DUMPSBOSS.

AWS Certified DevOps Engineer - Professional

Amazon AWS DOP-C02

Version Demo

Total Demo Questions: 51

Total Premium Questions: 517

Buy Premium PDF

https://dumpsboss.co

support@dumpsboss.co

support@dumpsboss.co
dumpsboss.co

Topic Break Down

Topic	No. of Questions
Topic 1, SDLC Automation	106
Topic 2, Configuration Management and IaC	72
Topic 3, Resilient Cloud Solutions	89
Topic 4, Monitoring and Logging	81
Topic 5, Incident and Event Response	22
Topic 6, Security and Compliance	147
Total	517

QUESTION NO: 1

A company has an organization in AWS Organizations with many OUs that contain many AWS accounts. The organization has a dedicated delegated administrator AWS account.

The company needs the accounts in one OU to have server-side encryption enforced for all Amazon Elastic Block Store (Amazon EBS) volumes and Amazon Simple Queue Service (Amazon SQS) queues that are created or updated on an AWS CloudFormation stack.

Which solution will enforce this policy before a CloudFormation stack operation in the accounts of this OU?

A. Activate trusted access to CloudFormation StackSets. Create a CloudFormation Hook that enforces server-side encryption on EBS volumes and SQS queues. Deploy the Hook across the accounts in the OU by using StackSets.

B. Set up AWS Config in all the accounts in the OU. Use AWS Systems Manager to deploy AWS Config rules that enforce server-side encryption for EBS volumes and SQS queues across the accounts in the OU.

C. Write an SCP to deny the creation of EBS volumes and SQS queues unless the EBS volumes and SQS queues have server-side encryption. Attach the SCP to the OU.

D. Create an AWS Lambda function in the delegated administrator account that checks whether server-side encryption is enforced for EBS volumes and SQS queues. Create an IAM role to provide the Lambda function access to the accounts in the OU.

ANSWER: A

Explanation:

Activate trusted access to CloudFormation StackSets. Create a CloudFormation Hook that enforces server-side encryption on EBS volumes and SQS queues. Deploy the Hook across the accounts in the OU by using StackSets. This approach provides preventive governance at the point where CloudFormation evaluates a stack create or update operation. A CloudFormation hook can inspect the properties of targeted resources before CloudFormation provisions or changes them. The hook can require encryption-related properties for `AWS::EC2::Volume` and `AWS::SQS::Queue`; when a resource does not meet the policy, the hook can fail the operation, preventing the noncompliant stack action from proceeding.

CloudFormation StackSets with AWS Organizations trusted access allow the delegated administrator to distribute the hook configuration consistently to the accounts in the target OU, rather than requiring separate manual activation in each account. This is particularly suitable for an organization with many accounts and OUs because StackSets can target organizational units and manage deployments centrally. The enforcement applies specifically to resources managed through CloudFormation and occurs before resource provisioning or update, which directly meets the timing requirement. AWS documents hooks as a mechanism for proactive validation during CloudFormation operations and supports StackSets for deploying hooks across organizational accounts. See AWS CloudFormation Hooks documentation and AWS Organizations trusted access for CloudFormation StackSets.

QUESTION NO: 2

A DevOps engineer must implement a solution that immediately terminates Amazon EC2 instances in Auto Scaling groups when cryptocurrency mining activity is detected.

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

A. Configure Amazon Route 53 to send query logs directly to Amazon CloudWatch Logs. Create an AWS Lambda function that runs every 5 minutes and checks the query logs for domains related to cryptocurrency activity. If the domains are found, terminate the identified EC2 instances.

B. Configure VPC Flow Logs to send flow logs to an Amazon S3 bucket. Create an AWS Lambda function that runs every 5 minutes and invokes an Amazon Athena query to find IP addresses associated with cryptocurrency activity. If the IP addresses are found, terminate the identified EC2 instances.

C. Enable Amazon GuardDuty. Monitor EC2 findings. Create an Amazon EventBridge rule with GuardDuty as the event source. Create an AWS Lambda function that is triggered by the EventBridge rule. Configure the Lambda function to parse the event and terminate the identified EC2 instances.

D. Enable AWS Security Hub. Monitor EC2 findings. Create an Amazon EventBridge rule with Security Hub as the event source. Create an AWS Lambda function that is triggered by the EventBridge rule. Configure the Lambda function to parse the event and terminate the identified EC2 instances.

ANSWER: C

Explanation:

Enable Amazon GuardDuty. Monitor EC2 findings. Create an Amazon EventBridge rule with GuardDuty as the event source. Create an AWS Lambda function that is triggered by the EventBridge rule. Configure the Lambda function to parse the event and terminate the identified EC2 instances. This solution uses AWS-managed threat detection and event-driven remediation, minimizing the amount of custom detection code and operational infrastructure required. GuardDuty continuously analyzes relevant AWS telemetry and produces findings for suspicious EC2 behavior, including cryptocurrency-mining-related activity. GuardDuty findings are automatically delivered to Amazon EventBridge, allowing an EventBridge rule to match the applicable finding types and invoke a Lambda function promptly. The function can obtain the affected instance ID from the finding details and call the Amazon EC2 termination API. Because the instance belongs to an Auto Scaling group, the group can launch a replacement instance as needed to maintain its desired capacity. This approach is well suited to immediate automated containment: GuardDuty supplies the managed detection capability, EventBridge supplies near-real-time routing, and Lambda performs the focused remediation action. AWS documents GuardDuty finding delivery through EventBridge at GuardDuty findings in EventBridge and lists cryptocurrency-related EC2 finding types at GuardDuty EC2 finding types.

QUESTION NO: 3

A company is reviewing its 1AM policies. One policy written by the DevOps engineer has been (lagged as too permissive. The policy is used by an AWS Lambda function that issues a stop command to Amazon EC2 instances tagged with Environment: NonProduccion over the weekend. The current policy is:

What changes should the engineer make to achieve a policy ot least permission? (Select THREE.)

A. Add a condition using lambda:SourceFunctionArn to allow the EC2 action only when it originates from the intended Lambda function.

B. Replace an unrestricted resource scope with Amazon EC2 instance ARNs limited to the required AWS Region and account.

C. Keep Resource set to * so that the function can stop any Amazon EC2 instance.

D. Add a condition requiring the ec2:ResourceTag/Environment tag to have the NonProduction value.

E. Grant ec2:* so that the function can manage all Amazon EC2 operations.

F. Attach the AWS managed AdministratorAccess policy to the Lambda execution role.

ANSWER: A B D

Explanation:

Use an IAM condition with the `lambda:SourceFunctionArn` global condition key to limit the EC2 API permission to requests that originate from the designated Lambda function. This protects the Lambda execution role's EC2 capability from being used outside the intended function context. Scope the policy resource to EC2 instance ARNs in the required AWS Region and account rather than using an unrestricted resource scope. Where the function must act on dynamically created instances, an instance ARN wildcard can still be constrained to the specific account and Region.

Add an `ec2:ResourceTag/Environment` condition requiring the intended nonproduction tag value. Amazon EC2 supports resource-tag condition keys for `ec2:StopInstances`, so the permission can be granted only when the target instance carries the approved Environment tag. Together, the source-function condition, instance-level resource scope, and resource-tag condition restrict who or what uses the permission, which instance resources can be targeted, and which tagged instances may be stopped. This is a practical application of IAM least privilege while allowing the scheduled operational automation to continue. See the Lambda source function ARN documentation and the Amazon EC2 service authorization reference.

QUESTION NO: 4

A company deployed an Amazon CloudFront distribution that accepts requests and routes to an Amazon API Gateway HTTP API. During a recent security audit, the company discovered that requests from the internet could reach the HTTP API without using the CloudFront distribution.

A DevOps engineer must ensure that connections to the HTTP API use the CloudFront distribution.

Which solution will meet these requirements?

A. Enable VPC Flow Logs to identify requests that reach the HTTP API.

B. Deploy AWS WAF in front of the CloudFront distribution.

C. Implement an identity-based policy on the CloudFront distribution that requires authentication to make requests to the HTTP API.

D. Implement a custom header in the CloudFront distribution. Implement an AWS Lambda authorizer associated with the HTTP API that verifies the custom header.

ANSWER: D

Explanation:

Implementing a custom header in the CloudFront distribution and an AWS Lambda authorizer on the HTTP API that verifies that header ensures that requests are accepted only when they have passed through the configured CloudFront origin. CloudFront can add a secret, static origin custom header whenever it forwards a viewer request to API Gateway. The Lambda authorizer receives the request headers before API Gateway invokes the API integration and can allow the request only when the expected header name and secret value are present. Requests made directly to the API Gateway invoke URL will not receive the CloudFront-added header and therefore will be denied by the authorizer. The header value should be treated as a secret, stored and managed securely, and rotated when appropriate. This pattern preserves CloudFront as the intended public entry point while allowing API Gateway HTTP APIs to use a request authorizer for enforcement. CloudFront origin custom headers are documented at Amazon CloudFront Developer Guide, and HTTP API Lambda authorizer behavior is documented in the Amazon API Gateway Developer Guide.

QUESTION NO: 5

A company is deploying a new application that uses Amazon EC2 instances. The company needs a solution to query application logs and AWS account API activity Which solution will meet these requirements?

A. Use the Amazon CloudWatch agent to send logs from the EC2 instances to Amazon CloudWatch Logs Configure AWS CloudTrail to deliver the API logs to Amazon S3 Use CloudWatch to query both sets of logs.

B. Use the Amazon CloudWatch agent to send logs from the EC2 instances to Amazon CloudWatch Logs Configure AWS CloudTrail to deliver the API logs to CloudWatch Logs Use CloudWatch Logs Insights to query both sets of logs.

C. Use the Amazon CloudWatch agent to send logs from the EC2 instances to Amazon Kinesis Configure AWS CloudTrail to deliver the API logs to Kinesis Use Kinesis to load the data into Amazon Redshift Use Amazon Redshift to query both sets of logs.

D. Use the Amazon CloudWatch agent to send logs from the EC2 instances to Amazon S3 Use AWS CloudTrail to deliver the API togs to Amazon S3 Use Amazon Athena to query both sets of logs in Amazon S3.

ANSWER: B

Explanation:

Use the Amazon CloudWatch agent to send logs from the EC2 instances to Amazon CloudWatch Logs Configure AWS CloudTrail to deliver the API logs to CloudWatch Logs Use CloudWatch Logs Insights to query both sets of logs. This solution centralizes application log events and AWS account API activity in CloudWatch Logs, where they can be searched together without building a separate ingestion pipeline or data warehouse. The CloudWatch agent can collect application and system log files from Amazon EC2 and publish them to designated CloudWatch Logs log groups. A CloudTrail trail can

also be configured with a CloudWatch Logs log group as its destination, allowing recorded management and data events to be delivered to CloudWatch Logs. CloudWatch Logs Insights then provides an interactive query language for analyzing log data across selected log groups, enabling operators to investigate application behavior alongside the API calls made in the AWS account. This is particularly useful for operational troubleshooting, auditing, and incident response because queries can filter on timestamps, fields, error messages, principals, API names, and resource identifiers. For configuration details, see the Amazon CloudWatch agent documentation and CloudTrail delivery to CloudWatch Logs documentation.

QUESTION NO: 6

A company uses an organization in AWS Organizations that has all features enabled. The company uses AWS Backup in a primary account and uses an AWS Key Management Service (AWS KMS) key to encrypt the backups.

The company needs to automate a cross-account backup of the resources that AWS Backup backs up in the primary account. The company configures cross-account backup in the Organizations management account. The company creates a new AWS account in the organization and configures an AWS Backup backup vault in the new account. The company creates a KMS key in the new account to encrypt the backups. Finally, the company configures a new backup plan in the primary account. The destination for the new backup plan is the backup vault in the new account.

When the AWS Backup job in the primary account is invoked, the job creates backups in the primary account. However, the backups are not copied to the new account ' s backup vault.

Which combination of steps must the company take so that backups can be copied to the new account ' s backup vault? (Select TWO.)

A. Edit the backup vault access policy in the new account to allow access to the primary account.

B. Edit the backup vault access policy in the primary account to allow access to the new account.

C. Edit the backup vault access policy in the primary account to allow access to the KMS key in the new account.

D. Edit the key policy of the KMS key in the primary account to share the key with the new account.

E. Edit the key policy of the KMS key in the new account to share the key with the primary account.

ANSWER: A E

Explanation:

Cross-account AWS Backup copy jobs require authorization at the destination backup vault and for the destination account’s encryption key. “Edit the backup vault access policy in the new account to allow access to the primary account.” is required because the destination vault must explicitly permit the source account to perform the `backup:CopyIntoBackupVault` action. AWS Backup vault access policies are resource-based policies, so this permission is evaluated by the vault that receives the copied recovery point.

“Edit the key policy of the KMS key in the new account to share the key with the primary account.” is also required. The destination vault encrypts the copied backup using the KMS key in the destination account. Its key policy must authorize the source account to use the key through AWS Backup, subject to the required KMS permissions. This enables AWS Backup to encrypt the copy when it is written into the destination vault while retaining the destination account’s control of the key.

AWS documents that the destination vault policy must allow copies from the source account and that cross-account backup copies require appropriate AWS KMS key-policy permissions. See Creating backup copies across AWS accounts and Encryption for AWS Backup.

QUESTION NO: 7

A DevOps engineer has created an AWS CloudFormation template that deploys an application on Amazon EC2 instances The EC2 instances run Amazon Linux The application is deployed to the EC2 instances by using shell scripts that contain user data. The EC2 instances have an 1AM instance profile that has an 1AM role with the AmazonSSMManagedInstanceCore managed policy attached

The DevOps engineer has modified the user data in the CloudFormation template to install a new version of the application. The engineer has also applied the stack update. However, the application was not updated on the running EC2 instances. The engineer needs to ensure that the changes to the application are installed on the running EC2 instances.

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

A. Configure the user data content to use the Multipurpose Internet Mail Extensions (MIME) multipart format. Set the scripts-user parameter to always in the text/cloud-config section.

B. Refactor the user data commands to use the cfn-init helper script. Update the user data to install and configure the cfn-hup helper script to monitor and apply metadata changes.

