alerts to an email distribution list or SNS topic subscribers. AWS Budgets supports granular cost filters, including specific service categories such as data transfer, regions, or linked accounts, ensuring precise visibility into inter-Region transfer costs.

By contrast, CloudWatch billing alarms (Option B) monitor total account charges only and do not allow detailed service-level filtering, such as data transfer between Regions. Cost and Usage Reports (Option A) are for detailed cost analysis, not real-time alerting, and VPC Flow Logs (Option D) capture traffic data, not billing or cost-based metrics.

Thus, using AWS Budgets with a 75% alert threshold best satisfies the operational and notification requirements.

Reference: AWS CloudOps and Cost Management Guide - Section: AWS Budgets for Cost Monitoring and Alerts

Question: 27

A CloudOps engineer needs to set up alerting and remediation for a web application. The application consists of Amazon EC2 instances that have AWS Systems Manager Agent (SSM Agent) installed. Each EC2 instance runs a custom web server. The EC2 instances run behind a load balancer and write logs locally.

The CloudOps engineer must implement a solution that restarts the web server software automatically if specific web errors are detected in the logs.

Which combination of steps will meet these requirements? (Select THREE.)

A. Install the Amazon CloudWatch agent on the EC2 instances.

B. Create an AWS CloudTrail metric filter for the web logs. Configure an alarm for the specific errors.

C. Create an Amazon CloudWatch metric filter for the web logs. Configure an alarm for the specific errors.

D. Publish alarm findings to Amazon Simple Email Service (Amazon SES). Invoke an AWS Lambda function to restart the web server software.

E. Create an Amazon EventBridge rule that responds to the alarm. Configure the rule to invoke an AWS Systems Manager Automation runbook to restart the web server software.

F. Create an Amazon Simple Notification Service (Amazon SNS) notification that responds to the alarm. Configure the notification to invoke an AWS Systems Manager Automation runbook to restart the web server software.

Answer: A, C, E

Explanation:

Per the AWS Cloud Operations, Monitoring, and Automation documentation, the correct workflow for automated operational remediation is:

Amazon CloudWatch Agent is installed on each EC2 instance (Option A) to collect local log data and push it to Amazon CloudWatch Logs.

A CloudWatch Metric Filter (Option C) is then defined to identify specific error strings or patterns within those logs (e.g., “HTTP 5xx” or “Service Unavailable”). When such an event occurs, CloudWatch Alarms are triggered.

Upon alarm activation, Amazon EventBridge rules (Option E) are configured to respond automatically by invoking an AWS Systems Manager Automation runbook, which executes an action to restart the web server process on the affected instance via SSM Agent.

This approach aligns directly with AWS's recommended CloudOps remediation pattern, known as event-driven automation, which ensures minimal downtime and eliminates manual intervention.

Options involving CloudTrail (B) or SES notifications (D) are incorrect because they are unrelated to log-based application monitoring and automated remediation workflows.

Reference: AWS Cloud Operations & Systems Manager Guide - Section: Automated Remediation using CloudWatch, EventBridge, and Systems Manager Automation

Question: 28

A CloudOps engineer is configuring an Amazon CloudFront distribution to use an SSL/TLS certificate. The CloudOps engineer must ensure automatic certificate renewal.

Which combination of steps will meet this requirement? (Select TWO.)

A. Use a certificate issued by AWS Certificate Manager (ACM).

B. Use a certificate issued by a third-party certificate authority (CA).

C. Configure CloudFront to automatically renew the certificate when the certificate expires.

D. Configure email validation for the certificate.

E. Configure DNS validation for the certificate.

Answer: A, E

Explanation:

The AWS Cloud Operations and Security documentation specifies that for Amazon CloudFront, automatic certificate renewal is only supported for certificates issued by AWS Certificate Manager (ACM). When a certificate is managed by ACM and validated through DNS validation, ACM automatically renews the certificate before expiration without requiring manual intervention.

Option A ensures that the certificate is issued and managed by ACM, enabling full integration with CloudFront. Option E (DNS validation) is essential for automation; AWS performs revalidation automatically as long as the DNS validation record remains in place.

By contrast, email validation (Option D) requires manual user confirmation upon renewal, which prevents automatic renewals. Certificates issued by third-party certificate authorities (Option B) are manually managed and must be reimported into ACM after renewal. CloudFront does not have a direct feature (Option C) to renew certificates; it relies on ACM's lifecycle management.

Thus, combining ACM-issued certificates (A) with DNS validation (E) ensures continuous, automated renewal with no downtime or human action required.

Reference: AWS Cloud Operations and Security Best Practices - Section: Using AWS Certificate Manager with CloudFront for Automatic Certificate Renewal

Question: 29

A company has an on-premises DNS solution and wants to resolve DNS records in an Amazon Route 53 private hosted zone for example.com. The company has set up an AWS Direct Connect connection for network connectivity between the on-premises network and the VPC. A CloudOps engineer must ensure that an on-premises server can query records in the example.com domain.

What should the CloudOps engineer do to meet these requirements?

A. Create a Route 53 Resolver inbound endpoint. Attach a security group to the endpoint to allow inbound traffic on TCP/UDP port 53 from the on-premises DNS servers.

B. Create a Route 53 Resolver inbound endpoint. Attach a security group to the endpoint to allow outbound traffic on TCP/UDP port 53 to the on-premises DNS servers.

C. Create a Route 53 Resolver outbound endpoint. Attach a security group to the endpoint to allow inbound traffic on TCP/UDP port 53 from the on-premises DNS servers.

D. Create a Route 53 Resolver outbound endpoint. Attach a security group to the endpoint to allow outbound traffic on TCP/UDP port 53 to the on-premises DNS servers.

Answer: A

Explanation:

According to AWS Cloud Operations and Networking documentation, Route 53 Resolver inbound endpoints allow DNS queries to originate from on-premises DNS servers and resolve private hosted zone records in AWS. The inbound endpoint provides DNS resolver IP addresses within the VPC, which the on-premises DNS servers can forward queries to over AWS Direct Connect or VPN connections.

The inbound endpoint must be associated with a security group that permits inbound traffic on TCP and UDP port 53 from the on-premises DNS server IP addresses. This ensures that DNS requests from the on-premises environment reach the VPC Resolver for resolution of private domains like example.com.

By contrast, outbound endpoints are used for the opposite direction—resolving external (onpremises or

internet) DNS names from within AWS VPCs. Therefore, only an inbound endpoint correctly satisfies the direction of resolution in this scenario.

Reference: AWS Cloud Operations & Route 53 Resolver Guide - Section: Inbound and Outbound Endpoints for Hybrid DNS Resolution

Question: 30

A medical research company uses an Amazon Bedrock powered AI assistant with agents and knowledge bases to provide physicians quick access to medical study protocols. The company needs to generate audit reports that contain user identities, usage data for Bedrock agents, access data for knowledge bases, and interaction parameters.

Which solution will meet these requirements?

A. Use AWS CloudTrail to log API events from generative AI workloads. Store the events in CloudTrail Lake. Use SQL-like queries to generate reports.

B. Use Amazon CloudWatch to capture generative AI application logs. Stream the logs to Amazon OpenSearch Service. Use an OpenSearch dashboard visualization to generate reports.

C. Use Amazon CloudWatch to log API events from generative AI workloads. Send the events to an Amazon S3 bucket. Use Amazon Athena queries to generate reports.

D. Use AWS CloudTrail to capture generative AI application logs. Stream the logs to Amazon Managed Service for Apache Flink. Use SQL queries to generate reports.

Answer: A

Explanation:

As per AWS Cloud Operations, Bedrock, and Governance documentation, AWS CloudTrail is the authoritative service for capturing API activity and audit trails across AWS accounts. For Amazon Bedrock, CloudTrail records all user-initiated API calls, including interactions with agents, knowledge bases, and generative AI model parameters.

Using CloudTrail Lake, organizations can store, query, and analyze CloudTrail events directly without needing to export data. CloudTrail Lake supports SQL-like queries for generating audit and compliance reports, enabling the company to retrieve information such as user identity, API usage, timestamp, model or agent ID, and invocation parameters.