C. Configure an EC2 launch template for the EC2 instances. Create a new EC2 Auto Scaling group. Associate the Auto Scaling group with the EC2 launch template Use the AutoScalingScheduledAction update policy for the Auto Scaling group.

D. Refactor the user data commands to use an AWS Systems Manager document (SSM document). Add an AWS CLI command in the user data to use Systems Manager Run Command to apply the SSM document to the EC2 instances

E. Refactor the user data command to use an AWS Systems Manager document (SSM document) Use Systems Manager State Manager to create an association between the SSM document and the EC2 instances.

ANSWER: B E

Explanation:

Refactoring the user data commands to use the `cfn-init` helper script and installing and configuring `cfn-hup` provides a CloudFormation-native mechanism for applying changes to existing instances. `cfn-init` reads configuration from the resource's CloudFormation metadata and performs the declared installation and configuration actions. The `cfn-hup` daemon runs continuously on the instance, detects changes to configured CloudFormation metadata paths, and can invoke `cfn-init` again to apply the updated application configuration. This avoids relying on ordinary EC2 user data, which is generally processed only during initial instance launch. AWS documents this update pattern in its cfn-hup reference.

Using an AWS Systems Manager document with a Systems Manager State Manager association is also appropriate because the instances already have the Systems Manager core managed policy. State Manager targets the managed instances and runs the association according to its schedule or when the association is created or updated. The SSM document can contain the shell commands required to install the required application version, giving the engineer a centrally managed and repeatable way to converge running instances to the desired configuration. For document revisions, the association can be configured to use the intended document version so updates are applied consistently. See the AWS Systems Manager State Manager documentation.

QUESTION NO: 8

A company uses AWS Secrets Manager to store a set of sensitive API keys that an AWS Lambda function uses. When the Lambda function is invoked, the Lambda function retrieves the API keys and makes an API call to an external service. The Secrets Manager secret is encrypted with the default AWS Key Management Service (AWS KMS) key.

A DevOps engineer needs to update the infrastructure to ensure that only the Lambda function ' s execution role can access the values in Secrets Manager. The solution must apply the principle of least privilege.

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

A. Update the default KMS key for Secrets Manager to allow only the Lambda function ' s execution role to decrypt.

B. Create a KMS customer managed key that trusts Secrets Manager and allows the Lambda function ' s execution role to decrypt. Update Secrets Manager to use the new customer managed key.

C. Create a KMS customer managed key that trusts Secrets Manager and allows the account ' s :root principal to decrypt. Update Secrets Manager to use the new customer managed key.

D. Ensure that the Lambda function ' s execution role has KMS permissions scoped to the customer managed key resource. Configure the permissions so that the role can decrypt the KMS key used to encrypt the Secrets Manager secret.

E. Remove all KMS permissions from the Lambda function ' s execution role.

ANSWER: B D

Explanation:

Using a KMS customer managed key that permits AWS Secrets Manager to use the key and permits the Lambda execution role to decrypt provides control that is not available with the AWS managed key for Secrets Manager. The customer managed key policy can authorize Secrets Manager's use of the key for the secret while restricting cryptographic use to the intended Lambda role. The secret must then be updated to use that customer managed key so its encrypted data is protected by the new key policy.

The Lambda execution role also needs narrowly scoped IAM permissions for the specific KMS key, including `kms:Decrypt`, because retrieving an encrypted secret requires KMS decryption authorization in addition to the Secrets Manager read permission. Scoping the permission's Resource to the customer managed key ARN ensures that the function cannot use unrelated KMS keys. Together, the customer managed key policy and the execution role's resource-scoped permissions form the required authorization path while limiting decryption capability to the role that runs the function. AWS documents the use of customer managed keys with Secrets Manager at Encrypt and decrypt secrets and the required KMS permissions for secret retrieval at IAM policy examples for AWS Secrets Manager.

QUESTION NO: 9

A company uses an AWS CodeArtifact repository to store Python packages that the company developed internally. A DevOps engineer needs to use AWS CodeDeploy to deploy an application to an Amazon EC2 instance. The application uses a Python package that is stored in the CodeArtifact repository. A BeforeInstall lifecycle event hook will install the package.

The DevOps engineer needs to grant the EC2 instance access to the CodeArtifact repository.

Which solution will meet this requirement?

A. Create a service-linked role for CodeArtifact. Associate the role with the EC2 instance. Use the aws codeartifact get-authorization-token CLI command on the instance.

B. Configure a resource-based policy for the CodeArtifact repository that allows the Read-FromRepository action for the EC2 instance principal.

C. Configure ACLs on the CodeArtifact repository to allow the EC2 instance to access the Python package.

D. Create an instance profile that contains an IAM role that has access to CodeArtifact. Associate the instance profile with the EC2 instance. Use the aws codeartifact login CLI command on the instance.

ANSWER: D

Explanation:

Create an instance profile that contains an IAM role that has access to CodeArtifact. Associate the instance profile with the EC2 instance. Use the aws codeartifact login CLI command on the instance. This approach provides the EC2 instance with temporary AWS credentials through its attached instance-profile role, avoiding static access keys in the CodeDeploy deployment or lifecycle hook. The role must allow the CodeArtifact actions required to obtain and use a repository authorization token, including `codeartifact:GetAuthorizationToken`, `codeartifact:GetRepositoryEndpoint`, and `codeartifact:ReadFromRepository` for the relevant domain and repository. It also needs `sts:GetServiceBearerToken` with the appropriate CodeArtifact service condition. During the BeforeInstall hook, `aws codeartifact login --tool pip` retrieves a short-lived authorization token and configures pip to use the specified CodeArtifact repository endpoint. The subsequent package installation therefore authenticates using the instance role's permissions and the generated repository configuration. This is well suited to automated CodeDeploy deployments because the instance can retrieve fresh credentials and tokens at deployment time. AWS documents CodeArtifact authentication and the package-manager configuration performed by the login command in the CodeArtifact pip configuration guide, and documents attaching IAM roles to EC2 through instance profiles.

QUESTION NO: 10

A DevOps engineer is working on a member account in an organization in AWS Organizations with all features enabled. The account has sensitive data stored in Amazon S3 buckets.

The DevOps engineer must ensure that all public access to S3 buckets in the account is blocked. If the account-level public access settings change in the future, the changes must be reverted automatically so that all public access is blocked again.

Which solution meets these requirements?

A. Enable AWS Security Hub in the account. Enable the Security Hub control to evaluate the account-level block public access settings. Enable automated remediation for the Security Hub control.

B. Set up AWS Config in the account. Create an AWS Config managed rule that evaluates the S3 block public access settings. Enable automated remediation for the rule by using a predefined AWS Systems Manager runbook to configure S3 block public access settings.

C. In the organization ' s management account, create an SCP that denies S3 actions from outside the AWS account. Attach the SCP to the member account.

D. Enable Amazon Macie in the account. Create an Amazon EventBridge rule with an event pattern that matches Macie policy findings. Configure the rule with an EventBridge target to run a predefined AWS Systems Manager runbook to configure S3 block public access settings.

ANSWER: B

Explanation:

Set up AWS Config in the account. Create an AWS Config managed rule that evaluates the S3 block public access settings. Enable automated remediation for the rule by using a predefined AWS Systems Manager runbook to configure S3 block public access settings. This approach provides both continuous compliance detection and automatic restoration of the intended configuration. AWS Config includes the managed `s3-account-level-public-access-blocks-periodic` rule, which evaluates whether all four account-level Amazon S3 Block Public Access settings are enabled. These settings prevent public ACLs and public bucket policies from being used to expose S3 data publicly.

When AWS Config finds that the account configuration no longer meets the rule requirements, an associated remediation action can invoke AWS Systems Manager Automation. The remediation workflow reapplies the required S3 Block Public Access configuration without requiring an engineer to manually identify or correct the change. AWS Config remediation can be configured to run automatically for noncompliant resources, which directly meets the requirement that future changes are reverted. The rule is periodic, so its evaluation and remediation behavior should be configured with the required compliance-detection interval in mind. See the AWS Config managed rule documentation and AWS Config automatic remediation documentation.

QUESTION NO: 11

An ecommerce company has chosen AWS to host its new platform. The company's DevOps team has started building an AWS Control Tower landing zone. The DevOps team has set the identity store within AWS IAM Identity Center (AWS Single Sign-On) to external identity provider (IdP) and has configured SAML 2.0.

The DevOps team wants a robust permission model that applies the principle of least privilege. The model must allow the team to build and manage only the team's own resources.

Which combination of steps will meet these requirements? (Choose three.)

A. Create IAM policies that include the required permissions. Include the aws:PrincipalTag condition key.

B. Create permission sets. Attach an inline policy that includes the required permissions and uses the aws:PrincipalTag condition key to scope the permissions.

C. Create a group in the IdP. Place users in the group. Assign the group to accounts and the permission sets in IAM Identity Center.

D. Create a group in the IdP. Place users in the group. Assign the group to OUs and IAM policies.

E. Enable attributes for access control in IAM Identity Center. Apply tags to users. Map the tags as key-value pairs.

F. Enable attributes for access control in IAM Identity Center. Map attributes from the IdP as key-value pairs.

ANSWER: B C F

Explanation:

Using IAM Identity Center permission sets with an inline policy that uses the `aws:PrincipalTag` condition key provides attribute-based access control (ABAC). The policy can grant resource-management actions only when a resource's tags match attributes carried in the authenticated user's session. For example, a team attribute supplied by the identity provider can be compared with a `Team` resource tag, allowing team members to manage only resources that belong to their own team. This supports least privilege while avoiding separate policies for every individual team.

Because the identity source is an external SAML 2.0 identity provider, attributes for access control must be enabled in IAM Identity Center and mapped from identity-provider attributes to key-value pairs. IAM Identity Center then passes the mapped attributes as principal tags when users access AWS accounts through assigned permission sets. Finally, managing membership in an identity-provider group and assigning that group to the appropriate AWS accounts and permission sets centrally controls which users can obtain the scoped role permissions. Together, these capabilities implement scalable, tag-based authorization across the Control Tower landing zone. See AWS IAM Identity Center ABAC documentation and AWS global condition key documentation for aws:PrincipalTag.

QUESTION NO: 12

A company is using an organization in AWS Organizations to manage multiple AWS accounts. The company's development team wants to use AWS Lambda functions to meet resiliency requirements and is rewriting all applications to work with Lambda functions that are deployed in a VPC. The development team is using Amazon Elastic Pile System (Amazon EFS) as shared storage in Account A in the organization.

The company wants to continue to use Amazon EPS with Lambda Company policy requires all serverless projects to be deployed in Account B.

A DevOps engineer needs to reconfigure an existing EFS file system to allow Lambda functions to access the data through an existing EPS access point.

Which combination of steps should the DevOps engineer take to meet these requirements? (Select THREE.)

A. Update the EFS file system policy to provide Account B with access to mount and write to the EFS file system in Account A.

B. Create SCPs to set permission guardrails with fine-grained control for Amazon EFS.

C. Create a new EFS file system in Account B Use AWS Database Migration Service (AWS DMS) to keep data from Account A and Account B synchronized.

D. Update the Lambda execution roles with permission to access the VPC and the EFS file system.

E. Create a VPC peering connection to connect Account A to Account B.

F. Configure the Lambda functions in Account B to assume an existing IAM role in Account A.

ANSWER: A D E

Explanation:

Cross-account Lambda access to Amazon EFS requires both network connectivity and authorization on each side of the request. Updating the EFS file system policy to provide Account B with access to mount and write to the EFS file system in Account A grants the consuming account permission at the EFS resource layer. The policy should allow the required EFS client actions, such as `elasticfilesystem:ClientMount` and, when writes are required, `elasticfilesystem:ClientWrite`. Access through the existing EFS access point also preserves the access point's configured POSIX identity and directory behavior.

Creating a VPC peering connection to connect Account A to Account B provides private network routing between the Lambda VPC and the VPC containing EFS mount targets. The associated route tables, security groups, and DNS configuration must permit NFS traffic on TCP port 2049 to the mount targets.

Updating the Lambda execution roles with permission to access the VPC and the EFS file system supplies the identity-based permissions Lambda needs. The execution role needs permissions for Lambda's VPC networking operations and the applicable EFS client actions. Together, the execution-role permissions and the EFS file system resource policy satisfy AWS's cross-account authorization model. See the AWS Lambda EFS configuration documentation and the Amazon EFS IAM authorization documentation.

QUESTION NO: 13

A healthcare services company is concerned about the growing costs of software licensing for an application for monitoring patient wellness. The company wants to create an audit process to ensure that the application is running exclusively on Amazon EC2 Dedicated Hosts. A DevOps engineer must create a workflow to audit the application to ensure compliance.

What steps should the engineer take to meet this requirement with the LEAST administrative overhead?

A. Use AWS Systems Manager Configuration Compliance. Use calls to the put-compliance-items API action to scan and build a database of noncompliant EC2 instances based on their host placement configuration. Use an Amazon DynamoDB table to store these instance IDs for fast access. Generate a report through Systems Manager by calling the list-compliance-summaries API action.

B. Use custom Java code running on an EC2 instance. Set up EC2 Auto Scaling for the instance depending on the number of instances to be checked. Send the list of noncompliant EC2 instance IDs to an Amazon SQS queue. Set up another worker instance to process instance IDs from the SQS queue and write them to Amazon DynamoD

C. Use AWS Config. Identify all EC2 instances to be audited by enabling Config Recording on all Amazon EC2 resources for the region. Create a custom AWS Config rule that triggers an AWS Lambda function by using the " config-rule-change-triggered " blueprint. Modify the LambdaevaluateCompliance () function to verify host placement to return a NON_COMPLIANT result if the instance is not running on an EC2 Dedicated Host. Use the AWS Config repo

D. Use AWS CloudTrail. Identify all EC2 instances to be audited by analyzing all calls to the EC2 RunCommand API action. Invoke a AWS Lambda function that analyzes the host placement of the instance. Store the EC2 instance ID of noncompliant resources in an Amazon RDS for MySQL DB instance. Generate a report by querying the RDS instance and exporting the query results to a CSV text file.

E. Use AWS Config and enable recording for Amazon EC2 instances. Deploy the AWS Config managed rule ec2-host-based to evaluate whether instances are running on dedicated hosts, and use AWS Config compliance results to audit and report compliance.

ANSWER: E

Explanation:

Use AWS Config and enable recording for Amazon EC2 instances. Deploy the AWS Config managed rule `ec2-host-based` to evaluate whether instances are running on dedicated hosts, and use AWS Config compliance results to audit and report compliance. This is the lowest-administration solution because AWS Config continuously records supported resource configurations and evaluates them against a managed rule maintained by AWS. The `ec2-host-based` rule determines whether an EC2 instance is launched on a Dedicated Host and reports resources that do not meet that requirement as noncompliant. AWS Config retains configuration history and compliance results, providing an auditable record without the company needing to operate custom scanners, maintain a database, or implement its own compliance-evaluation logic. The engineer can use the AWS Config console, APIs, or aggregated Config data to review the compliance posture and produce audit evidence. Enabling EC2 recording ensures that instance configuration changes, including launches or placement changes relevant to the rule evaluation, are captured and assessed. AWS documents the managed rule and its dedicated-host evaluation behavior at AWS Config: ec2-host-based. More generally, AWS Config provides resource configuration history and compliance evaluation capabilities as described in the AWS Config Developer Guide.

QUESTION NO: 14

A company containerized its Java app and uses CodePipeline. They want to scan images in ECR for vulnerabilities and reject images with critical vulnerabilities in a manual approval stage.

Which solution meets these?

A. Basic scanning with EventBridge for Inspector findings and Lambda to reject manual approval if critical vulnerabilities found.

B. Enhanced scanning, Lambda invokes Inspector for SBOM, exports to S3, Athena queries SBOM, rejects manual approval on critical findings.

C. Enhanced scanning, EventBridge listens to Detective scan findings, Lambda rejects manual approval on critical vulnerabilities.

D. Enhanced scanning, EventBridge listens to Inspector scan findings, Lambda rejects manual approval on critical vulnerabilities.

ANSWER: D

Explanation:

Enhanced scanning, EventBridge listens to Inspector scan findings, Lambda rejects manual approval on critical vulnerabilities is correct because Amazon ECR enhanced scanning is integrated with Amazon Inspector and provides continuous vulnerability scanning for container images. When Inspector identifies or updates an ECR image finding, it emits an `Inspector2 Finding` event to Amazon EventBridge. An EventBridge rule can filter these events for ECR container-image findings whose severity is `CRITICAL` and invoke a Lambda function. The function can correlate the finding to the pipeline execution and call the CodePipeline `PutApprovalResult` API for the relevant manual approval action with a rejected status. This automates the decision while retaining the manual approval stage as the pipeline's quality gate: images without critical findings can proceed for human review or approval, whereas an image with a critical vulnerability causes the approval action to be rejected and stops the release. Amazon Inspector's ECR integration and EventBridge finding events make this an event-driven solution without needing to periodically query scan results or construct a separate vulnerability-analysis workflow. See the Amazon Inspector documentation for ECR scanning and the CodePipeline PutApprovalResult API reference.

QUESTION NO: 15

A company is building a web and mobile application that uses a serverless architecture powered by AWS Lambda and Amazon API Gateway The company wants to fully automate the backend Lambda deployment based on code that is pushed to the appropriate environment branch in an AWS CodeCommit repository

The deployment must have the following:

• Separate environment pipelines for testing and production

• Automatic deployment that occurs for test environments only

Which steps should be taken to meet these requirements ' ?

A. Configure a new AWS CodePipelme service Create a CodeCommit repository for each environment Set up CodePipeline to retrieve the source code from the appropriate repository Set up the deployment step to deploy the Lambda functions with AWS CloudFormation.

B. Create two AWS CodePipeline configurations for test and production environments Configure the production pipeline to have a manual approval step Create aCodeCommit repository for each environment Set up each CodePipeline to retrieve the source code from the appropriate repository Set up the deployment step to deploy the Lambda functions with AWS CloudFormation.

C. Create two AWS CodePipeline configurations for test and production environments Configure the production pipeline to have a manual approval step. Create one CodeCommit repository with a branch for each environment Set up each CodePipeline to retrieve the source code from the appropriate branch in the repository. Set up the deployment step to deploy the Lambda functions with AWS CloudFormation

D. Create an AWS CodeBuild configuration for test and production environments Configure the production pipeline to have a manual approval step. Create one CodeCommit repository with a branch for each environment Push the Lambda function code to an Amazon S3 bucket Set up the deployment step to deploy the Lambda functions from the S3 bucket.

ANSWER: C

Explanation:

Create two AWS CodePipeline configurations for test and production environments, with each pipeline using a distinct branch in one AWS CodeCommit repository as its source. This arrangement directly maps branch pushes to the relevant environment: a change pushed to the test branch starts the test pipeline and can proceed automatically through its deployment stages. The production pipeline can use the production branch but include a manual approval action before the deployment action, ensuring that a person explicitly authorizes a production release while test deployments remain fully automated.

AWS CodePipeline supports CodeCommit as a source provider and can detect source changes to initiate pipeline executions. A manual approval action pauses a pipeline until an authorized IAM principal approves or rejects the release, which provides the required production deployment control. AWS CloudFormation is suitable for the deployment stage because a version-controlled template can define and update the Lambda functions, API Gateway resources, IAM roles, and related serverless infrastructure consistently for each environment. Environment-specific parameter values or configuration can be supplied separately to the test and production CloudFormation deployments while retaining a common application definition. See the AWS CodePipeline triggering documentation and the AWS CodePipeline manual approval documentation.

QUESTION NO: 16

A company uses the AWS Cloud Development Kit (AWS CDK) to define its application. The company uses a pipeline that consists of AWS CodePipeline and AWS CodeBuild to deploy the CDK application. The company wants to introduce unit tests to the pipeline to test various infrastructure components. The company wants to ensure that a deployment proceeds if no unit tests result in a failure.

Which combination of steps will enforce the testing requirement in the pipeline? (Select TWO.)

A. Update the CodeBuild build phase commands to run the tests then to deploy the application. Set the OnFailure phase property to ABORT.

B. Update the CodeBuild build phase commands to run the tests then to deploy the application. Add the --rollback true flag to the cdk deploy command.

C. Update the CodeBuild build phase commands to run the tests then to deploy the application. Add the --require-approval any-change flag to the cdk deploy command.

D. Create a test that uses the AWS CDK assertions module. Use the template.hasResourceProperties assertion to test that resources have the expected properties.

E. Create a test that uses the cdk diff command. Configure the test to fail if any resources have changed.

ANSWER: A D

Explanation:

"Create a test that uses the AWS CDK assertions module. Use the template.hasResourceProperties assertion to test that resources have the expected properties." is correct because CDK assertions allow unit tests to inspect the synthesized AWS CloudFormation template without deploying infrastructure. A test can synthesize a stack into a Template object and use `hasResourceProperties` to verify that a resource of a specified type contains the required configuration. This provides repeatable validation of the infrastructure definition as part of normal application testing.

"Update the CodeBuild build phase commands to run the tests then to deploy the application. Set the OnFailure phase property to ABORT." is correct because CodeBuild executes build commands in order, so deployment is reached only after the test command succeeds. Configuring the phase failure behavior as `ABORT` explicitly stops the build when a command in that phase fails. The failed CodeBuild action is then reported to CodePipeline, preventing subsequent deployment

processing. Together, template assertions and an aborting test phase ensure that a deployment proceeds only when the infrastructure unit tests pass. See the AWS CDK testing guide and the CodeBuild buildspec reference.

QUESTION NO: 17

A video-sharing company stores its videos in Amazon S3. The company has observed a sudden increase in video access requests, but the company does not know which videos are most popular. The company needs to identify the general access pattern for the video files. This pattern includes the number of users who access a certain file on a given day, as well as the numb A DevOps engineer manages a large commercial website that runs on Amazon EC2 The website uses Amazon Kinesis Data Streams to collect and process web togs The DevOps engineer manages the Kinesis consumer application, which also runs on Amazon EC2

Sudden increases of data cause the Kinesis consumer application to (all behind and the Kinesis data streams drop records before the records can be processed The DevOps engineer must implement a solution to improve stream handling

Which solution meets these requirements with the MOST operational efficiency ' '

er of pull requests for certain files.

How can the company meet these requirements with the LEAST amount of effort?

A. Activate S3 server access logging. Import the access logs into an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.

B. Activate S3 server access logging. Use Amazon Athena to create an external table with the log files. Use Athena to create a SQL query to analyze the access patterns.

C. Invoke an AWS Lambda function for every S3 object access event. Configure the Lambda function to write the file access information, such as user. S3 bucket, and file key, to an Amazon Aurora database. Use an Aurora SQL query to analyze the access patterns.

D. Record an Amazon CloudWatch Logs log message for every S3 object access event. Configure a CloudWatch Logs log stream to write the file access information, such as user, S3 bucket, and file key, to an Amazon Kinesis Data Analytics for SQL application. Perform a sliding window analysis.

ANSWER: B

Explanation:

Activate S3 server access logging. Use Amazon Athena to create an external table with the log files. Use Athena to create a SQL query to analyze the access patterns. S3 server access logging provides detailed records for requests made against an S3 bucket, including the requester, requested object key, operation, time, and other request details. Delivering these logs to an S3 destination creates a durable, low-management data set that can be queried directly. Athena can define an external table over the log files in S3 and run standard SQL without loading data into, operating, or scaling a database. Queries can group records by object key and day, count distinct requesters to estimate the number of users accessing each video, and count request operations to identify heavily requested files. This serverless approach is well suited to ad hoc exploration when the company does not yet know which objects are popular, because it requires no continuously running ingestion or analytics infrastructure. Athena charges based on data scanned, and partitioning the access-log data by date can further improve query performance and cost as log volume grows. AWS documents S3 access-log fields and delivery behavior in the Amazon S3 Server access logging guide, and documents querying S3 data through external tables in the Amazon Athena User Guide.

QUESTION NO: 18

A DevOps learn has created a Custom Lambda rule in AWS Config. The rule monitors Amazon Elastic Container Repository (Amazon ECR) policy statements for ecr: ' actions. When a noncompliant repository is detected, Amazon EventBridge uses Amazon Simple Notification Service (Amazon SNS) to route the notification to a security team.

When the custom AWS Config rule is evaluated, the AWS Lambda function fails to run.

Which solution will resolve the issue?

A. Modify the Lambda function ' s resource policy to grant AWS Config permission to invoke the function.

B. Modify the SNS topic policy to include configuration changes for EventBridge to publish to the SNS topic.

C. Modify the Lambda function ' s execution role to include configuration changes for custom AWS Config rules.

D. Modify all the ECR repository policies to grant AWS Config access to the necessary ECR API actions.

ANSWER: A

Explanation:

Modify the Lambda function ' s resource policy to grant AWS Config permission to invoke the function. AWS Config custom Lambda rules are evaluated by having the AWS Config service invoke the specified Lambda function. Lambda invocation by an AWS service principal requires an explicit resource-based permission on the function, commonly created with the `lambda:AddPermission` API action. The permission should allow the `config.amazonaws.com` principal to perform `lambda:InvokeFunction`; it can also be scoped to the relevant AWS Config rule by using a source ARN condition. Once this permission exists, AWS Config can invoke the function whenever it evaluates the rule, allowing the function to inspect the ECR repository configuration and return its compliance evaluation. This service-to-function invocation authorization is distinct from the Lambda execution role, which supplies permissions that the function itself uses after invocation. AWS documentation for custom Lambda rules explicitly requires permission for AWS Config to invoke the Lambda function. See the AWS Config custom Lambda rule documentation and the Lambda service invocation permissions documentation.

QUESTION NO: 19

An ecommerce company is receiving reports that its order history page is experiencing delays in reflecting the processing status of orders. The order processing system consists of an AWS Lambda function that uses reserved concurrency. The Lambda function processes order messages from an Amazon Simple Queue Service (Amazon SQS) queue and inserts processed orders into an Amazon DynamoDB table. The DynamoDB table has auto scaling enabled for read and write capacity.

Which actions should a DevOps engineer take to resolve this delay? (Choose two.)

A. Check the ApproximateAgeOfOldestMessage metric for the SQS queue. Increase the Lambda function concurrency limit.

B. Check the ApproximateAgeOfOldestMessage metnc for the SQS queue Configure a redrive policy on the SQS queue.

C. Check the NumberOfMessagesSent metric for the SQS queue. Increase the SQS queue visibility timeout.

D. Check the WriteThrottleEvents metric for the DynamoDB table. Increase the maximum write capacity units (WCUs) for the table's scaling policy.

E. Check the Throttles metric for the Lambda function. Increase the Lambda function timeout.

ANSWER: A D

Explanation:

Check the ApproximateAgeOfOldestMessage metric for the SQS queue. Increase the Lambda function concurrency limit. This metric indicates how long the oldest unprocessed message has been waiting in the queue, making it a direct signal of a processing backlog. Because the function has reserved concurrency, that reservation places an upper limit on the number of simultaneous Lambda invocations available to poll and process SQS messages. Increasing the reserved concurrency limit, when account concurrency is available, lets Lambda process more batches concurrently and reduce message age and order-status latency.

Check the WriteThrottleEvents metric for the DynamoDB table. Increase the maximum write capacity units (WCUs) for the table's scaling policy. Each successfully processed order must be written to DynamoDB before the order history page can reflect its status. Write throttling means the table lacks sufficient available write capacity for the workload. Although auto scaling can increase provisioned capacity, its configured maximum can prevent it from scaling high enough during sustained or sudden demand. Raising that maximum permits DynamoDB auto scaling to provision the write throughput required to absorb the processing rate. See the AWS Lambda documentation for SQS event source mappings and the DynamoDB monitoring documentation.

QUESTION NO: 20

A DevOps engineer needs a solution to catalog all applications that are installed on on-premises servers and Amazon EC2 instances. The solution must track historical changes and retain the data for 7 years.