In contrast, CloudWatch focuses on operational metrics and log streaming, not API-level identity data. OpenSearch or Flink would add unnecessary complexity and cost for this use case.

Thus, the AWS-recommended CloudOps best practice is to leverage CloudTrail with CloudTrail Lake to maintain

auditable, queryable API activity for Bedrock workloads, fulfilling governance and compliance requirements.

Reference: AWS Cloud Operations & Governance Guide - Section: Auditing and Governance for Generative AI Workloads Using AWS CloudTrail and CloudTrail Lake

Question: 31

A company needs to enforce tagging requirements for Amazon DynamoDB tables in its AWS accounts. A CloudOps engineer must implement a solution to identify and remediate all DynamoDB tables that do not have the appropriate tags.

Which solution will meet these requirements with the LEAST operational overhead?

A. Create a custom AWS Lambda function to evaluate and remediate all DynamoDB tables. Create an Amazon EventBridge scheduled rule to invoke the Lambda function.

B. Create a custom AWS Lambda function to evaluate and remediate all DynamoDB tables. Create an AWS Config custom rule to invoke the Lambda function.

C. Use the required-tags AWS Config managed rule to evaluate all DynamoDB tables for the appropriate tags. Configure an automatic remediation action that uses an AWS Systems Manager Automation custom runbook.

D. Create an Amazon EventBridge managed rule to evaluate all DynamoDB tables for the appropriate tags. Configure the EventBridge rule to run an AWS Systems Manager Automation custom runbook for remediation.

Answer: C

Explanation:

According to the AWS Cloud Operations, Governance, and Compliance documentation, AWS Config provides managed rules that automatically evaluate resource configurations for compliance. The "required-tags" managed rule allows CloudOps teams to specify mandatory tags (e.g., Environment, Owner, CostCenter) and automatically detect non-compliant resources such as DynamoDB tables.

Furthermore, AWS Config supports automatic remediation through AWS Systems Manager Automation runbooks, enabling correction actions (for example, adding missing tags) without manual intervention. This automation minimizes operational overhead and ensures continuous compliance across multiple accounts.

Using a custom Lambda function (Options A or B) introduces unnecessary management complexity, while EventBridge rules alone (Option D) do not provide resource compliance tracking or historical visibility.

Therefore, Option C provides the most efficient, fully managed, and compliant CloudOps solution.

Reference: AWS Cloud Operations & Governance Guide - Section: Compliance Automation Using AWS Config Managed Rules and Systems Manager Remediation

Question: 32

A company's website runs on an Amazon EC2 Linux instance. The website needs to serve PDF files from an Amazon S3 bucket. All public access to the S3 bucket is blocked at the account level. The company needs to allow website users to download the PDF files.

Which solution will meet these requirements with the LEAST administrative effort?

A. Create an IAM role that has a policy that allows s3:list* and s3:get* permissions. Assign the role to the EC2 instance. Assign a company employee to download requested PDF files to the EC2 instance and deliver the files to website users. Create an AWS Lambda function to periodically delete local files.

B. Create an Amazon CloudFront distribution that uses an origin access control (OAC) that points to the S3 bucket. Apply a bucket policy to the bucket to allow connections from the CloudFront distribution. Assign a company employee to provide a download URL that contains the distribution URL and the object path to users when users request PDF files.

C. Change the S3 bucket permissions to allow public access on the source S3 bucket. Assign a company employee to provide a PDF file URL to users when users request the PDF files.

D. Deploy an EC2 instance that has an IAM instance profile to a public subnet. Use a signed URL from the EC2 instance to provide temporary access to the S3 bucket for website users.

Answer: B

Explanation:

Per the AWS Cloud Operations, Networking, and Security documentation, the best practice for serving private S3 content securely to end users is to use Amazon CloudFront with Origin Access Control (OAC).

OAC enables CloudFront to access S3 buckets privately, even when Block Public Access settings are enabled at the account level. This allows content to be delivered globally and securely without making the S3 bucket public. The bucket policy explicitly allows access only from the CloudFront distribution, ensuring that users can retrieve PDF files only via CloudFront URLs.

This configuration offers:

Automatic scalability through CloudFront caching,

Improved security via private access control,

Minimal administration effort with fully managed services.

Other options require manual handling or make the bucket public, violating AWS security best practices.

Therefore, Option B—using CloudFront with Origin Access Control and a restrictive bucket policy— provides the most secure, efficient, and low-maintenance CloudOps solution.

Reference: AWS Cloud Operations and Content Delivery Guide - Section: Serving Private Content Securely from Amazon S3 via CloudFront Using Origin Access Control

Question: 33

A financial services company stores customer images in an Amazon S3 bucket in the us-east-1 Region. To comply with regulations, the company must ensure that all existing objects are replicated to an S3 bucket in a second AWS Region. If an object replication fails, the company must be able to retry replication for the object.

What solution will meet these requirements?

A. Configure Amazon S3 Cross-Region Replication (CRR). Use Amazon S3 live replication to replicate existing objects.

B. Configure Amazon S3 Cross-Region Replication (CRR). Use S3 Batch Replication to replicate existing objects.

C. Configure Amazon S3 Cross-Region Replication (CRR). Use S3 Replication Time Control (S3 RTC) to replicate existing objects.

D. Use S3 Lifecycle rules to move objects to the destination bucket in a second Region.

Answer: B

Explanation:

Per the AWS Cloud Operations and S3 Data Management documentation, Cross-Region Replication (CRR) automatically replicates new objects between S3 buckets across Regions. However, CRR alone does not retroactively replicate existing objects created before replication configuration. To include such objects, AWS introduced S3 Batch Replication.

S3 Batch Replication scans the source bucket and replicates all existing objects that were not copied previously. Additionally, it can retry failed replication tasks automatically, ensuring regulatory compliance for complete dataset replication.

S3 Replication Time Control (S3 RTC) guarantees predictable replication times for new objects only— it does not cover previously stored data. S3 Lifecycle rules (Option D) move or transition objects between storage classes or buckets, but not in a replication context.

Therefore, the correct solution is to use S3 Cross-Region Replication (CRR) combined with S3 Batch Replication to ensure all current and future data is synchronized across Regions with retry capability.

Reference: AWS Cloud Operations and S3 Guide - Section: Cross-Region Replication and Batch Replication for Existing Objects

Question: 34

A CloudOps engineer has created a VPC that contains a public subnet and a private subnet. Amazon EC2 instances that were launched in the private subnet cannot access the internet. The default network ACL is active on all subnets in the VPC, and all security groups allow outbound traffic.

Which solution will provide the EC2 instances in the private subnet with access to the internet?

A. Create a NAT gateway in the public subnet. Create a route from the private subnet to the NAT gateway.

B. Create a NAT gateway in the public subnet. Create a route from the public subnet to the NAT gateway.

C. Create a NAT gateway in the private subnet. Create a route from the public subnet to the NAT gateway.

D. Create a NAT gateway in the private subnet. Create a route from the private subnet to the NAT gateway.

Answer: A

Explanation:

According to the AWS Cloud Operations and Networking documentation, instances in a private subnet do not have a direct route to the internet gateway and thus require a NAT gateway for outbound internet access.

The correct configuration is to create a NAT gateway in the public subnet, associate an Elastic IP address, and then update the private subnet's route table to send all 0.0.0.0/0 traffic to the NAT gateway. This enables instances in the private subnet to initiate outbound connections while keeping inbound traffic blocked for security.

Placing the NAT gateway inside the private subnet (Options C or D) prevents connectivity because it would not have a route to the internet gateway. Configuring routes from the public subnet to the NAT gateway (Option B) does not serve private subnet traffic.

Hence, Option A follows AWS best practices for enabling secure, managed, outbound -only internet access from private resources.

Reference: AWS Cloud Operations & Networking Guide - Section: Providing Internet Access to Private Subnets Using NAT Gateway

Question: 35

Optimization]

A company's architecture team must receive immediate email notifications whenever new Amazon EC2 instances are launched in the company's main AWS production account.

What should a CloudOps engineer do to meet this requirement?

A. Create a user data script that sends an email message through a smart host connector. Include the architecture team's email address in the user data script as the recipient. Ensure that all new EC2 instances include the user data script as part of a standardized build process.