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

A. Install the AWS Systems Manager Agent (SSM Agent) on the on-premises servers and the EC2 instances. Use Systems Manager Inventory to collect inventory data. Set the retention period for the inventory data to 7 years.

B. Install the AWS Systems Manager Agent (SSM Agent) on the on-premises servers and the EC2 instances. Use Systems Manager Inventory to collect inventory data. Use a Systems Manager resource data sync to send inventory data from all the managed nodes to a single Amazon S3 bucket. Set an object lifecycle rule to delete objects after 7 years.

C. Install the AWS Systems Manager Agent (SSM Agent) on the on-premises servers and the EC2 instances. Use Systems Manager Inventory to collect inventory data. Use AWS Config to capture history and track changes. Set the retention period for the AWS Config data to 7 years.

D. Configure an Amazon S3 bucket with an object lifecycle rule that deletes objects after 7 years. Create a custom script to collect inventory data from the on-premises servers and the EC2 instances. Configure the script to send the inventory data to the S3 bucket. Configure an IAM role for the EC2 instances and IAM Roles Anywhere for the on-premises servers with appropriate permissions.

ANSWER: B

Explanation:

Installing the AWS Systems Manager Agent (SSM Agent) on the on-premises servers and the EC2 instances, using Systems Manager Inventory, and configuring a Systems Manager resource data sync to Amazon S3 provides a managed, centralized inventory solution with minimal custom operational work. Systems Manager Inventory can collect application inventory from managed nodes, including both EC2 instances and registered on-premises servers. A resource data sync automatically sends inventory data from managed nodes to a designated S3 bucket, allowing inventory records to be centrally retained and queried over time rather than relying solely on the current inventory view. Applying an S3 Lifecycle expiration rule to the synchronized objects for 7 years meets the required retention period using durable, low-maintenance S3 storage. This approach avoids maintaining custom collection scripts, scheduling infrastructure, and bespoke authentication workflows. The inventory collection cadence and managed-node registration remain under Systems Manager, while S3 lifecycle management enforces retention automatically. For implementation details, see the AWS Systems Manager Inventory documentation and the Systems Manager resource data sync documentation.

QUESTION NO: 21

A company has multiple AWS accounts. The company uses **AWS IAM Identity Center** that is integrated with a third-party **SAML 2.0 identity provider (IdP)** .

The **attributes for access control** feature is enabled in IAM Identity Center. The attribute mapping list maps the **department** key from the IdP to the ${path:enterprise.department} attribute. All existing Amazon EC2 instances have a **d1** , **d2** , or **d3** department tag that corresponds to three of the companyâ€™s departments.

A DevOps engineer must create policies based on the matching attributes. The policies must grant each user access to only the EC2 instances that are tagged with the userâ€™s respective department name.

Which condition key should the DevOps engineer include in the custom permissions policies to meet these requirements?

A. " Condition " : { " ForAllValues:StringEquals " : { " aws:TagKeys " : [" department "] }}

B. " Condition " : { " StringEquals " : { " aws:PrincipalTag/department " : " ${aws:ResourceTag/department} " }}

C. " Condition " : { " StringEquals " : { " ec2:ResourceTag/department " : " ${aws:PrincipalTag/department} " }}

D. " Condition " : { " ForAllValues:StringEquals " : { " ec2:ResourceTag/department " : [" d1 " , " d2 " , " d3 "] }}

E. "Condition": { "StringEquals": { "ec2:ResourceTag/department": "${path:enterprise.department}" }}

ANSWER: E

Explanation:

"ec2:ResourceTag/department": "${path:enterprise.department}" is the appropriate condition because it implements attribute-based access control using the IAM Identity Center access-control attribute that was explicitly configured in the scenario. IAM Identity Center makes mapped identity attributes available for use in AWS account permission policies through attribute references in the `${path:...}` format. Since the IdP's department attribute is mapped to `${path:enterprise.department}`, that exact reference represents the signed-in user's department value during authorization.

The EC2 service authorization model supports the `ec2:ResourceTag/tag-key` condition key for applicable EC2 actions. Setting `ec2:ResourceTag/department` as the condition key evaluates the target EC2 resource's `department` tag. Comparing it to `${path:enterprise.department}` requires the instance tag value to equal the authenticated user's mapped department attribute. Therefore, a user whose mapped department is `d1` can access only instances tagged `department=d1`, with the same policy dynamically enforcing the corresponding boundary for `d2` and `d3` users. This avoids separate static policies for each department while preserving least-privilege access. See the IAM Identity Center ABAC documentation and the Amazon EC2 service authorization reference.

QUESTION NO: 22

A company runs an application in an Amazon Elastic Container Service (Amazon ECS) service that is associated with an Elastic Load Balancing (ELB) target group. A DevOps engineer updates the service to include a new task definition version. The DevOps engineer notices that the deployment does not finish running. New tasks enter a stopped state soon after the tasks launch. The task definition references an Amazon CloudWatch Logs log group.

Which issues are most likely the cause of the failing deployment? (Select TWO.)

A. The target group health check is failing, which causes Amazon ECS to stop the tasks.

B. The IAM role that the DevOps engineer used to update the ECS service does not have the Amazon ECS RunTask permission.

C. The CloudWatch Logs log group that is referenced in the task definition does not exist.

D. The task role does not have the required permissions to launch the task.

E. An essential container in the ECS task is exiting.

ANSWER: C E

Explanation:

The CloudWatch Logs log group that is referenced in the task definition does not exist. Amazon ECS initializes the configured `awslogs` log driver as part of task startup. If the referenced log group is absent and ECS cannot create it, task initialization fails and the task stops. A log group can be created automatically only when `awslogs-create-group` is enabled and the task execution role has the required `logs:CreateLogGroup` permission. Otherwise, the log group must already exist. This failure is commonly reported in the task stopped reason as a resource-initialization or log-stream creation error. See the Amazon ECS awslogs logging driver documentation.

An essential container in the ECS task is exiting. ECS treats a task as stopped when an essential container exits. Consequently, a new revision whose main application container terminates immediately—for example, because of an application startup error, invalid command, or missing runtime configuration—will not provide the healthy running tasks needed for the service deployment to reach steady state. The stopped-task details, container exit code, and CloudWatch logs are the appropriate evidence to inspect. AWS documents this essential-container lifecycle behavior in the ECS task definition parameters documentation.

QUESTION NO: 23

A DevOps team supports an application that runs on a large number of Amazon EC2 instances in an Auto Scaling group. The DevOps team uses AWS CloudFormation to deploy the EC2 instances. The application recently experienced an issue. A single instance returned errors to a large percentage of requests. The EC2 instance responded as healthy to both Amazon EC2 and Elastic Load Balancing health checks. The DevOps team collects application logs in Amazon CloudWatch by using the embedded metric format. The DevOps team needs to receive an alert if any EC2 instance is responsible for more than half of all errors. Which combination of steps will meet these requirements with the LEAST operational overhead? (Select TWO.)

A. Create a CloudWatch Contributor Insights rule that groups logs from the CloudWatch application logs based on instance ID and errors.

B. Create a resource group in AWS Resource Groups. Use the CloudFormation stack to group the resources for the application. Add the application to CloudWatch Application Insights. Use the resource group to identify the application.

C. Create a metric filter for the application logs to count the occurrence of the term " Error. " Create a CloudWatch alarm that uses the METRIC_COUNT function to determine whether errors have occurred. Configure the CloudWatch alarm to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic to notify the DevOps team.

D. Create a CloudWatch alarm that uses the INSIGHT_RULE_METRIC function to determine whether a specific instance is responsible for more than half of all errors reported by EC2 instances. Configure the CloudWatch alarm to send a notification to an Amazon Simple Notification Service (Amazon SNS) topic to notify the DevOps team.

E. Create a CloudWatch subscription filter for the application logs that filters for errors and invokes an AWS Lambda function. Configure the Lambda function to send the instance ID and error in a notification to an Amazon Simple Notification Service (Amazon SNS) topic to notify the DevOps team.

ANSWER: A D

Explanation:

Creating a CloudWatch Contributor Insights rule that groups application log events by instance ID and errors provides the required per-instance error attribution without maintaining custom aggregation code. Contributor Insights analyzes matching CloudWatch Logs events and identifies the top contributors according to the rule’s grouping keys and aggregate values. A rule can therefore track how many errors each EC2 instance contributes during each evaluation period. Because the application already emits structured data through embedded metric format, the logs contain a suitable source for identifying error events and their originating instance IDs.

A CloudWatch alarm using the `INSIGHT_RULE_METRIC` function can then consume Contributor Insights outputs in metric math. In particular, the alarm expression can compare the largest contributor’s error count with the total error count and enter the ALARM state when that contributor exceeds half of the total. Sending the alarm state change to an Amazon SNS topic provides the requested team notification. This is managed entirely through CloudWatch rules, metric math, alarms, and SNS, avoiding operationally managed Lambda code or custom monitoring infrastructure. See the Amazon CloudWatch Contributor Insights documentation and the CloudWatch metric math documentation.

QUESTION NO: 24

A company's DevOps engineer uses AWS Systems Manager to perform maintenance tasks during maintenance windows. The company has a few Amazon EC2 instances that require a restart after notifications from AWS Health. The DevOps engineer needs to implement an automated solution to remediate these notifications. The DevOps engineer creates an Amazon EventBridge rule.

How should the DevOps engineer configure the EventBridge rule to meet these requirements?

A. Configure an event source of AWS Health, a service of EC2. and an event type that indicates instance maintenance. Target a Systems Manager document to restart the EC2 instance.

B. Configure an event source of Systems Manager and an event type that indicates a maintenance window. Target a Systems Manager document to restart the EC2 instance.

C. Configure an event source of AWS Health, a service of EC2, and an event type that indicates instance maintenance. Target a newly created AWS Lambda function that registers an automation task to restart the EC2 instance during a maintenance window.

D. Configure an event source of EC2 and an event type that indicates instance maintenance. Target a newly created AWS Lambda function that registers an automation task to restart the EC2 instance during a maintenance window.

ANSWER: A

Explanation:

Configure an event source of AWS Health, a service of EC2, and an event type that indicates instance maintenance. Target a Systems Manager document to restart the EC2 instance. AWS Health publishes service events to Amazon EventBridge, including events for Amazon EC2 scheduled changes and instance-maintenance-related conditions. An EventBridge event pattern can therefore filter for AWS Health events whose affected service is EC2 and whose event type represents the maintenance condition requiring remediation. The event includes information about affected resources, enabling the target automation to identify the relevant instance.

Amazon EventBridge supports AWS Systems Manager Automation as a target. The rule can invoke an Automation runbook, such as a suitable Systems Manager document that performs an EC2 restart, and pass the affected instance ID as an input parameter. This creates an event-driven remediation path without manual intervention: AWS Health emits the notification, EventBridge matches it, and Systems Manager runs the restart automation. Systems Manager maintenance windows remain useful for separately scheduled operational work, while this configuration directly responds to the AWS Health event that identifies the necessary restart. See Monitoring AWS Health events with Amazon EventBridge and Amazon EventBridge targets.

QUESTION NO: 25

A company has a single developer writing code for an automated deployment pipeline. The developer is storing source code in an Amazon S3 bucket for each project. The company wants to add more developers to the team but is concerned about code conflicts and lost work The company also wants to build a test environment to deploy newer versions of code for testing and allow developers to automatically deploy to both environments when code is changed in the repository.

What is the MOST efficient way to meet these requirements?

A. Create an AWS CodeCommit repository tor each project, use the mam branch for production code: and create a testing branch for code deployed to testing Use feature branches to develop new features and pull requests to merge code to testing and main branches.

B. Create another S3 bucket for each project for testing code, and use an AWS Lambda function to promote code changes between testing and production buckets Enable versioning on all buckets to prevent code conflicts.

C. Create an AWS CodeCommit repository for each project, and use the main branch for production and test code with different deployment pipelines for each environment Use feature branches to develop new features.

D. Enable versioning and branching on each S3 bucket, use the main branch for production code, and create a testing branch for code deployed to testing. Have developers use each branch for developing in each environment.

ANSWER: A

Explanation:

Create an AWS CodeCommit repository tor each project, use the mam branch for production code: and create a testing branch for code deployed to testing Use feature branches to develop new features and pull requests to merge code to testing and main branches. This approach provides the source-control workflows needed for a growing development team. A Git-based repository supports isolated feature development, commits, history, and branch-based collaboration, while pull requests provide a controlled review and merge process. This protects work from being inadvertently overwritten and gives the team a traceable way to resolve merge conflicts before changes reach shared branches.

The testing branch can act as the source revision for an automated deployment pipeline targeting the test environment. After validation, a pull request can merge the same changes into the main branch, which triggers the production deployment pipeline. AWS CodePipeline can use a source-code repository and detect source changes to automatically start releases, allowing the two environment-specific pipelines to deploy the appropriate branch without manual copying of source artifacts. Feature branches also let several developers work independently until their changes are ready for integration. See the AWS CodeCommit branch documentation and AWS CodePipeline concepts documentation.

QUESTION NO: 26

A company wants to deploy a workload on several hundred Amazon EC2 instances. The company will provision the EC2 instances in an Auto Scaling group by using a launch template.

The workload will pull files from an Amazon S3 bucket, process the data, and put the results into a different S3 bucket. The EC2 instances must have least-privilege permissions and must use temporary security credentials.

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

A. Create an IAM role that has the appropriate permissions for S3 buckets. Add the IAM role to an instance profile.

B. Update the launch template to include the IAM instance profile.

C. Create an IAM user that has the appropriate permissions for Amazon S3. Generate a secret key and token.

D. Create a trust anchor and profile. Attach the IAM role to the profile.

E. Update the launch template. Modify the user data to use the new secret key and token.

ANSWER: A B

Explanation:

Create an IAM role that has the appropriate permissions for S3 buckets. Add the IAM role to an instance profile. This provides the workload with an identity whose permissions can be scoped precisely to the required S3 actions and resources—for example, permission to read objects from the source bucket and write objects to the destination bucket. The role must trust the EC2 service principal, and the instance profile is the mechanism through which the role is associated with an EC2 instance. EC2 retrieves and rotates temporary role credentials automatically; applications can obtain those credentials through the instance metadata service rather than storing access keys.

Update the launch template to include the IAM instance profile. Because the Auto Scaling group launches all instances from this template, the association ensures every newly launched instance receives the intended role consistently, including instances added during scaling or replaced after failure. This approach supports hundreds of instances without distributing, rotating, or embedding long-term credentials, while enforcing the same least-privilege access model across the fleet. AWS documents the EC2 role and instance-profile association at IAM roles for Amazon EC2 and describes specifying an instance profile in a launch template at Create a launch template.

QUESTION NO: 27

A company uses AWS CodeBuild projects in private subnets to build Java applications. The projects have no internet access. The company stores private Maven packages in AWS CodeArtifact and must allow the build projects to download packages without using long-term IAM credentials.

Which combination of actions should a DevOps engineer take to meet these requirements? (Choose three.)

A. Create interface VPC endpoints for the CodeArtifact API service and the CodeArtifact repositories service. Enable private DNS for the endpoints.

B. Grant the CodeBuild service role permissions for codeartifact:GetAuthorizationToken and codeartifact:ReadFromRepository for the required domain and repository.

C. Configure the build project to obtain a temporary CodeArtifact authorization token by using the CodeBuild service role credentials.

D. Store an IAM access key and secret access key for a package user in the CodeBuild project environment variables.

E. Create a gateway VPC endpoint for CodeArtifact and associate the endpoint with the private subnet route table.

F. Configure an internet gateway on the VPC and assign public IP addresses to the CodeBuild build containers.

ANSWER: A B C

Explanation:

The CodeBuild service role provides temporary credentials to build containers. The role needs permissions to retrieve a CodeArtifact authorization token and to read packages from the required repository. These permissions can be scoped to the specific CodeArtifact domain and repository.

For private connectivity, create interface VPC endpoints for the CodeArtifact API and repositories services and enable private DNS. The build project can then access the CodeArtifact APIs and repository endpoints without a NAT gateway or internet gateway. The build commands can use the AWS CLI or the CodeArtifact package-manager configuration commands to obtain a temporary authorization token by using the service role credentials.

See AWS CodeArtifact interface VPC endpoints and Authentication and tokens for CodeArtifact.

QUESTION NO: 28

A company has configured an Amazon S3 event source on an AWS Lambda function The company needs the Lambda function to run when a new object is created or an existing object IS modified In a particular S3 bucket The Lambda function will use the S3 bucket name and the S3 object key of the incoming event to read the contents of the created or modified S3 object The Lambda function will parse the contents and save the parsed contents to an Amazon DynamoDB table.

The Lambda function ' s execution role has permissions to read from the S3 bucket and to write to the DynamoDB table, During testing, a DevOps engineer discovers that the Lambda

function does not run when objects are added to the S3 bucket or when existing objects are modified.

Which solution will resolve this problem?

A. Increase the memory of the Lambda function to give the function the ability to process large files from the S3 bucket.

B. Create a resource policy on the Lambda function to grant Amazon S3 the permission to invoke the Lambda function for the S3 bucket

C. Configure an Amazon Simple Queue Service (Amazon SQS) queue as an OnFailure destination for the Lambda function

D. Provision space in the /tmp folder of the Lambda function to give the function the ability to process large files from the S3 bucket

ANSWER: B

Explanation:

Create a resource policy on the Lambda function to grant Amazon S3 the permission to invoke the Lambda function for the S3 bucket. Amazon S3 event notifications invoke Lambda asynchronously, but S3 must first be explicitly authorized by a resource-based policy attached to the target Lambda function. The Lambda execution role governs what the function code can do after it starts, such as calling `s3:GetObject` and writing records to DynamoDB. It does not grant an AWS service permission to invoke the function.

The required Lambda resource-policy permission uses the `lambda:AddPermission` action and identifies `s3.amazonaws.com` as the principal. It should also restrict the permission with the source S3 bucket ARN, and commonly the owning AWS account, to ensure that only notifications from the intended bucket can invoke the function. When configured through the S3 console or AWS CLI, AWS validates the destination configuration; this invoke permission is a required part of a valid S3-to-Lambda notification setup. After the permission and the bucket notification configuration are in place, object-created event types can trigger the function and provide the bucket and object-key details needed for processing. See Using AWS Lambda with Amazon S3 and Lambda AddPermission API.

QUESTION NO: 29

A company has an application that runs on a fleet of Amazon EC2 instances. The application requires frequent restarts. The application logs contain error messages when a restart is required. The application logs are published to a log group in Amazon CloudWatch Logs.

An Amazon CloudWatch alarm notifies an application engineer through an Amazon Simple Notification Service (Amazon SNS) topic when the logs contain a large number of restart-related error messages. The application engineer manually restarts the application on the instances after the application engineer receives a notification from the SNS topic.

A DevOps engineer needs to implement a solution to automate the application restart on the instances without restarting the instances.

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

A. Configure an AWS Systems Manager Automation runbook that runs a script to restart the application on the instances. Configure the SNS topic to invoke the runbook.

B. Create an AWS Lambda function that restarts the application on the instances. Configure the Lambda function as an event destination of the SNS topic.

C. Configure an AWS Systems Manager Automation runbook that runs a script to restart the application on the instances. Create an AWS Lambda function to invoke the runbook. Configure the Lambda function as an event destination of the SNS topic.

D. Configure an AWS Systems Manager Automation runbook that runs a script to restart the application on the instances. Configure an Amazon EventBridge rule that reacts when the CloudWatch alarm enters ALARM state. Specify the runbook as a target of the rule.

ANSWER: D

Explanation:

Configure an AWS Systems Manager Automation runbook that runs a script to restart the application on the instances. Configure an Amazon EventBridge rule that reacts when the CloudWatch alarm enters ALARM state. Specify the runbook as a target of the rule.

This approach creates a direct, managed event-driven remediation path. CloudWatch alarm state changes are emitted as events to Amazon EventBridge, so a rule can match the transition to the `ALARM` state and initiate the Systems Manager Automation runbook. The runbook can use Systems Manager capabilities to run the required application-level restart command on the managed EC2 instances, without rebooting or otherwise restarting those instances.

Using EventBridge as the integration point avoids relying on a human response to the SNS notification and does not require custom event-processing code merely to bridge the alarm to the remediation workflow. Systems Manager Automation also provides an auditable execution history and supports controlled operational actions through a defined runbook. The EventBridge target must have an IAM role that permits starting the Automation execution, and the instances must be configured as Systems Manager managed nodes with the permissions needed for the runbook action. See the CloudWatch documentation for EventBridge alarm events and the EventBridge targets documentation.

QUESTION NO: 30

A company wants to use AWS development tools to replace its current bash deployment scripts. The company currently deploys a LAMP application to a group of Amazon EC2 instances behind an Application Load Balancer (ALB). During the deployments, the company unit tests the committed application, stops and starts services, unregisters and re-registers instances with the load balancer, and updates file permissions. The company wants to maintain the same deployment functionality through the shift to using AWS services.

Which solution will meet these requirements?

A. Use AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy's appspec.yml file to restart services, and deregister and register instances with the ALB. Use the appspec.yml file to update file permissions without a custom script.

B. Use AWS CodePipeline to move the application from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy's deployment group to test the application, unregister and re-register instances with the ALB, and restart services. Use the appspec.yml file to update file permissions without a custom script.

C. Use AWS CodePipeline to move the application source code from the AWS CodeCommit repository to AWS CodeDeploy. Use CodeDeploy to test the application. Use CodeDeploy's appspec.yml file to restart services and update permissions without a custom script. Use AWS CodeBuild to unregister and re-register instances with the ALB.

D. Use AWS CodePipeline to trigger AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy's appspec.yml file to restart services. Unregister and re-register the instances in the AWS CodeDeploy deployment group with the ALB. Update the appspec.yml file to update file permissions without a custom script.

ANSWER: D

Explanation:

Use AWS CodePipeline to trigger AWS CodeBuild to test the application. Use bash scripts invoked by AWS CodeDeploy's appspec.yml file to restart services. Unregister and re-register the instances in the AWS CodeDeploy deployment group with the ALB. Update the appspec.yml file to update file permissions without a custom script. This solution assigns each deployment activity to the AWS service designed for it while preserving the existing operational behavior. CodePipeline orchestrates the workflow, including invoking CodeBuild to run the unit tests before deployment proceeds. CodeDeploy deploys the revision to the EC2 fleet and supports load balancer integration for a deployment group. With that integration, CodeDeploy takes instances out of service during deployment and returns them to the ALB after the deployment lifecycle completes, helping avoid routing production traffic to instances while they are being updated. The AppSpec file supports lifecycle event hooks that invoke scripts on the target instances, which is appropriate for stopping and starting LAMP services. For EC2/On-Premises deployments, the AppSpec file also has a permissions section that can set ownership and mode permissions on deployed files without requiring a separate custom permissions script. This produces a repeatable pipeline with testing, traffic management, service control, and file-permission configuration. See the CodeDeploy load balancer integration documentation and the AppSpec permissions reference.

QUESTION NO: 31

A growing company manages more than 50 accounts in an organization in AWS Organizations. The company has configured its applications to send logs to Amazon CloudWatch Logs.

A DevOps engineer needs to aggregate logs so that the company can quickly search the logs to respond to future security incidents. The DevOps engineer has created a new AWS account for centralized monitoring.

Which combination of steps should the DevOps engineer take to make the application logs searchable from the monitoring account? (Select THREE.)

A. In the monitoring account, download an AWS CloudFormation template from CloudWatch to use in Organizations. Use CloudFormation StackSets in the organization's management account to deploy the CloudFormation template to the entire organization.

B. Create an AWS CloudFormation template that defines an IAM role. Configure the role to allow logs.amazonaws.com to perform the logs:Link action if the aws:ResourceAccount property is equal to the monitoring account ID. Use CloudFormation StackSets in the organization's management account to deploy the CloudFormation template to the entire organization.

C. Create an IAM role in the monitoring account that allows logs.amazonaws.com to perform the logs:CreateSink action. Attach a trust policy that permits assumption by logs.amazonaws.com only if the aws:PrincipalOrgID property is equal to the organization ID.

D. In the organization's management account, enable the logging policies for the organization.

E. Use CloudWatch Observability Access Manager in the monitoring account to create a sink. Allow logs to be shared with the monitoring account. Configure the monitoring account data selection to view the Observability data from the organization ID.

F. In the monitoring account, attach the CloudWatchLogsReadOnlyAccess AWS managed policy to an IAM role that can be assumed to search the logs.

ANSWER: B C F

Explanation:

CloudWatch Logs centralized logging uses service roles to establish organization-wide log sharing between source accounts and a monitoring account. In each source account, CloudWatch Logs must be able to assume a role that grants `logs:Link` permission for linking that account's log groups to the centralized destination. Deploying this role through AWS CloudFormation StackSets is an appropriate operational approach for more than 50 accounts because it applies the consistent role configuration across the organization.

The monitoring account also needs a role that CloudWatch Logs can assume to create the centralized sink. Restricting its trust relationship with the `aws:PrincipalOrgID` condition ensures that only accounts belonging to the company's AWS Organization can participate. After log data is centralized, personnel need an assumable role with read-only CloudWatch Logs permissions to query, inspect, and investigate the aggregated logs. The AWS managed `CloudWatchLogsReadOnlyAccess` policy provides the required read access without granting log modification permissions. These roles collectively enable the CloudWatch Logs service to centralize data and authorized incident responders to search it from the monitoring account. See Centralizing CloudWatch Logs and the CloudWatchLogsReadOnlyAccess policy reference.

QUESTION NO: 32

A company has multiple member accounts that are part of an organization in AWS Organizations. The security team needs to review every Amazon EC2 security group and their inbound and outbound rules. The security team wants to programmatically retrieve this information from the member accounts using an AWS Lambda function in the management account of the organization.

Which combination of access changes will meet these requirements? (Choose three.)

A. Create a trust relationship that allows users in the member accounts to assume the management account IAM role.

B. Create a trust relationship that allows users in the management account to assume the IAM roles of the member accounts.

C. Create an IAM role in each member account that has access to the AmazonEC2ReadOnlyAccess managed policy.

D. Create an I AM role in each member account to allow the sts:AssumeRole action against the management account IAM role's ARN.

E. Create an I AM role in the management account that allows the sts:AssumeRole action against the member account IAM role's ARN.

F. Create an IAM role in the management account that has access to the AmazonEC2ReadOnlyAccess managed policy.

ANSWER: B C E

Explanation:

Cross-account access from a Lambda function in the management account requires a role in each member account that Lambda can assume, plus permissions on both sides of that role assumption. "Create a trust relationship that allows users in the management account to assume the IAM roles of the member accounts" supplies the member-account role trust policy. In practice, the trusted principal should be the IAM role used as the Lambda execution role in the management account, allowing AWS Security Token Service to issue temporary credentials for the member account.

"Create an IAM role in each member account that has access to the AmazonEC2ReadOnlyAccess managed policy" supplies the permissions needed after assumption. This AWS managed policy includes EC2 Describe actions, including the calls used to retrieve security groups and their ingress and egress rule configuration. "Create an I AM role in the management account that allows the sts:AssumeRole action against the member account IAM role's ARN" supplies the identity-based permission for the Lambda execution role to call `sts:AssumeRole` on each target role. Lambda can then iterate through the member accounts, assume each role, and use the temporary credentials to call the relevant EC2 Describe APIs. This follows AWS's standard cross-account IAM-role model, which requires both a trusted principal in the target role's trust policy and permission for that principal to assume the target role. See AWS IAM cross-account role access and AmazonEC2ReadOnlyAccess policy reference.

QUESTION NO: 33

A DevOps engineer needs to design a cloud-based solution to standardize deployment artifacts for AWS Cloud deployments and on-premises deployments. There is currently no routing traffic between the on-premises data center and the AWS environment.

The solution must be able to consume downstream packages from public repositories and must be highly available. Data must be encrypted in transit and at rest. The solution must store the deployment artifacts in object storage and deploy the deployment artifacts into Amazon Elastic Container Service (Amazon ECS). The deployment artifacts must be encrypted in transit if the deployment artifacts travel across the public internet.