B. Create an Amazon Simple Notification Service (Amazon SNS) topic and a subscription that uses the email protocol. Enter the architecture team's email address as the subscriber. Create an Amazon EventBridge rule that reacts when EC2 instances are launched. Specify the SNS topic as the rule's target.

C. Create an Amazon Simple Queue Service (Amazon SQS) queue and a subscription that uses the email protocol. Enter the architecture team's email address as the subscriber. Create an Amazon EventBridge rule that reacts when EC2 instances are launched. Specify the SQS queue as the rule's target.

D. Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure AWS Systems Manager to publish EC2 events to the SNS topic. Create an AWS Lambda function to poll the SNS topic. Configure the Lambda function to send any messages to the architecture team's email address.

Answer: B

Explanation:

As per the AWS Cloud Operations and Event Monitoring documentation, the most efficient method for event-driven notification is to use Amazon EventBridge to detect specific EC2 API events and trigger a Simple Notification Service (SNS) alert.

EventBridge continuously monitors AWS service events, including RunInstances, which signals the creation of new EC2 instances. When such an event occurs, EventBridge sends it to an SNS topic, which then immediately emails subscribed recipients — in this case, the architecture team.

This combination provides real-time, serverless notifications with minimal management. SQS (Option C) is designed for queue-based processing, not direct user alerts. User data scripts (Option A) and custom polling with Lambda (Option D) introduce unnecessary operational complexity and latency.

Hence, Option B is the correct and AWS-recommended CloudOps design for immediate launch notifications.

Reference: AWS Cloud Operations & Monitoring Guide - Section: EventBridge and SNS Integration for EC2 Event Notifications

Question: 36

A company runs an application on Amazon EC2 that connects to an Amazon Aurora PostgreSQL database. A developer accidentally drops a table from the database, causing application errors. Two hours later, a CloudOps engineer needs to recover the data and make the application functional again.

Which solution will meet this requirement?

A. Use the Aurora Backtrack feature to rewind the database to a specified time, 2 hours in the past.

B. Perform a point-in-time recovery on the existing database to restore the database to a specified point in time, 2 hours in the past.

C. Perform a point-in-time recovery and create a new database to restore the database to a specified point in time, 2 hours in the past. Reconfigure the application to use a new database endpoint.

D. Create a new Aurora cluster. Choose the Restore data from S3 bucket option. Choose log files up to the failure time 2 hours in the past.

Answer: C

Explanation:

In the AWS Cloud Operations and Aurora documentation, when data loss occurs due to human error such as dropped tables, Point-in-Time Recovery (PITR) is the recommended method for restoration. PITR creates a new Aurora cluster restored to a specific time before the failure.

The restored cluster has a new endpoint that must be reconfigured in the application to resume normal operations. AWS does not support performing PITR directly on an existing production database because that would overwrite current data.

Aurora Backtrack (Option A) applies only to Aurora MySQL, not PostgreSQL. Option B is incorrect because PITR cannot be executed in place. Option D refers to an import process from S3, which is unrelated to time-based recovery.

Hence, Option C is correct and follows the AWS CloudOps standard recovery pattern for PostgreSQL workloads.

Reference: AWS Cloud Operations & Aurora Guide - Section: Performing Point-in-Time Recovery for Aurora PostgreSQL Clusters

Question: 37

A company is using an Amazon Aurora MySQL DB cluster that has point-in-time recovery, backtracking, and automatic backup enabled. A CloudOps engineer needs to roll back the DB cluster to a specific recovery point within the previous 72 hours. Restores must be completed in the same production DB cluster.

Which solution will meet these requirements?

A. Create an Aurora Replica. Promote the replica to replace the primary DB instance.

B. Create an AWS Lambda function to restore an automatic backup to the existing DB cluster.

C. Use backtracking to rewind the existing DB cluster to the desired recovery point.

D. Use point-in-time recovery to restore the existing DB cluster to the desired recovery point.

_____ Answer: C

Explanation:

As documented in AWS Cloud Operations and Database Recovery, Aurora Backtrack allows you to rewind the existing database cluster to a chosen point in time without creating a new cluster. This feature supports fine-grained rollback for accidental data changes, making it ideal for scenarios like table deletions or logical corruption.

Backtracking maintains continuous transaction logs and permits rewinding within a configurable window (up to 72 hours). It does not require creating a new cluster or endpoint, and it preserves the same production environment, fulfilling the operational requirement for in-place recovery.

In contrast, Point-in-Time Recovery (Option D) always creates a new cluster, while replica promotion (Option A) and Lambda restoration (Option B) are unrelated to immediate rollback operations.

Therefore, Option C, using Aurora Backtrack, best meets the requirement for same-cluster restoration and minimal downtime.

Reference: AWS Cloud Operations & Database Management Guide - Section: Using Aurora Backtrack for Fast In-Place Recovery

Question: 38 _____

Optimization]

An application runs on Amazon EC2 instances that are in an Auto Scaling group. A CloudOps engineer needs to implement a solution that provides a central storage location for errors that the application logs to disk. The solution must also provide an alert when the application logs an error.

What should the CloudOps engineer do to meet these requirements?

A. Deploy and configure the Amazon CloudWatch agent on the EC2 instances to log to a CloudWatch log group. Create a metric filter on the target CloudWatch log group. Create a CloudWatch alarm that publishes to an Amazon Simple Notification Service (Amazon SNS) topic that has an email subscription.

B. Create a cron job on the EC2 instances to identify errors and push the errors to an Amazon CloudWatch metric filter. Configure the filter to publish to an Amazon Simple Notification Service (Amazon SNS) topic that has an SMS subscription.

C. Deploy an AWS Lambda function that pushes the errors directly to Amazon CloudWatch Logs. Configure the Lambda function to run every time the log file is updated on disk.

D. Create an Auto Scaling lifecycle hook that invokes an EC2-based script to identify errors. Configure the script to push the error messages to an Amazon CloudWatch log group when the EC2 instances scale in. Create a CloudWatch alarm that publishes to an Amazon Simple Notification Service (Amazon SNS) topic that has an email subscription when the number of error messages exceeds a threshold.

Answer: A

Explanation:

The AWS Cloud Operations and Monitoring documentation specifies that the Amazon CloudWatch Agent is the recommended tool for collecting system and application logs from EC2 instances. The agent pushes these logs into a centralized CloudWatch Logs group, providing durable storage and real-time monitoring.

Once the logs are centralized, a CloudWatch Metric Filter can be configured to search for specific error keywords (for example, "ERROR" or "FAILURE"). This filter transforms matching log entries into custom metrics. From there, a CloudWatch Alarm can monitor the metric threshold and publish notifications to an Amazon SNS topic, which can send email or SMS alerts to subscribed recipients.

This combination provides a fully automated, managed, and serverless solution for log aggregation and error alerting. It eliminates the need for manual cron jobs (Option B), custom scripts (Option D), or Lambda-based log streaming (Option C).

Reference: AWS Cloud Operations & Monitoring Guide - Collecting Application Logs and Creating Alarms Using CloudWatch Agent, Metric Filters, and SNS Notifications

Question: 39

A company's security policy prohibits connecting to Amazon EC2 instances through SSH and RDP. Instead, staff must use AWS Systems Manager Session Manager. Users report they cannot connect to one Ubuntu instance, even though they can connect to others.

What should a CloudOps engineer do to resolve this issue?

A. Add an inbound rule for port 22 in the security group associated with the Ubuntu instance.

B. Assign the AmazonSSMManagedInstanceCore managed policy to the EC2 instance profile for the Ubuntu instance.

C. Configure the SSM Agent to log in with a user name of "ubuntu".

D. Generate a new key pair, configure Session Manager to use this new key pair, and provide the private key to the users.

Answer: B

Explanation:

According to AWS Cloud Operations and Systems Manager documentation, Session Manager requires that each managed instance be associated with an IAM instance profile that grants Systems Manager core permissions. The required permissions are provided by the AmazonSSMManagedInstanceCore AWS-managed policy.