The DevOps engineer needs to deploy this solution in less than two weeks.

Which solution will meet these requirements?

A. Use a third-party software VPN appliance to connect the on-premises data center and AWS. Use AWS CodeArtifact to store the deployment artifacts.

B. Use an AWS Direct Connect connection and a VPN connection to connect the on-premises data center to AWS. Deploy third-party artifact management software on Amazon EC2 instances.

C. Use two AWS VPN connections to connect the on-premises data center to AWS. Use AWS CodeArtifact to store the deployment artifacts.

D. Use parallel AWS Direct Connect connections to connect the on-premises data center to AWS. Deploy third-party artifact management software on Amazon EC2 instances.

ANSWER: C

Explanation:

Use two AWS VPN connections to connect the on-premises data center to AWS. Use AWS CodeArtifact to store the deployment artifacts. AWS CodeArtifact is a fully managed repository service for software packages, removing the time and operational effort of deploying, maintaining, scaling, and making third-party artifact-management servers highly available. It can connect repositories to supported public package repositories through an external connection, allowing internal teams to consume required upstream packages while using a centralized repository. CodeArtifact stores package assets in AWS-managed durable storage and encrypts assets at rest; communications with the service use TLS for encryption in transit. Workloads and deployment processes in AWS, including those that deploy to Amazon ECS, can retrieve the required packages from this central repository.

Two AWS Site-to-Site VPN connections provide redundant encrypted network paths between the data center and AWS without waiting for Direct Connect provisioning. VPN traffic that traverses the public internet is protected by IPsec encryption, meeting the transit-encryption requirement. AWS recommends redundant VPN connectivity for resilient hybrid connectivity, and this managed approach can be implemented within the stated timeline. Together, managed artifact storage with public-repository upstream access and redundant VPN connectivity satisfies the availability, encryption, hybrid-access, and rapid-deployment requirements. See the AWS CodeArtifact external connections documentation and AWS Site-to-Site VPN redundant connection guidance.

QUESTION NO: 34

A company uses Amazon Redshift as its data warehouse solution. The company wants to create a dashboard to view changes to the Redshift users and the queries the users perform.

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

A. Create an Amazon CloudWatch log group. Create an AWS CloudTrail trail that writes to the CloudWatch log group.

B. Create a new Amazon S3 bucket. Configure default audit logging on the Redshift cluster. Configure the S3 bucket as the target.

C. Configure the Redshift cluster database audit logging to include user activity logs. Configure Amazon CloudWatch as the target.

D. Create an Amazon CloudWatch dashboard that has a log widget. Configure the widget to display user details from the Redshift logs.

E. Create an AWS Lambda function that uses Amazon Athena to query the Redshift logs. Create an Amazon CloudWatch dashboard that has a custom widget type that uses the Lambda function.

ANSWER: C D

Explanation:

“Configure the Redshift cluster database audit logging to include user activity logs. Configure Amazon CloudWatch as the target.” is correct because Amazon Redshift audit logging captures the information needed for this use case. Redshift produces a user log that records changes to database-user definitions, such as user creation, alteration, and deletion. It also produces a user activity log that records queries run on the database. Redshift can export these audit logs directly to Amazon CloudWatch Logs, where the log streams can be searched and queried with CloudWatch Logs Insights.

“Create an Amazon CloudWatch dashboard that has a log widget. Configure the widget to display user details from the Redshift logs.” is correct because a CloudWatch dashboard log widget can display the results of a CloudWatch Logs Insights query. Queries against the Redshift user-log and user-activity-log data can filter, aggregate, and present user changes and query activity directly on the operational dashboard. This provides a managed, near-real-time visualization path without requiring a separate data-processing workflow. See the Amazon Redshift audit logging documentation and the CloudWatch Logs Insights dashboard widget documentation.

QUESTION NO: 35

A DevOps team is merging code revisions for an application that uses an Amazon RDS Multi-AZ DB cluster for its production database. The DevOps team uses continuous integration to periodically verify that the application works. The DevOps team needs to test the changes before the changes are deployed to the production database.

Which solution will meet these requirements ' ?

A. Use a buildspec file in AWS CodeBuild to restore the DB cluster from a snapshot of the production database, run integration tests, and drop the restored database after verification.

B. Deploy the application to production. Configure an audit log of data control language (DCL) operations to capture database activities to perform if verification fails.

C. Create a snapshot of the DB cluster before deploying the application. Use the Update requires Replacement property on the DB instance in AWS CloudFormation to deploy the application and apply the changes.

D. Ensure that the DB cluster is a Multi-AZ deployment. Deploy the application with the updates. Fail over to the standby instance if verification fails.

ANSWER: A

Explanation:

Use a buildspec file in AWS CodeBuild to restore the DB cluster from a snapshot of the production database, run integration tests, and drop the restored database after verification. This creates an isolated, temporary database environment that closely reflects production data and configuration, allowing the continuous integration workflow to validate both the application revisions and their database interactions before any production deployment occurs. A CodeBuild buildspec can define the commands and phases needed to invoke AWS service APIs or the AWS CLI, wait for the restored cluster to become available, run the application’s integration test suite, and perform cleanup regardless of test outcome. Restoring from an RDS snapshot is especially appropriate when tests require realistic schema and data rather than an empty development database. The restored cluster must use a new identifier, so production remains untouched throughout verification. Automating deletion of the temporary restored resources after testing also prevents unnecessary ongoing database costs and keeps the CI environment ephemeral. Amazon RDS documents that DB cluster snapshots can be restored to create a new DB cluster, and CodeBuild documents buildspec files as the mechanism for defining build commands and test phases. See Restoring from a DB snapshot and Build specification reference for CodeBuild.

QUESTION NO: 36

A company ' s application has an API that retrieves workload metrics. The company needs to audit, analyze, and visualize these metrics from the application to detect issues at scale.

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

A. Configure an Amazon EventBridge schedule to invoke an AWS Lambda function that calls the API to retrieve workload metrics. Store the workload metric data in an Amazon S3 bucket.

B. Configure an Amazon EventBridge schedule to invoke an AWS Lambda function that calls the API to retrieve workload metrics. Store the workload metric data in an Amazon DynamoDB table that has a DynamoDB stream enabled.

C. Create an AWS Glue crawler to catalog the workload metric data in the Amazon S3 bucket. Create views in Amazon Athena for the cataloged data.

D. Connect an AWS Glue crawler to the Amazon DynamoDB stream to catalog the workload metric data. Create views in Amazon Athena for the cataloged data.

E. Create Amazon QuickSight datasets from the Amazon Athena views. Create a QuickSight analysis to visualize the workload metric data as a dashboard.

F. Create an Amazon CloudWatch dashboard that has custom widgets that invoke AWS Lambda functions. Configure the Lambda functions to query the workload metrics data from the Amazon Athena views.

ANSWER: A C E

Explanation:

The correct architecture uses a serverless data lake and analytics workflow. “Configure an Amazon EventBridge schedule to invoke an AWS Lambda function that calls the API to retrieve workload metrics. Store the workload metric data in an Amazon S3 bucket.” provides automated, recurring collection of the application’s API-based metrics and durable, scalable storage for historical auditing. EventBridge Scheduler or scheduled EventBridge rules can invoke Lambda on a schedule, while S3 is well suited to retaining large volumes of metric records for later analysis.

“Create an AWS Glue crawler to catalog the workload metric data in the Amazon S3 bucket. Create views in Amazon Athena for the cataloged data.” makes the S3 data discoverable through the AWS Glue Data Catalog and queryable with Athena using SQL. Athena views can present curated queries and simplify access to the relevant metric dimensions, time ranges, and aggregations.

“Create Amazon QuickSight datasets from the Amazon Athena views. Create a QuickSight analysis to visualize the workload metric data as a dashboard.” completes the solution by connecting a managed BI service to the queried data and publishing dashboards for operational investigation and trend detection at scale. This separation of collection, storage, cataloging, querying, and visualization supports independent scaling and low operational overhead. See the Amazon Athena documentation for the AWS Glue Data Catalog and the Amazon QuickSight documentation for Athena data sources.

QUESTION NO: 37

An ecommerce company hosts a web application on Amazon EC2 instances that are in an Auto Scaling group. The company deploys the application across multiple Availability Zones.

Application users are reporting intermittent performance issues with the application.

The company enables basic Amazon CloudWatch monitoring for the EC2 instances. The company identifies and implements a fix for the performance issues. After resolving the issues, the company wants to implement a monitoring solution that will quickly alert the company about future performance issues.

Which solution will meet this requirement?

A. Enable detailed monitoring for the EC2 instances. Create custom CloudWatch metrics for application-specific performance indicators. Set up CloudWatch alarms based on the custom metrics. Use CloudWatch Logs Insights to analyze application logs for error patterns.

B. Use AWS X-Ray to implement distributed tracing. Integrate X-Ray with Amazon CloudWatch RUM. Use Amazon EventBridge to trigger automatic scaling actions based on custom events.

C. Use Amazon CloudFront to deliver the application. Use AWS CloudTrail to monitor API calls. Use AWS Trusted Advisor to generate recommendations to optimize performance. Use Amazon GuardDuty to detect potential performance issues.

D. Enable VPC Flow Logs. Use Amazon Data Firehose to stream flow logs to Amazon S3. Use Amazon Athena to analyze the logs and to send alerts to the company.

ANSWER: A

Explanation:

“Enable detailed monitoring for the EC2 instances. Create custom CloudWatch metrics for application-specific performance indicators. Set up CloudWatch alarms based on the custom metrics. Use CloudWatch Logs Insights to analyze application logs for error patterns.” is the appropriate solution because it combines timely infrastructure monitoring with direct visibility into the application behavior that users experience. Detailed monitoring changes the resolution of Amazon EC2 metrics from five-minute intervals to one-minute intervals, allowing alarms to detect sustained resource pressure more quickly. Custom Amazon CloudWatch metrics let the company measure meaningful application signals, such as request latency, error counts, failed transactions, queue depth, or response-time percentiles. CloudWatch alarms can evaluate these metrics against operational thresholds and promptly notify an incident-response channel through Amazon SNS or an integrated alerting system. CloudWatch Logs Insights complements the alarms by enabling engineers to rapidly query application logs for errors, timeouts, and recurring patterns during an incident. This approach is suitable for an Auto Scaling, multi-Availability Zone deployment because metrics and alarms can aggregate behavior across the fleet while preserving instance-level diagnostic detail. AWS documents detailed monitoring at Amazon EC2 monitoring and alarm configuration at Using Amazon CloudWatch alarms.

QUESTION NO: 38

A company uses AWS WAF to protect its cloud infrastructure. A DevOps engineer needs to give an operations team the ability to analyze log messages from AWS WAR. The operations team needs to be able to create alarms for specific patterns in the log output.

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

A. Create an Amazon CloudWatch Logs log group. Configure the appropriate AWS WAF web ACL to send log messages to the log group. Instruct the operations team to create CloudWatch metric filters.

B. Create an Amazon OpenSearch Service cluster and appropriate indexes. Configure an Amazon Kinesis Data Firehose delivery stream to stream log data to the indexes. Use OpenSearch Dashboards to create filters and widgets.

C. Create an Amazon S3 bucket for the log output. Configure AWS WAF to send log outputs to the S3 bucket. Instruct the operations team to create AWS Lambda functions that detect each desired log message pattern. Configure the Lambda functions to publish to an Amazon Simple Notification Service (Amazon SNS) topic.

D. Create an Amazon S3 bucket for the log output. Configure AWS WAF to send log outputs to the S3 bucket. Use Amazon Athena to create an external table definition that fits the log message pattern. Instruct the operations team to write SOL queries and to create Amazon CloudWatch metric filters for the Athena queries.

ANSWER: A

Explanation:

Create an Amazon CloudWatch Logs log group. Configure the appropriate AWS WAF web ACL to send log messages to the log group. Instruct the operations team to create CloudWatch metric filters. AWS WAF logging can deliver web ACL logs directly to an Amazon CloudWatch Logs log group, providing a managed, near-real-time destination that the operations team can query and inspect without operating ingestion infrastructure or writing custom processing code. The log group name must begin with the `aws-waf-logs-` prefix when it is configured as an AWS WAF logging destination. CloudWatch Logs metric filters evaluate incoming log events against defined filter patterns and publish matching-event counts as CloudWatch metrics. The team can then create CloudWatch alarms on those metrics, such as alarming when blocked requests, particular rule actions, or suspicious request characteristics exceed a threshold during a chosen evaluation period. This directly connects AWS WAF log delivery, pattern detection, metric generation, and alarm notification through managed AWS services, minimizing operational work while satisfying both log-analysis and alerting needs. See the AWS WAF documentation for logging to CloudWatch Logs and the CloudWatch Logs metric filter documentation.

QUESTION NO: 39

A company wants to improve its security practices by enforcing least privilege across all projects. Developers must be able to access Amazon EC2 resources but not Amazon RDS resources. Database administrators must have access only to Amazon RDS resources.

Every employee has a unique IAM user. There are already pre-existing IAM policies for developer and database administrator job functions. All AWS resources are already tagged with appropriate project tags. All the IAM users are tagged with the appropriate project and job function.

The company must ensure that each employee can access only the project that the employee is working on.

Which solution will meet these requirements? (Select THRE

E.

)

A. For each project, create one IAM role for developers and one IAM role for database administrators. Tag the IAM roles with the corresponding projects and job functions.

B. Modify the pre-existing IAM policies to include a StringEquals condition that compares the ResourceTag for projects with the PrincipalTag value. Attach the modified policies to the IAM roles for each job function.

C. Create an IAM policy that allows users to assume a role only when the ResourceTag values match the PrincipalTag values for project tags and job function tags. Attach the new policy to all IAM users.

D. Create an IAM policy that allows users to assume a role only when the ResourceTag values match the PrincipalTag values for project tags and job function tags. Attach the new policy to the IAM roles for each job function.

E. Tag the pre-existing IAM policies with the appropriate projects and job functions. Attach the modified policies to IAM roles for each job function.

F. For each project, create one IAM group for developers and one IAM group for database administrators. Add the appropriate users to each group so the users can assume their respective IAM roles.

ANSWER: A B C

Explanation:

This solution uses IAM attribute-based access control (ABAC) to enforce both job-function permissions and project isolation. Creating separate IAM roles for developers and database administrators in each project provides a role resource that can carry project and job-function tags. The pre-existing job-function policies can then continue to grant only the relevant service permissions: developer permissions for Amazon EC2 and database-administrator permissions for Amazon RDS.

Adding a `StringEquals` condition that compares `aws:ResourceTag/Project` to `aws:PrincipalTag/Project` ensures that permissions apply only when the tagged resource belongs to the same project as the requesting employee. Because IAM evaluates principal tags from the tagged IAM user and resource tags from the requested AWS resource, this avoids maintaining separate permissions policies for every employee or project.

Finally, allowing `sts:AssumeRole` only when the IAM role's resource tags match the user's principal tags for both project and job function prevents a user from assuming a role intended for another project or job function. Together, these controls implement scalable least privilege by combining role-based service access with tag-based authorization. AWS documents this ABAC pattern and the global condition keys used for principal and resource tags in the IAM ABAC guide and IAM condition key reference.

QUESTION NO: 40

A company runs applications in AWS accounts that are in an organization in AWS Organizations The applications use Amazon EC2 instances and Amazon S3.

The company wants to detect potentially compromised EC2 instances suspicious network activity and unusual API activity in its existing AWS accounts and in any AWS accounts that the company creates in the future When the company detects one

to these events the company wants to use an existing Amazon Simple Notification Service (Amazon SNS) topic to send a notification to its operational support team for investigation and remediation.

Which solution will meet these requirements in accordance with AWS best practices?

A. In the organization ' s management account configure an AWS account as the Amazon GuardDuty administrator account. In the GuardDuty administrator account add the company ' s existing AWS accounts to GuardDuty as members In the GuardDuty administrator account create an Amazon EventBridge rule with an event pattern to match GuardDuty events and to forward matching events to the SNS topic.

B. In the organization ' s management account configure Amazon GuardDuty to add newly created AWS accounts by invitation and to send invitations to the existing AWS accounts Create an AWS Cloud Formation stack set that accepts the GuardDuty invitation and creates an Amazon EventBridge rule Configure the rule with an event pattern to match. GuardDuty events and to forward matching events to the SNS topic. Configure the Cloud Formation stack set

C. In the organization ' s management account. create an AWS CloudTrail organization trail Activate the organization trail in all AWS accounts in the organization. Create an SCP that enables VPC Flow Logs in each account in the organization. Configure AWS Security Hub for the organization Create an Amazon EventBridge rule with an even pattern to match Security Hub events and to forward matching events to the SNS topic.

D. In the organization ' s management account configure an AWS account as the AWS CloudTrail administrator account in the CloudTrail administrator account create a CloudTrail organization trail. Add the company ' s existing AWS accounts to the organization trail Create an SCP that enables VPC Flow Logs in each account in the organization. Configure AWS Security Hub for the organization. Create an Amazon EventBridge rule with an event pattern t

E. Designate a GuardDuty delegated administrator, enable GuardDuty for the organization with automatic enrollment of new accounts, and enable the required protection plans. Create an EventBridge rule in the delegated administrator account that matches GuardDuty findings and targets the existing SNS topic.

ANSWER: E

Explanation:

Configure an AWS account as the Amazon GuardDuty delegated administrator, enable GuardDuty organization-wide, and configure automatic enrollment for new organization accounts. GuardDuty is the appropriate managed threat-detection service because it continuously analyzes AWS CloudTrail management events, VPC Flow Logs, DNS logs, and other security telemetry to produce findings for potentially compromised EC2 instances, suspicious network behavior, and anomalous API activity. Designating a delegated administrator centralizes management and visibility of findings from member accounts while avoiding use of the Organizations management account for daily security operations.

Automatic enrollment is essential: it ensures that accounts created or added to the organization are covered without requiring invitations, individual account actions, or repeated deployment of account-level configuration. The delegated administrator can also enable appropriate GuardDuty protection plans for organization accounts. GuardDuty findings are delivered to Amazon EventBridge, so an EventBridge rule in the delegated administrator account can match GuardDuty finding events and use the existing SNS topic as its target. This provides prompt operational notification while retaining a centralized, scalable detection and response architecture. Configure the solution in each AWS Region where the organization runs workloads, because GuardDuty is a Regional service. See the Amazon GuardDuty Organizations documentation and the GuardDuty findings in Amazon EventBridge documentation.

QUESTION NO: 41

A company is performing vulnerability scanning for all Amazon EC2 instances across many accounts. The accounts are in an organization in AWS Organizations. Each account's VPCs are attached to a shared transit gateway. The VPCs send traffic to the internet through a central egress VPC. The company has enabled Amazon Inspector in a delegated administrator account and has enabled scanning for all member accounts.

A DevOps engineer discovers that some EC2 instances are listed in the "not scanning" tab in Amazon Inspector.

Which combination of actions should the DevOps engineer take to resolve this issue? (Choose three.)

A. Verify that AWS Systems Manager Agent is installed and is running on the EC2 instances that Amazon Inspector is not scanning.

B. Associate the target EC2 instances with security groups that allow outbound communication on port 443 to the AWS Systems Manager service endpoint.

C. Grant inspector:StartAssessmentRun permissions to the IAM role that the DevOps engineer is using.

D. Configure EC2 Instance Connect for the EC2 instances that Amazon Inspector is not scanning.

E. Associate the target EC2 instances with instance profiles that grant permissions to communicate with AWS Systems Manager.

F. Create a managed-instance activation. Use the Activation Code and the Activation ID to register the EC2 instances.

ANSWER: A B E

Explanation:

Amazon Inspector uses AWS Systems Manager to perform agent-based EC2 scanning where required. Therefore, the affected instances must have the AWS Systems Manager Agent installed and running so that they can be managed by Systems Manager and participate in Inspector scanning. The instances also need network connectivity to the Systems Manager service endpoints over HTTPS. In this centrally routed environment, allowing outbound communication on port 443 to Systems Manager endpoints through the central egress path satisfies that connectivity requirement; interface VPC endpoints can also be used where appropriate. Finally, each instance needs an attached instance profile whose IAM role grants the Systems Manager permissions required to register and communicate as a managed node. AWS recommends the `AmazonSSMManagedInstanceCore` managed policy for this purpose. Once the agent, HTTPS endpoint access, and instance-profile permissions are in place, Systems Manager can manage the instances and Amazon Inspector can begin or resume vulnerability scanning. Amazon Inspector's EC2 scanning prerequisites and troubleshooting guidance identify Systems Manager management, agent health, and connectivity as essential conditions for scanning. See Amazon Inspector EC2 scanning and Systems Manager instance permissions.

QUESTION NO: 42

A company has set up AWS CodeArtifact repositories with public upstream repositories The company ' s development team consumes open source dependencies from the repositories in the company ' s internal network.

The company ' s security team recently discovered a critical vulnerability in the most recent version of a package that the development team consumes. The security team has produced a patched version to fix the vulnerability. The company needs to prevent the vulnerable version from being downloaded. The company also needs to allow the security team to publish the patched version.

Which combination of steps will meet these requirements? {Select TWO.)

A. Update the status of the affected CodeArtifact package version to unlisted

B. Update the status of the affected CodeArtifact package version to deleted

C. Update the status of the affected CodeArtifact package version to archived.

D. Update the CodeArtifact package origin control settings to allow direct publishing and to block upstream operations

E. Update the CodeArtifact package origin control settings to block direct publishing and to allow upstream operations.

ANSWER: B D

Explanation:

"Update the status of the affected CodeArtifact package version to deleted" prevents the vulnerable version from being retrieved from the CodeArtifact repository. A deleted package version is not available for normal package-manager download requests, which removes the compromised version from developers' dependency resolution path. This is the appropriate response when the organization must ensure that a known-vulnerable artifact can no longer be consumed.

"Update the CodeArtifact package origin control settings to allow direct publishing and to block upstream operations" enables the required remediation workflow. Direct publishing must be allowed so the security team can publish its patched package version into the internal CodeArtifact repository. Blocking upstream operations prevents CodeArtifact from obtaining package

versions from an upstream public repository for that package, ensuring the organization controls the package source and avoids reintroducing an externally available vulnerable version through upstream package retrieval. Package origin controls can independently govern direct publishing and upstream ingestion behavior, allowing this combination of internal publication and upstream restriction. See the AWS CodeArtifact package origin controls documentation and the AWS guidance for deleting package versions.

QUESTION NO: 43

A company deploys an API by using an Application Load Balancer (ALB) that targets an AWS Lambda function. The API takes a list of tasks as an input and typically processes the tasks in 40 seconds. The API is CPU intensive and performs writes to an Amazon Aurora PostgreSQL DB cluster. The API also performs frequent large-scale Amazon S3 object PUT requests to the same prefix of an S3 bucket.

The API usage and the number of input tasks is increasing significantly. The increased demand is causing the API response time to increase, and some API requests fail because the database connection quota is being exceeded. A DevOps engineer must improve the performance of the API.

Which combination of solutions will meet this requirement? (Select THREE.)

A. Deploy a Network Load Balancer (NLB) and set the ALB as the target group for the NLB.

B. Refactor the API to use Amazon API Gateway instead of the ALB.

C. Create an Amazon RDS Proxy and configure the Lambda function to use it to perform database operations.

D. Increase the amount of memory for the Lambda function.

E. Add a read replica for the Aurora DB cluster.

F. Update the Lambda function to distribute S3 object PUT requests to multiple prefixes.

ANSWER: C D F

Explanation:

Creating an Amazon RDS Proxy and configuring the Lambda function to use it for database operations addresses the Aurora PostgreSQL connection-quota failures. Lambda can scale quickly by creating concurrent execution environments, each of which might otherwise establish database connections. RDS Proxy pools and reuses database connections, reducing connection-management overhead and protecting the database from connection storms while allowing the application to scale more reliably. AWS describes RDS Proxy as fully managed connection pooling for applications that use Amazon RDS and Aurora databases.

Increasing the Lambda function's memory improves this CPU-intensive API because Lambda allocates CPU capacity in proportion to the configured memory. More memory therefore supplies more compute resources and can reduce the duration of task processing, provided the function is tested and sized appropriately.

Updating the function to distribute Amazon S3 PUT requests across multiple prefixes increases available write throughput. Amazon S3 automatically scales request handling, and each prefix can support thousands of write requests per second. Using multiple prefixes distributes this high-volume write workload and avoids concentrating requests on a single logical prefix. See Amazon RDS Proxy documentation and Amazon S3 performance optimization guidance.

QUESTION NO: 44

A DevOps engineer notices that all Amazon EC2 instances running behind an Application Load Balancer in an Auto Scaling group are failing to respond to user requests. The EC2 instances are also failing target group HTTP health checks

Upon inspection, the engineer notices the application process was not running in any EC2 instances. There are a significant number of out of memory messages in the system logs. The engineer needs to improve the resilience of the application to cope with a potential application memory leak. Monitoring and notifications should be enabled to alert when there is an issue

Which combination of actions will meet these requirements? (Select TWO.)

A. Change the Auto Scaling configuration to replace the instances when they fail the load balancer ' s health checks.

B. Change the target group health check HealthCheckIntervalSeconds parameter to reduce the interval between health checks.

C. Change the target group health checks from HTTP to TCP to check if the port where the application is listening is reachable.

D. Enable the available memory consumption metric within the Amazon CloudWatch dashboard for the entire Auto Scaling group Create an alarm when the memory utilization is high Associate an Amazon SNS topic to the alarm to receive notifications when the alarm goes off

E. Use the Amazon CloudWatch agent to collect the memory utilization of the EC2 instances in the Auto Scaling group Create an alarm when the memory utilization is high and associate an Amazon SNS topic to receive a notification.

ANSWER: A E

Explanation:

Changing the Auto Scaling configuration to replace instances when they fail the load balancer's health checks provides automated recovery when an instance's application is no longer able to serve requests. An Auto Scaling group can use Elastic Load Balancing health checks in addition to EC2 status checks. When the Application Load Balancer reports a target as unhealthy because its HTTP health check fails, the Auto Scaling group can mark that instance unhealthy, terminate it, and launch a replacement instance. This restores healthy capacity without requiring manual intervention, which is particularly important when a memory leak causes the application process to stop.

Using the Amazon CloudWatch agent to collect memory utilization from the EC2 instances, then creating a high-memory alarm that publishes to an Amazon SNS topic, supplies the required early-warning capability. Amazon EC2 does not provide guest operating system memory utilization as a default CloudWatch metric; the CloudWatch agent collects memory metrics such as `mem_used_percent`. An alarm can evaluate that metric for each affected instance and notify subscribers through SNS before memory exhaustion causes application failure. Relevant AWS guidance is available in Amazon EC2 Auto Scaling health checks and CloudWatch agent collected metrics.

QUESTION NO: 45

A company requires an RPO of 2 hours and an RTO of 10 minutes for its data and application at all times. An application uses a MySQL database and Amazon EC2 web servers. The development team needs a strategy for failover and disaster recovery.

Which combination of deployment strategies will meet these requirements? (Select TWO.)

A. Create an Amazon Aurora cluster in one Availability Zone across multiple Regions as the data store Use Aurora ' s automatic recovery capabilities in the event of a disaster

B. Create an Amazon Aurora global database in two Regions as the data store. In the event of a failure promote the secondary Region as the primary for the application.

C. Create an Amazon Aurora multi-master cluster across multiple Regions as the data store. Use a Network Load Balancer to balance the database traffic in different Regions.

D. Set up the application in two Regions and use Amazon Route 53 failover-based routing that points to the Application Load Balancers in both Regions. Use hearth checks to determine the availability in a given Region. Use Auto Scaling groups in each Region to adjust capacity based on demand.

E. Set up the application m two Regions and use a multi-Region Auto Scaling group behind Application Load Balancers to manage the capacity based on demand. In the event of a disaster adjust the Auto Scaling group ' s desired instance count to increase baseline capacity in the failover Region.