If this policy is missing or misconfigured, the Systems Manager Agent (SSM Agent) cannot communicate with the Systems Manager service, causing connection failures even if the agent is installed and running. This explains why other instances work—those instances likely have the correct IAM role attached.

Enabling port 22 (Option A) violates the company's security policy, while configuring user names (Option C) and key pairs (Option D) are irrelevant because Session Manager operates over secure API channels, not SSH keys.

Therefore, the correct resolution is to attach or update the instance profile with the AmazonSSMManagedInstanceCore policy, restoring Session Manager connectivity.

Reference: AWS Cloud Operations & Systems Manager Guide - Instance Profile Requirements for Session Manager Connectivity

Question: 40

A company deploys an application on Amazon EC2 instances in an Auto Scaling group behind an Application Load Balancer (ALB). The company wants to protect the application from SQL injection attacks.

Which solution will meet this requirement?

A. Deploy AWS Shield Advanced in front of the ALB. Enable SQL injection filtering.

B. Deploy AWS Shield Standard in front of the ALB. Enable SQL injection filtering.

C. Deploy a vulnerability scanner on each EC2 instance. Continuously scan the application code.

D. Deploy AWS WAF in front of the ALB. Subscribe to an AWS Managed Rule for SQL injection filtering.

Answer: D

Explanation:

The AWS Cloud Operations and Security documentation confirms that AWS WAF (Web Application Firewall) is designed to protect web applications from application-layer threats, including SQL injection, cross-site scripting (XSS), and other OWASP Top 10 vulnerabilities.

When integrated with an Application Load Balancer, AWS WAF inspects incoming traffic using rule groups. The AWS Managed Rules for SQL Injection Protection provide preconfigured, continuously updated filters that detect and block malicious SQL patterns.

AWS Shield (Standard or Advanced) defends against DDoS attacks, not application-layer SQL attacks, and vulnerability scanners (Option C) only detect, not prevent, exploitation.

Thus, Option D provides the correct, managed, and automated protection aligned with AWS best practices.

Reference: AWS Cloud Operations & Security Guide - Protecting Applications from SQL Injection with AWS WAF Managed Rules

Question: 41

Optimization]

An AWS Lambda function is intermittently failing several times a day. A CloudOps engineer must find out how often this error occurred in the last 7 days.

Which action will meet this requirement in the MOST operationally efficient manner?

A. Use Amazon Athena to query the Amazon CloudWatch logs that are associated with the Lambda function.

B. Use Amazon Athena to query the AWS CloudTrail logs that are associated with the Lambda function.

C. Use Amazon CloudWatch Logs Insights to query the associated Lambda function logs.

D. Use Amazon OpenSearch Service to stream the Amazon CloudWatch logs for the Lambda function.

Answer: C

Explanation:

The AWS Cloud Operations and Monitoring documentation states that Amazon CloudWatch Logs Insights provides a purpose-built query engine for analyzing and visualizing log data directly within CloudWatch. For Lambda, all invocation results (including errors) are automatically logged to CloudWatch Logs.

By querying these logs with CloudWatch Logs Insights, the CloudOps engineer can efficiently count the number of "ERROR" or "Exception" occurrences over the past 7 days using simple SQL-like commands. This method is serverless, cost-efficient, and real-time.

Athena (Options A and B) would require exporting data to Amazon S3, and OpenSearch (Option D) adds unnecessary operational complexity.

Thus, Option C provides the most efficient and native AWS CloudOps approach for rapid Lambda error analysis.

Reference: AWS Cloud Operations & Monitoring Guide - Analyzing Lambda Logs with CloudWatch Logs Insights

Question: 42

A company uses AWS Organizations to manage multiple AWS accounts. A CloudOps engineer must identify all IPv4 ports open to 0.0.0.0/0 across the organization's accounts.

Which solution will meet this requirement with the LEAST operational effort?

A. Use the AWS CLI to print all security group rules for review.

B. Review AWS Trusted Advisor findings in an organizational view for the Security Groups - Specific Ports Unrestricted check.

C. Create an AWS Lambda function to gather security group rules from all accounts. Aggregate the findings in an Amazon S3 bucket.

D. Enable Amazon Inspector in each account. Run an automated workload discovery job.

Answer: B

Explanation:

According to AWS Cloud Operations and Governance documentation, AWS Trusted Advisor provides automated checks for security group rules across all accounts, including identifying ports open to 0.0.0.0/0.

When viewed in organizational mode, Trusted Advisor integrates with AWS Organizations, allowing administrators to access organization-wide security findings from a central management account. This approach requires no custom code, additional infrastructure, or manual inspection, providing immediate visibility and the lowest operational overhead.

AWS CLI scripts (Option A) or Lambda automation (Option C) introduce additional maintenance, and Amazon Inspector (Option D) is focused on instance-level vulnerabilities, not network access rules.

Therefore, Option B is the AWS-recommended CloudOps best practice for centralized and low-effort open-port auditing.

Reference: AWS Cloud Operations & Governance Guide - Using Trusted Advisor Organizational View for

Security Group Port Checks

Question: 43

A company hosts a static website in an Amazon S3 bucket, accessed globally via Amazon CloudFront. The Cache-Control max-age header is set to 1 hour, and Maximum TTL is set to 5 minutes. The CloudOps engineer observes that CloudFront is not caching objects for the expected duration.

What is the reason for this issue?

A. The Expires header has been set to 3 hours.

B. Cached assets are not expiring in the edge location.

C. Cache invalidation is missing in the CloudFront configuration.

D. Cache-duration settings conflict with each other.

Answer: D

Explanation:

As per the AWS Cloud Operations and Content Delivery documentation, CloudFront determines cache behavior by evaluating both origin headers (e.g., Cache-Control and Expires) and distribution-level TTL settings.

When Cache-Control max-age conflicts with the Maximum TTL configured in CloudFront, the shorter TTL value takes precedence. This results in CloudFront caching content for only 5 minutes instead of 1 hour, despite the origin headers suggesting a longer duration.

AWS documentation explicitly states: "When both origin cache headers and CloudFront TTL settings are defined, CloudFront uses the most restrictive caching period." This mismatch causes the perceived performance drop, as CloudFront frequently revalidates content.

Therefore, Option D is correct — cache-duration settings conflict with each other, leading to unexpected caching behavior.

Reference: AWS Cloud Operations & Content Delivery Guide - Interaction Between Cache-Control Headers and CloudFront TTL Settings

Question: 44

A company runs applications on Amazon EC2 instances. The company wants to ensure that SSH ports on the EC2 instances are never open. The company has enabled AWS Config and has set up the restricted-ssh AWS managed rule.

A CloudOps engineer must implement a solution to remediate SSH port access for noncompliant security groups.

What should the engineer do to meet this requirement with the MOST operational efficiency?

A. Configure the AWS Config rule to identify noncompliant security groups. Configure the rule to use the AWS-PublishSNSNotification AWS Systems Manager Automation runbook to send notifications about noncompliant resources.

B. Configure the AWS Config rule to identify noncompliant security groups. Configure the rule to use the AWS-DisableIncomingSSHOnPort22 AWS Systems Manager Automation runbook to remediate noncompliant resources.

C. Make an AWS Config API call to search for noncompliant security groups. Disable SSH access for noncompliant security groups by using a Deny rule.

D. Configure the AWS Config rule to identify noncompliant security groups. Manually update each noncompliant security group to remove the Allow rule.

Answer: B

Explanation:

The AWS Cloud Operations and Governance documentation specifies that AWS Config can be paired with AWS Systems Manager Automation runbooks for automatic remediation of noncompliant resources.

For SSH restrictions, the restricted-ssh managed rule detects any security group allowing inbound traffic on port 22. To automatically remediate these findings, AWS provides the AWS- DisableIncomingSSHOnPort22 runbook. This runbook programmatically removes inbound rules that allow port 22 traffic from affected security groups.

This approach achieves continuous compliance with minimal human intervention. By contrast, sending notifications (Option A) does not enforce remediation, API-based scripts (Option C) add operational overhead, and manual remediation (Option D) violates automation best practices.

Therefore, the most efficient CloudOps solution is Option B, using AWS Config with the AWS-DisableIncomingSSHOnPort22 automation runbook for automatic, scalable enforcement.

Reference: AWS Cloud Operations & Governance Guide - Automated Security Remediation Using Config Managed Rules and Systems Manager Runbooks

Question: 45

A CloudOps engineer created a VPC with a private subnet, a security group allowing all outbound traffic, and an endpoint for EC2 Instance Connect in the private subnet. The EC2 instance was launched without an SSH key

pair, using the same subnet and security group. However, the engineer cannot connect via EC2 Instance Connect endpoint.

How can the CloudOps engineer connect to the instance?

A. Create an inbound rule in the security group to allow HTTPS traffic on port 443 from the private subnet.

B. Create an inbound rule in the security group to allow SSH traffic on port 22 from the private subnet.

C. Create an IAM instance profile that allows AWS Systems Manager Session Manager to access the EC2 instance. Associate the instance profile with the instance.

D. Recreate the EC2 instance. Associate an SSH key pair with the instance.

Answer: C

Explanation:

According to the AWS Cloud Operations and EC2 Connectivity documentation, EC2 Instance Connect Endpoint allows access to instances without internet exposure or open SSH ports. However, for successful connectivity, the EC2 instance must have Systems Manager permissions through an IAM instance profile.

If no IAM instance profile is attached, the instance cannot establish a control channel with the Systems Manager service, and EC2 Instance Connect cannot authenticate the session.

Opening port 22 (Option B) is unnecessary and contradicts the private subnet design. HTTPS rules (Option A) are irrelevant because EC2 Instance Connect communicates through AWS APIs, not direct HTTPS connections. Recreating the instance with a key pair (Option D) bypasses the intended keyless connection mechanism.

Therefore, Option C — attaching an IAM instance profile with Systems Manager permissions — enables secure, private access through EC2 Instance Connect Endpoint.

Reference: AWS Cloud Operations & EC2 Connectivity Guide - Enabling EC2 Instance Connect Endpoint Access via Systems Manager Permissions

Question: 46

A company needs to upload gigabytes of files daily to Amazon S3 and requires higher throughput and faster upload speeds.

Which action should a CloudOps engineer take?

A. Create an Amazon CloudFront distribution with the GET HTTP method allowed and the S3 bucket as an origin.

B. Create an Amazon ElastiCache cluster and enable caching for the S3 bucket.

C. Set up AWS Global Accelerator and configure it with the S3 bucket.

D. Enable S3 Transfer Acceleration and use the acceleration endpoint when uploading files.

Answer: D

Explanation:

The AWS Cloud Operations and Storage documentation confirms that S3 Transfer Acceleration is designed to increase upload speed for objects transferred to S3 buckets over long distances.

It uses AWS Global Edge Network and Amazon CloudFront edge locations to route data through optimized network paths, reducing latency and achieving higher throughput compared to standard S3 uploads.

After enabling Transfer Acceleration on the bucket, users upload files to the accelerated endpoint (e.g., bucketname.s3-accelerate.amazonaws.com). This feature requires no changes to application logic besides endpoint modification and provides immediate performance improvement.

CloudFront (Option A) is for content delivery, not uploads. ElastiCache (Option B) and Global Accelerator (Option C) are unrelated to S3 upload performance.

Thus, Option D is correct — enable S3 Transfer Acceleration for faster, optimized file uploads.

Reference: AWS Cloud Operations & Storage Guide - Enhancing Upload Speed with Amazon S3 Transfer Acceleration

Question: 47

Optimization]

An ecommerce company uses Amazon ElastiCache (Redis OSS) for caching product queries. The CloudOps engineer observes a large number of cache evictions in Amazon CloudWatch metrics and needs to reduce evictions while retaining popular data in cache.

Which solution meets these requirements with the least operational overhead?

A. Add another node to the ElastiCache cluster.

B. Increase the ElastiCache TTL value.

C. Decrease the ElastiCache TTL value.

D. Migrate to a new ElastiCache cluster with larger nodes.

Answer: D

Explanation:

According to the AWS Cloud Operations and ElastiCache documentation, cache evictions occur when the cache runs out of memory and must remove items to make space for new data.

To reduce evictions and retain frequently accessed items, AWS recommends increasing the total available memory — either by scaling up to larger node types or scaling out by adding shards/nodes. Migrating to a cluster with larger nodes is the simplest and most efficient solution because it immediately expands capacity without architectural changes.

Adjusting TTL (Options B and C) controls expiration timing, not memory allocation. Adding a single node (Option A) may help, but redistributing data requires resharding, introducing more complexity.

Thus, Option D provides the lowest operational overhead and ensures high cache hit rates by increasing total cache memory.

Reference: AWS Cloud Operations & Performance Optimization Guide - Reducing Evictions and Scaling Amazon ElastiCache Clusters

Question: 48

A company's application servers in AWS account 111122223333 use a security group sg-1234abcd. They need to access a database hosted in account 444455556666. The VPCs are connected using a VPC peering connection (pcx-b04deed9).

A CloudOps engineer must configure the database's security group to allow new connections only from the application servers.

What should the engineer do?

A. Add an inbound rule to the database's security group. Reference 111122223333/sg-1234abcd as the source.

B. Add an inbound rule to the database's security group. Reference pcx-b04deed9/sg-1234abcd as the source.

C. Add an inbound rule to the database's security group. Reference sg-1234abcd as the source.

D. Add an inbound rule to the database's security group. Reference 444455556666/sg-1234abcd as the source.

Answer: C

Explanation:

According to AWS Cloud Operations and VPC Networking documentation, when VPCs are peered, security groups can reference peer account security groups directly to restrict traffic between them.

This feature allows specifying the security group ID (sg-1234abcd) from the source account (111122223333) in

the target database's security group inbound rule. AWS automatically validates that the VPCs are connected through an existing VPC peering connection and that mutual permissions are properly configured.

You do not prefix the security group ID with the account or peering connection (Options A and B), and using the destination account ID (Option D) is incorrect because it represents the database side, not the source.

Hence, the correct configuration is Option C, which references the application servers' security group directly for precise, least-privilege access control.

Reference: AWS Cloud Operations & Networking Guide - Using Security Group References Across VPC Peering Connections

Question: 49

A company's developers manually install software modules on Amazon EC2 instances to deploy new versions of a service. A security audit finds that instances contain inconsistent and unapproved modules.

A CloudOps engineer must create a new instance image that contains only approved software.

Which solution will meet these requirements?

A. Use Amazon Detective to continuously find and uninstall unauthorized modules from the instances.

B. Use Amazon GuardDuty to create and deploy an Amazon Machine Image (AMI) that includes only the approved modules.

C. Use AWS Systems Manager Run Command to install the approved modules on all running instances during an in-place update.

D. Use EC2 Image Builder to create and test an Amazon Machine Image (AMI) that includes only the approved modules. Update the deployment workflow to use the new AMI.

Answer: D

Explanation:

According to the AWS Cloud Operations and Deployment documentation, EC2 Image Builder is the AWS-managed service for automating the creation, maintenance, validation, and deployment of secure and compliant Amazon Machine Images (AMIs).

It allows CloudOps teams to define image pipelines that include only approved software modules and configuration scripts. EC2 Image Builder automatically tests and verifies these AMIs for compliance before deployment.

This process ensures configuration consistency, eliminates manual installation errors, and simplifies ongoing patch management. The service integrates with AWS Systems Manager, Amazon Inspector, and AWS

CloudFormation for end-to-end automation.

In contrast:

Amazon Detective and GuardDuty (Options A & B) are security monitoring tools, not image management solutions.

Run Command (Option C) applies ad-hoc updates but does not create standard, reusable AMIs.

Therefore, Option D is correct—EC2 Image Builder provides the most operationally efficient and compliant way to create an approved baseline AMI for future deployments.

Reference: AWS Cloud Operations & Deployment Guide - Building Secure, Consistent AMIs Using EC2 Image Builder

Question: 50

A company with millions of subscribers needs to automatically send notifications every Saturday. The company already uses Amazon SNS to send messages but has historically sent them manually.

Which solution will meet these requirements in the MOST operationally efficient way?

A. Launch a new Amazon EC2 instance. Configure a cron job to use the AWS SDK to send an SNS notification to subscribers every Saturday.