ANSWER: B D

Explanation:

Creating an Amazon Aurora global database in two Regions as the data store and promoting the secondary Region during a failure provides a cross-Region database disaster-recovery design for the MySQL workload. Aurora Global Database replicates data from the primary Region to secondary Regions with typical replication latency of less than one second. This is substantially within the required two-hour recovery point objective. In a regional outage, the secondary cluster can be promoted to become the read/write primary cluster. Aurora Global Database is designed for recovery from a Region-wide outage, with managed failover procedures that can meet a ten-minute recovery-time objective when incorporated into a tested operational runbook.

Setting up the application in two Regions with Route 53 failover routing to Application Load Balancers provides the corresponding application-tier failover mechanism. Route 53 health checks detect an unhealthy primary endpoint and DNS failover directs clients to the healthy Regional endpoint. Auto Scaling groups in each Region ensure that web capacity can be maintained or expanded according to demand after traffic is redirected. Together, the globally replicated and promotable database tier plus independently deployed, health-checked application stacks address both data recovery and service availability. See the Amazon Aurora Global Database documentation and Route 53 failover routing documentation.

QUESTION NO: 46

A company that uses electronic patient health records runs a fleet of Amazon EC2 instances with an Amazon Linux operating system. The company must continuously ensure that the EC2 instances are running operating system patches and application patches that are in compliance with current privacy regulations. The company uses a custom repository to store application patches.

A DevOps engineer needs to automate the deployment of operating system patches and application patches. The DevOps engineer wants to use both the default operating system patch repository and the custom patch repository.

Which solution will meet these requirements with the LEAST effort?

A. Use AWS Systems Manager to create a new custom patch baseline that includes the default operating system repository and the custom repository. Run the AWS-RunPatchBaseline document by using the Run command to verify and install patches. Use the BaselineOverride API to configure the new custom patch baseline.

B. Use AWS Direct Connect to integrate the custom repository with the EC2 instances. Use Amazon EventBridge events to deploy the patches.

C. Use the yum-config-manager command to add the custom repository to the /etc/yum.repos.d configuration. Run the yum-config-manager-enable command to activate the new repository.

D. Use AWS Systems Manager to create a patch baseline for the default operating system repository and a second patch baseline for the custom repository. Run the AWS-RunPatchBaseline document by using the Run command to verify and install patches. Use the BaselineOverride API to configure the default patch baseline and the custom patch baseline.

E. Configure and enable the custom repository in the Amazon Linux yum configuration. Create a scheduled AWS Systems Manager State Manager association that runs AWS-RunPatchBaseline by using a patch baseline that approves the required operating system and application patches.

ANSWER: E

Explanation:

Configure the custom repository in the Amazon Linux yum configuration, and use a scheduled Systems Manager State Manager association to run AWS-RunPatchBaseline. This combines the required repository access with managed, recurring patch automation. Amazon Linux package operations use yum (or its compatible tooling), which reads enabled repository definitions from the system configuration. Therefore, after the custom application repository is configured and enabled, the native package manager can resolve applicable packages from both the standard Amazon Linux repositories and the organization’s custom repository.

A Systems Manager State Manager association provides the continuous automation that repository configuration alone does not provide. The association can run the AWS-RunPatchBaseline document on a defined schedule across the managed EC2 fleet. Patch Manager uses the instance’s native Linux package-management tooling to scan for and install patches according to the selected patch baseline. This also provides centralized execution history and compliance reporting, which are useful for demonstrating an ongoing patch-management process for regulated workloads. A patch baseline can be tailored to approve the required operating system and application packages while the enabled repository configuration supplies the

package sources. AWS documents Linux Patch Manager behavior at How Patch Manager works on Linux and scheduled associations at AWS Systems Manager State Manager.

QUESTION NO: 47

A company deploys a web application on Amazon EC2 instances that are behind an Application Load Balancer (ALB). The company stores the application code in an AWS CodeCommit repository. When code is merged to the main branch, an AWS Lambda function invokes an AWS CodeBuild project. The CodeBuild project packages the code, stores the packaged code in AWS CodeArtifact, and invokes AWS Systems Manager Run Command to deploy the packaged code to the EC2 instances.

Previous deployments have resulted in defects, EC2 instances that are not running the latest version of the packaged code, and inconsistencies between instances.

Which combination of actions should a DevOps engineer take to implement a more reliable deployment solution? (Select TWO.)

A. Create a pipeline in AWS CodePipeline that uses the CodeCommit repository as a source provider. Configure pipeline stages that run the CodeBuild project in parallel to build and test the application. In the pipeline, pass the CodeBuild project output artifact to an AWS CodeDeploy action.

B. Create a pipeline in AWS CodePipeline that uses the CodeCommit repository as a source provider. Create separate pipeline stages that run a CodeBuild project to build and then test the application. In the pipeline, pass the CodeBuild project output artifact to an AWS CodeDeploy action.

C. Create an AWS CodeDeploy application and a deployment group to deploy the packaged code to the EC2 instances. Configure the ALB for the deployment group.

D. Create individual Lambda functions that use AWS CodeDeploy instead of Systems Manager to run build, test, and deploy actions.

E. Create an Amazon S3 bucket. Modify the CodeBuild project to store the packages in the S3 bucket instead of in CodeArtifact. Use deploy actions in CodeDeploy to deploy the artifact to the EC2 instances.

ANSWER: A C

Explanation:

Creating a CodePipeline pipeline with CodeCommit as the source, parallel CodeBuild build and test actions, and a CodeDeploy deployment action establishes a managed, repeatable continuous-delivery workflow. CodePipeline coordinates each revision through defined stages and passes versioned output artifacts between actions. Running independent build and test work in parallel can shorten feedback time while ensuring the deployable artifact proceeds through an automated release process. CodeDeploy then provides a purpose-built deployment mechanism rather than relying on ad hoc remote commands to update each instance.

Creating a CodeDeploy application and deployment group for the EC2 fleet, with the Application Load Balancer configured for that deployment group, provides controlled fleet-wide deployments. CodeDeploy can coordinate deployment lifecycle events across all targeted instances, use deployment configurations to control rollout behavior, and integrate load balancer health checks. During deployments, instances can be taken out of service while they are updated and returned to service after successful validation. This helps ensure a consistent application revision across the fleet and supports automated rollback when a deployment fails. See the AWS CodePipeline User Guide and the AWS CodeDeploy EC2/on-premises deployment documentation.

QUESTION NO: 48

A company hosts its staging website using an Amazon EC2 instance backed with Amazon EBS storage. The company wants to recover quickly with minimal data losses in the event of network connectivity issues or power failures on the EC2 instance.

Which solution will meet these requirements?

A. Add the instance to an EC2 Auto Scaling group with the minimum, maximum, and desired capacity set to 1.

B. Add the instance to an EC2 Auto Scaling group with a lifecycle hook to detach the EBS volume when the EC2 instance shuts down or terminates.

C. Create an Amazon CloudWatch alarm for the StatusCheckFailed System metric and select the EC2 action to recover the instance.

D. Create an Amazon CloudWatch alarm for the StatusCheckFailed Instance metric and select the EC2 action to reboot the instance.

ANSWER: C

Explanation:

Create an Amazon CloudWatch alarm for the StatusCheckFailed System metric and select the EC2 action to recover the instance. A system status check detects problems with the underlying AWS infrastructure that hosts the instance, including failures involving network connectivity, power, or physical host hardware. When this alarm enters the ALARM state, the EC2 recover action automatically moves the instance to healthy hardware and restarts it.

Instance recovery is designed to minimize disruption for an EBS-backed instance. The recovered instance retains its instance ID, Elastic IP addresses, private IP addresses, network interface configuration, IAM role, and attached EBS volumes. Because the EBS volumes are preserved and reattached, persistent application data remains available after recovery. This directly supports rapid restoration while minimizing data loss from infrastructure-level failures. The EC2 recovery action is specifically supported for system status-check failures and is configured through a CloudWatch alarm. See the Amazon EC2 instance recovery documentation and the CloudWatch alarm actions documentation.

QUESTION NO: 49

A DevOps engineer is implementing governance controls for a company that requires its infrastructure to be housed within the United States. The company has many AWS accounts in an organization in AWS Organizations that has all features enabled. The engineer must restrict which AWS Regions the company can use. The engineer must also ensure that an alert is sent as soon as possible if any activity outside the governance policy occurs. The controls must be automatically enabled on any new Region outside the United States. Which combination of steps will meet these requirements? (Select TWO.)

A. Create an Organizations SCP deny policy that has a condition that the aws:RequestedRegion property does not match a list of all US Regions. Include an exception in the policy for global services. Attach the policy to the root of the organization.

B. Configure AWS CloudTrail to send logs to Amazon CloudWatch Logs. Enable CloudTrail for all Regions. Use a CloudWatch Logs metric filter to create a metric in non-US Regions. Configure a CloudWatch alarm to send an alert if the metric is greater than 0.

C. Use an AWS Lambda function that checks for AWS service activity. Deploy the Lambda function to all Regions. Write an Amazon EventBridge rule that runs the Lambda function every hour. Configure the rule to send an alert if the Lambda function finds any activity in a non-US Region.

D. Use an AWS Lambda function to query Amazon Inspector to look for service activity in non-US Regions. Configure the Lambda function to send alerts if Amazon Inspector finds any activity.

E. Create an Organizations SCP allow policy that has a condition that the aws:RequestedRegion property matches a list of all US Regions. Include an exception in the policy for global services. Attach the policy to the root of the organization.

ANSWER: A B

Explanation:

Create an Organizations SCP deny policy that has a condition that the aws:RequestedRegion property does not match a list of all US Regions. Include an exception in the policy for global services. Attach the policy to the root of the organization. This establishes a preventive guardrail for every member account under the organization root. An explicit deny based on `aws:RequestedRegion` blocks requests directed to Regions outside the approved US Region list, while the global-service exception avoids unintentionally blocking services whose API operations are handled through a designated global endpoint.

Because the policy denies every Region not on the approved list, it also applies automatically when AWS launches a new Region outside the United States.

Configure AWS CloudTrail to send logs to Amazon CloudWatch Logs. Enable CloudTrail for all Regions. Use a CloudWatch Logs metric filter to create a metric in non-US Regions. Configure a CloudWatch alarm to send an alert if the metric is greater than 0. A multi-Region CloudTrail trail records account activity from all current and future AWS Regions, including events from newly launched Regions. Delivering those events to CloudWatch Logs enables metric filters and alarms to detect relevant activity and notify responders promptly. Together, the SCP provides prevention and CloudTrail/CloudWatch provides centralized detection and alerting. See the AWS documentation for SCP Region-denial examples and multi-Region CloudTrail trails.

QUESTION NO: 50

A company uses a trunk-based development branching strategy. The company has two AWS CodePipeline pipelines that are integrated with a Git provider. The pull_request pipeline has a branch filter that matches the feature branches. The main_branch pipeline has a branch filter that matches the main branch.

When pull requests are merged into the main branch, the pull requests are deployed by using the main_branch pipeline. The company ' s developers need test results for all submitted pull requests as quickly as possible from the pull_request pipeline. The company wants to ensure that the main_branch pipeline ' s test results finish and that each deployment is complete before the next pipeline execution.

Which solution will meet these requirements?

A. Configure the pull_request pipeline to use SUPERSEDED mode. Configure the main_branch pipeline to use QUEUED mode.

B. Configure the pull_request pipeline to use PARALLEL mode. Configure the main_branch pipeline to use QUEUED mode.

C. Configure the pull_request pipeline to use PARALLEL mode. Configure the main_branch pipeline to use SUPERSEDED mode.

D. Configure the pull_request pipeline to use QUEUED mode. Configure the main_branch pipeline to use SUPERSEDED mode.

ANSWER: B

Explanation:

Configure the pull_request pipeline to use PARALLEL mode. Configure the main_branch pipeline to use QUEUED mode. This configuration directly separates the feedback workload for feature-branch pull requests from the controlled deployment workload for the main branch. In PARALLEL execution mode, CodePipeline can run multiple pipeline executions concurrently. Therefore, separate submitted pull requests can each begin their validation work without waiting for an earlier pull request pipeline execution to finish, providing developers with test feedback as quickly as available build and test capacity permits.

For changes merged into the main branch, QUEUED execution mode preserves execution order by placing a new execution in a queue when another execution is already in progress. The next main-branch execution starts only after the preceding execution completes. This ensures that test stages complete and that each deployment reaches completion before a later merged change proceeds through the pipeline. This is particularly important when successive deployments target the same environments or depend on the state produced by prior deployments. AWS documents PARALLEL mode for concurrent executions and QUEUED mode for sequentially processed executions in its CodePipeline execution mode documentation. Additional implementation details are available in the AWS CodePipeline pipeline requirements guide.

QUESTION NO: 51

A company uses AWS Organizations with all features enabled. The security team must centrally collect AWS CloudTrail management events from every account and AWS Region. The security team must also ensure that CloudTrail log files cannot be deleted or modified during a 1-year retention period.

Which combination of actions should a DevOps engineer take to meet these requirements? (Choose three.)

A. Create a multi-Region organization trail that delivers management events to an Amazon S3 bucket in the security account.

B. Add a bucket policy that allows the CloudTrail service principal to write log files to the destination bucket.

C. Enable S3 Object Lock on the destination bucket and configure a 1-year retention period for delivered log objects.

D. Create separate trails in each member account and configure each trail to write logs to an S3 bucket in the same account.

E. Enable S3 Cross-Region Replication on the destination bucket to prevent CloudTrail log files from being modified.

F. Create an AWS Config managed rule to prevent users from calling CloudTrail StopLogging.

ANSWER: A B C

Explanation:

An organization trail records events for all accounts in an AWS Organization. Configuring the trail as multi-Region ensures that it records management events in all current and future AWS Regions. The organization trail delivers log files to a centrally managed Amazon S3 bucket.

The destination bucket policy must allow the CloudTrail service principal to write log files and to read the bucket ACL. To enforce the retention requirement, enable S3 Object Lock on the destination bucket and use a retention configuration that prevents objects from being deleted or overwritten for 1 year. Object Lock requires S3 Versioning and provides write-once-read-many protection for retained object versions.

See Creating a trail for an organization and Using S3 Object Lock.