B. Create a rule in Amazon EventBridge that triggers every Saturday. Configure the rule to publish a notification to an SNS topic.

C. Create an SNS subscription to a message fanout that sends notifications to subscribers every Saturday.

D. Use AWS Step Functions scheduling to run a step every Saturday. Configure the step to publish a message to an SNS topic.

Answer: B

Explanation:

Per the AWS Cloud Operations and Event Management documentation, Amazon EventBridge provides native scheduling capabilities that can trigger events at defined intervals—such as weekly, daily, or cron-based schedules.

Creating an EventBridge rule that runs every Saturday and publishes a message to an SNS topic fully automates the notification process without maintaining servers or manual jobs. This approach is serverless, highly reliable, and fully managed by AWS.

By contrast:

EC2 cron jobs (Option A) require instance management, patching, and cost overhead.

SNS subscriptions (Option C) handle message delivery, not scheduling.

Step Functions (Option D) are designed for complex workflows, not simple scheduled triggers.

Thus, Option B provides the most operationally efficient CloudOps solution by integrating EventBridge scheduled events with SNS topics for automated, recurring notifications.

Reference: AWS Cloud Operations & Event Automation Guide - Scheduling Tasks and Notifications Using EventBridge and SNS

Question: 51

A company hosts a static website in Amazon S3 behind an Amazon CloudFront distribution. When new versions are deployed, users sometimes do not see updated content immediately.

Which solution will meet this requirement?

A. Configure the CloudFront distribution to add a custom Cache-Control header to requests for content from the S3 bucket.

B. Modify the distribution settings to specify the protocol as HTTPS only.

C. Attach the CachingOptimized managed cache policy to the distribution.

D. Create a CloudFront invalidation.

Answer: D

Explanation:

The AWS Cloud Operations and Content Delivery documentation explains that Amazon CloudFront caches objects in edge locations for a defined time based on TTL settings or origin headers. When new content is deployed to the S3 origin, previously cached versions remain in edge caches until they expire.

To immediately serve the new version, CloudOps engineers must initiate a CloudFront invalidation, which removes cached objects from all edge locations. This forces CloudFront to fetch the latest version from the origin (S3).

Invalidations can target individual objects (e.g., /index.html) or wildcard paths (e.g., /*) and are the AWS-recommended approach for dynamic content refresh after static site updates.

Changing headers (Option A), enforcing HTTPS (Option B), or applying caching policies (Option C) do not directly refresh outdated cache content.

Thus, Option D — issuing a CloudFront invalidation — ensures users receive the latest website content immediately after deployment.

Reference: AWS Cloud Operations & Content Delivery Guide - Using Invalidation to Refresh Cached Content in CloudFront

Question: 52

A company uses AWS Systems Manager Session Manager to manage EC2 instances in the eu-west-1 Region. The company wants private connectivity using VPC endpoints.

Which VPC endpoints are required to meet these requirements? (Select THREE.)

A. com.amazonaws.eu-west-1.ssm

B. com.amazonaws.eu-west-1.ec2messages

C. com.amazonaws.eu-west-1.ec2

D. com.amazonaws.eu-west-1.ssmmessages

E. com.amazonaws.eu-west-1.s3

F. com.amazonaws.eu-west-1.states

Answer: A, B, D

Explanation:

The AWS Cloud Operations and Systems Manager documentation states that to use Session Manager privately within a VPC (without internet access), three interface VPC endpoints must be configured:
com.amazonaws.<region>.ssm - enables Systems Manager core API communication.

com.amazonaws.<region>.ec2messages - allows the agent to send and receive messages between EC2 and Systems Manager.

com.amazonaws.<region>.ssmmessages - enables real-time interactive communication for Session Manager connections.

These endpoints ensure secure, private connectivity over the AWS network, eliminating the need for public internet routing.

Endpoints for S3, Step Functions, or EC2 API (Options C, E, F) are not required for Session Manager functionality.

Thus, the correct combination is A, B, and D, aligning with AWS CloudOps best practices for secure, private Systems Manager access.

Reference: AWS Cloud Operations & Systems Manager Guide - Configuring VPC Endpoints for Session Manager Private Connectivity

Question: 53

A company runs an application on Amazon EC2 instances behind an Elastic Load Balancer (ELB) in an Auto Scaling group. The application performs well except during a 2-hour period of daily peak traffic, when performance slows.

A CloudOps engineer must resolve this issue with minimal operational effort.

What should the engineer do?

A. Adjust the minimum capacity of the Auto Scaling group to the size required to meet the increased demand during the 2-hour period.

B. Adjust the launch template that is associated with the Auto Scaling group to be more sensitive to increases in user traffic.

C. Create a scheduled scaling action to scale out the number of EC2 instances shortly before the increase in user traffic occurs.

D. Manually add a few more EC2 instances to the Auto Scaling group to support the increase in user traffic. Enable instance scale-in protection on the Auto Scaling group.

_____ Answer: C

Explanation:

According to the AWS Cloud Operations and Compute documentation, when workloads exhibit predictable traffic patterns, the best practice is to use scheduled scaling for Amazon EC2 Auto Scaling groups.

With scheduled scaling, administrators can predefine the desired capacity of an Auto Scaling group to increase before anticipated demand (in this case, before the 2-hour peak) and scale back down afterward. This ensures that sufficient compute capacity is provisioned proactively, avoiding performance degradation while maintaining cost efficiency.

AWS notes: “Scheduled actions enable scaling your Auto Scaling group at predictable times, allowing you to

pre-warm instances before demand spikes."

Manual scaling (Option D) adds operational overhead. Adjusting launch templates (Option B) doesn't affect scaling behavior, and permanently increasing minimum capacity (Option A) wastes resources outside of peak hours.

Thus, Option C provides an automated, cost-effective, and operationally efficient CloudOps solution.

Reference: AWS Cloud Operations & Compute Optimization Guide - Using Scheduled Scaling for Predictable Workload Patterns

Question: 54

A company is storing backups in an Amazon S3 bucket. These backups must not be deleted for at least 3 months after creation.
What should the CloudOps engineer do?

A. Configure an IAM policy that denies the s3:DeleteObject action for all users. Three months after an object is written, remove the policy.

B. Enable S3 Object Lock on a new S3 bucket in compliance mode. Place all backups in the new S3 bucket with a retention period of 3 months.

C. Enable S3 Versioning on the existing S3 bucket. Configure S3 Lifecycle rules to protect the backups.

D. Enable S3 Object Lock on a new S3 bucket in governance mode. Place all backups in the new S3 bucket with a retention period of 3 months.

_____ Answer: B

Explanation:

Per the AWS Cloud Operations and Data Protection documentation, S3 Object Lock enforces write- once-read-many (WORM) protection on objects for a defined retention period.

There are two modes:

Compliance mode: Even the root user cannot delete or modify objects during the retention period.

Governance mode: Privileged users with special permissions can override lock settings.

For regulatory or audit requirements that prohibit deletion, Compliance mode is the correct choice. When configured with a 3-month retention period, all backup objects are protected from deletion until expiration, ensuring compliance with data retention mandates.

Versioning (Option C) alone does not prevent deletion. IAM-based restrictions (Option A) lack timebased enforcement and require manual intervention. Governance mode (Option D) is less strict and unsuitable for regulatory retention.

Thus, Option B is the correct CloudOps solution for immutable S3 backups.

Reference: AWS Cloud Operations & Storage Governance Guide - Implementing Retention with Amazon S3 Object Lock in Compliance Mode

Question: 55

A company's Amazon EC2 instance with high CPU utilization is a t3.large instance running a test web app. The company determines the app would run better on a compute-optimized large instance.

What should the CloudOps engineer do?

A. Migrate the EC2 instance to a compute optimized instance by using AWS VM Import/Export.

B. Enable hibernation on the EC2 instance. Change the instance type to a compute optimized instance. Disable hibernation on the EC2 instance.

C. Stop the EC2 instance. Change the instance type to a compute optimized instance. Start the EC2 instance.

D. Change the instance type to a compute optimized instance while the EC2 instance is running.

Answer: C

Explanation:

As described in the AWS Cloud Operations and EC2 Management documentation, changing an instance type (e.g., from T3 to C5) requires that the instance be stopped first. Once stopped, the engineer can modify the instance type through the AWS Management Console, CLI, or API, then start the instance again to apply changes.

This process preserves the root volume, networking configuration, and data, making it an operationally safe and efficient way to upgrade to a different instance family.

Changing the instance type while running (Option D) is unsupported. VM Import/Export (Option A) is for external VM migration. Hibernation (Option B) does not apply to type changes.

Thus, Option C is correct — stopping the instance, changing its type, and restarting it meets AWS best practices.

Reference: AWS Cloud Operations & Compute Guide - Modifying EC2 Instance Types for Performance Optimization

Question: 56

A company's CloudOps engineer monitors multiple AWS accounts in an organization and checks each account's AWS Health Dashboard. After adding 10 new accounts, the engineer wants to consolidate health alerts from all accounts.

Which solution meets this requirement with the least operational effort?

A. Enable organizational view in AWS Health.

B. Configure the Health Dashboard in each account to forward events to a central AWS CloudTrail log.

C. Create an AWS Lambda function to query the AWS Health API and write all events to an Amazon DynamoDB table.

D. Use the AWS Health API to write events to an Amazon DynamoDB table.

Answer: A

Explanation:
The AWS Cloud Operations and Governance documentation defines that enabling Organizational View in AWS Health allows the management account in AWS Organizations to view and aggregate health events from all member accounts.

This feature provides a single-pane-of-glass view of service health issues, account-specific events, and planned maintenance across the organization — without requiring additional automation or data pipelines.

Alternative options (B, C, and D) require custom integration and ongoing maintenance. CloudTrail does not natively forward AWS Health events, and custom Lambda or DynamoDB approaches increase complexity.

Therefore, Option A — enabling the Organizational View feature in AWS Health — is the most operationally efficient and AWS-recommended solution.

Reference: AWS Cloud Operations & Governance Guide - Consolidating Multi-Account Health Events with AWS Health Organizational View

Question: 57

A company that uses AWS Organizations recently implemented AWS Control Tower. The company now needs to centralize identity management. A CloudOps engineer must federate AWS IAM Identity Center with an external SAML 2.0 identity provider (IdP) to centrally manage access to all AWS accounts and cloud applications.

Which prerequisites must the CloudOps engineer have so that the CloudOps engineer can connect to the external IdP? (Select TWO.)

A. A copy of the IAM Identity Center SAML metadata

B. The IdP metadata, including the public X.509 certificate

C. The IP address of the IdP

D. Root access to the management account

E. Administrative permissions to the member accounts of the organization

Answer: A, B

Explanation:

According to the AWS Cloud Operations and Identity Management documentation, when configuring federation between IAM Identity Center (formerly AWS SSO) and an external SAML 2.0 identity provider, two key prerequisites are required:

The IAM Identity Center SAML metadata file — This is uploaded to the external IdP to establish trust, define SAML endpoints, and enable identity federation.

The IdP metadata (including the public X.509 certificate) — This information is imported into IAM Identity Center to validate authentication assertions and encryption signatures.

IAM Identity Center and the IdP exchange this metadata to mutually establish secure, bidirectional federation.

Network-level details such as IP addresses (Option C) are unnecessary. Root access (Option D) or permissions to member accounts (Option E) are not required; only Control Tower or IAM administrative permissions in the management account are needed for setup.

Thus, the correct answer is A and B — the SAML metadata from both sides is required for federation.

Reference: AWS Cloud Operations & Identity Management Guide - Federating IAM Identity Center with an External SAML 2.0 Provider

Question: 58

A company hosts an FTP server on EC2 instances. AWS Security Hub sends findings to Amazon EventBridge when the FTP port becomes publicly exposed in attached security groups.

A CloudOps engineer needs an automated, event-driven remediation solution to remove public access from security groups.

Which solution will meet these requirements?

A. Configure the existing EventBridge event to stop the EC2 instances that have the exposed port.

B. Create a cron job for the FTP server to invoke an AWS Lambda function. Configure the Lambda function to modify the security group of the identified EC2 instances and to remove the instances that allow public access.

C. Create a cron job for the FTP server that invokes an AWS Lambda function. Configure the Lambda function to modify the server to use SFTP instead of FTP.

D. Configure the existing EventBridge event to invoke an AWS Lambda function. Configure the function to remove the security group rule that allows public access.

Answer: D

Explanation:

Per the AWS Cloud Operations and Security Automation documentation, Security Hub integrates with Amazon EventBridge to publish findings in real time. These events can trigger automated responses using AWS Lambda functions or AWS Systems Manager Automation runbooks.

In this scenario, the correct CloudOps approach is to configure the existing EventBridge rule to invoke a Lambda function that inspects the event payload, identifies the affected security group, and removes the offending inbound rule (e.g., port 21 open to 0.0.0.0/0).

This event-driven remediation provides continuous compliance and eliminates manual intervention. Cron jobs

(Options B and C) contradict event-driven design and add operational overhead. Stopping instances (Option A) doesn't address the root cause — the insecure security group.

Thus, Option D aligns with AWS best practices for automated security remediation through EventBridge and Lambda.

Reference: AWS Cloud Operations & Security Hub Guide - Automating Security Remediation Using EventBridge and Lambda

Question: 59

A company has an application running on EC2 that stores data in an Amazon RDS for MySQL Single- AZ DB instance. The application requires both read and write operations, and the company needs failover capability with minimal downtime.

Which solution will meet these requirements?

A. Modify the DB instance to be a Multi-AZ DB instance deployment.

B. Add a read replica in the same Availability Zone where the DB instance is deployed.

C. Add the DB instance to an Auto Scaling group that has a minimum capacity of 2 and a desired capacity of 2.

D. Use RDS Proxy to configure a proxy in front of the DB instance.

_____ Answer: A

Explanation:

According to the AWS Cloud Operations and Database Reliability documentation, Amazon RDS MultiAZ deployments provide high availability and automatic failover by maintaining a synchronous standby replica in a different Availability Zone.

In the event of instance failure, planned maintenance, or Availability Zone outage, Amazon RDS automatically

promotes the standby to primary with minimal downtime (typically less than 60 seconds). The failover is transparent to applications because the DB endpoint remains the same.

By contrast, read replicas (Option B) are asynchronous and do not provide automated failover. Auto Scaling (Option C) applies to EC2, not RDS. RDS Proxy (Option D) improves connection management but does not add redundancy.

Thus, Option A — converting the RDS instance into a Multi-AZ deployment — delivers the required high availability and business continuity with minimal operational effort.

Reference: AWS Cloud Operations & Database Continuity Guide - Implementing Multi-AZ Deployments for Automatic RDS Failover

Question: 60

A company has an AWS CloudFormation template that includes an AWS::EC2::Instance resource and a custom resource (Lambda function). The Lambda function fails because it runs before the EC2 instance is launched.

Which solution will resolve this issue?

A. Add a DependsOn attribute to the custom resource. Specify the EC2 instance in the DependsOn attribute.

B. Update the custom resource's service token to point to a valid Lambda function.

C. Update the Lambda function to use the cfn-response module to send a response to the custom resource.

D. Use the Fn::If intrinsic function to check for the EC2 instance before the custom resource runs.

Answer: A

Explanation:

The AWS Cloud Operations and Infrastructure-as-Code documentation specifies that when using AWS CloudFormation, resources are created in parallel by default unless explicitly ordered using DependsOn.

If a custom resource (Lambda) depends on another resource (like an EC2 instance) to exist before execution, a DependsOn attribute must be added to enforce creation order. This ensures the EC2 instance is launched and available before the custom resource executes its automation logic.

Updating the service token (Option B) doesn't affect order of execution. The cfn-response module (Option C) handles callback communication but not sequencing. Fn::If (Option D) is for conditional creation, not dependency control.

Therefore, Option A is correct — adding a DependsOn attribute guarantees that CloudFormation provisions the EC2 instance before executing the Lambda custom resource.

Reference: AWS Cloud Operations & Infrastructure-as-Code Guide - Using DependsOn for Resource Creation Order in CloudFormation Templates

Question: 61

Optimization]

A company uses an Amazon Simple Queue Service (Amazon SQS) queue and Amazon EC2 instances in an Auto Scaling group with target tracking for a web application. The company collects the ASGAverageNetworkIn metric but notices that instances do not scale fast enough during peak traffic. There are a large number of SQS messages accumulating in the queue.

A CloudOps engineer must reduce the number of SQS messages during peak periods.

Which solution will meet this requirement?

A. Define and use a new custom Amazon CloudWatch metric based on the SQS ApproximateNumberOfMessagesDelayed metric in the target tracking policy.

B. Define and use Amazon CloudWatch metric math to calculate the SQS queue backlog for each instance in the target tracking policy.

C. Define and use step scaling by specifying a ChangeInCapacity value for the EC2 instances.

D. Define and use simple scaling by specifying a ChangeInCapacity value for the EC2 instances.

Answer: B

Explanation:

According to the AWS Cloud Operations and Auto Scaling documentation, scaling applications that consume Amazon SQS messages should be driven by queue backlog per instance, not by general system metrics such as network traffic or CPU.

The correct approach is to calculate a custom metric using CloudWatch metric math that divides the SQS metric ApproximateNumberOfMessagesVisible by the number of active EC2 instances in the Auto Scaling group. This "backlog per instance" value represents the average number of messages waiting to be processed by each instance.

Then, the CloudOps engineer can create a target tracking policy that automatically scales out or in based on maintaining a desired backlog threshold. This approach ensures dynamic, workload-driven scaling behavior that reacts in near real time to message volume.

Step and simple scaling (Options C and D) require manual thresholds and do not automatically balance the load per instance.

Thus, Option B—using CloudWatch metric math to define queue backlog per instance for target tracking—is the most effective and AWS-recommended CloudOps practice.

Reference: AWS Cloud Operations & Scaling Guide - Scaling Based on Amazon SQS Queue Backlog Per Instance

Question: 62

A CloudOps engineer has created an AWS Service Catalog portfolio and shared it with a second AWS account in the company, managed by a different CloudOps engineer.

Which action can the CloudOps engineer in the second account perform?

A. Add a product from the imported portfolio to a local portfolio.

B. Add new products to the imported portfolio.

C. Change the launch role for the products contained in the imported portfolio.

D. Customize the products in the imported portfolio.

Answer: A

Explanation:

Per the AWS Cloud Operations and Service Catalog documentation, when a portfolio is shared across AWS accounts, the recipient account imports the shared portfolio.

The recipient CloudOps engineer cannot modify the original products or their configurations but can:

Add products from the imported portfolio into their local portfolios for deployment,

Control end-user access in the recipient account, and

Manage local constraints or permissions.

However, the recipient cannot edit, delete, or reconfigure the shared products (Options B, C, and D). The source (owner) account retains full administrative control over products, launch roles, and lifecycle policies.

This model aligns with AWS CloudOps principles of centralized governance with distributed selfservice deployment across multiple accounts.

Thus, Option A is correct—imported portfolios allow the recipient to add products to a local portfolio but not alter the shared configuration.

Reference: AWS Cloud Operations & Governance Guide - Managing Shared AWS Service Catalog Portfolios Across Multiple Accounts

Question: 63

A company is migrating a legacy application to AWS. The application runs on EC2 instances across multiple Availability Zones behind an Application Load Balancer (ALB). The target group routing algorithm is set to weighted random, and the application requires session affinity (sticky sessions).

After deployment, users report random application errors that were not present before migration, even though target health checks are passing.

Which solution will meet this requirement?

A. Set the routing algorithm of the target group to least outstanding requests.

B. Turn on anomaly mitigation for the target group.

C. Turn off the cross-zone load balancing attribute of the target group.

D. Increase the deregistration delay attribute of the target group.

Answer: A

Explanation:

According to the AWS Cloud Operations and Elastic Load Balancing documentation, Application Load Balancer

(ALB) supports multiple routing algorithms to distribute requests among targets:

Round robin (default)

Least outstanding requests (LOR)

Weighted random

When applications require session affinity, AWS recommends using “least outstanding requests” as the load balancing algorithm because it reduces latency, distributes load evenly, and ensures consistent target responsiveness during high traffic.

Using weighted random routing with sticky sessions can cause sessions to be routed inconsistently if one target's capacity fluctuates, leading to session mismatches and application errors — especially when user sessions rely on instance-specific state.

Disabling cross-zone balancing (Option C) or adjusting deregistration delay (Option D) does not address routing inconsistency. Anomaly mitigation (Option B) protects against target performance degradation, not sticky-session misrouting.

Therefore, the correct solution is Option A — changing the target group's routing algorithm to least outstanding requests ensures smoother, predictable session handling and resolves random application errors.

Reference: AWS Cloud Operations & Load Balancing Guide - Optimizing Application Load Balancer Routing Algorithms for Stateful Applications

Question: 64

A CloudOps engineer must manage the security of an AWS account. Recently, an IAM user's access key was mistakenly uploaded to a public code repository. The engineer must identify everything that was changed using this compromised key.

How should the CloudOps engineer meet these requirements?

A. Create an Amazon EventBridge rule to send all IAM events to an AWS Lambda function for analysis.

B. Query Amazon EC2 logs by using Amazon CloudWatch Logs Insights for all events initiated with the compromised access key within the suspected timeframe.

C. Search AWS CloudTrail event history for all events initiated with the compromised access key within the suspected timeframe.

D. Search VPC Flow Logs for all events initiated with the compromised access key within the suspected timeframe.

_____ Answer: C

Explanation:

According to the AWS Cloud Operations and Security documentation, AWS CloudTrail is the authoritative service for recording API activity across all AWS services within an account.

When an access key is compromised, CloudTrail logs all API requests made using that key, including details such as:

The user identity (access key ID) that made the request,

The service, operation, resource, and timestamp affected, and

The source IP address and region of the request.

By searching the CloudTrail event history for the specific access key ID, the CloudOps engineer can identify every action performed by that key during the suspected breach window.

Other options are incorrect:

EventBridge (A) is event-driven, not historical.

CloudWatch Logs (B) monitors system logs, not AWS API activity.

VPC Flow Logs (D) track network-level traffic, not API calls.

Therefore, the correct solution is Option C — using AWS CloudTrail event history to audit and trace all actions executed via the compromised access key.

Reference: AWS Cloud Operations & Security Management Guide - Investigating Compromised Access Keys Using AWS CloudTrail

Question: 65

A company runs custom statistical analysis software on a cluster of Amazon EC2 instances. The software is highly sensitive to network latency between nodes, although network throughput is not a limitation.

Which solution will minimize network latency?

A. Place all the EC2 instances into a cluster placement group.

B. Configure and assign two Elastic IP addresses for each EC2 instance.

C. Configure jumbo frames on all the EC2 instances in the cluster.

D. Place all the EC2 instances into a spread placement group in the same AWS Region.

_____ Answer: A

Explanation:

The AWS Cloud Operations and Compute documentation explains that placement groups control how EC2 instances are physically arranged within AWS data centers to optimize network performance.

Among the available placement strategies:

Cluster placement groups place instances physically close together within a single Availability Zone, connected through high-bandwidth, low-latency networking (ideal for tightly coupled, HPC, or distributed workloads).

Spread placement groups distribute instances across distinct racks or Availability Zones for fault tolerance, increasing latency.

Partition placement groups separate instances into partitions for isolation, not latency reduction.

Therefore, to minimize latency for workloads such as computational clusters, the CloudOps engineer should use a cluster placement group. This placement ensures single-digit microsecond latency and enhanced packet rate performance between instances.

Elastic IPs (Option B) do not influence internal networking. Jumbo frames (Option C) can marginally improve throughput but do not reduce propagation latency. Spread placement (Option D) increases distance, worsening latency.

Hence, Option A — using a cluster placement group — delivers the lowest possible network latency and is AWS's best-practice design for HPC-style clusters.

Reference: AWS Cloud Operations & Compute Optimization Guide - Optimizing EC2 Networking with Cluster Placement Groups for Low Latency Workloads