ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

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

www.atmicnetworks .com

Warning: Keep connected with our support team for latest updates

Question: 1

A company is implementing an application on Amazon EC2 instances. The application needs to process incoming transactions. When the application detects a transaction that is not valid, the application must send a chat message to the company's support team. To send the message, the application needs to retrieve the access token to authenticate by using the chat API.

A developer needs to implement a solution to store the access token. The access token must be encrypted at rest and in transit. The access token must also be accessible from other AWS accounts.

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

A. Use an AWS Systems Manager Parameter Store SecureString parameter that uses an AWS Key Management Service (AWS KMS) AWS managed key to store the access token. Add a resource-based policy to the parameter to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Parameter Store. Retrieve the token from Parameter Store with the decrypt flag enabled. Use the decrypted access token to send the message to the chat.

B. Encrypt the access token by using an AWS Key Management Service (AWS KMS) customer managed key. Store the access token in an Amazon DynamoDB table. Update the IAM role of the EC2 instances with permissions to access DynamoDB and AWS KMS. Retrieve the token from DynamoDB. Decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the message to the chat.

C. Use AWS Secrets Manager with an AWS Key Management Service (AWS KMS) customer managed key to store the access token. Add a resource-based policy to the secret to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Secrets Manager. Retrieve the token from Secrets Manager. Use the decrypted access token to send the message to the chat.

D. Encrypt the access token by using an AWS Key Management Service (AWS KMS) AWS managed key. Store the access token in an Amazon S3 bucket. Add a bucket policy to the S3 bucket to allow access from other accounts. Update the IAM role of the EC2 instances with permissions to access Amazon S3 and AWS KMS. Retrieve the token from the S3 bucket. Decrypt the token by using AWS KMS on the EC2 instances. Use the decrypted access token to send the massage to the chat.

Answer: C

Explanation:

https://aws.amazon.com/premiumsupport/knowledge-center/secrets-manager-share-between- accounts/

https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access examples cross.html

Question: 2

A company is running Amazon EC2 instances in multiple AWS accounts. A developer needs to implement an application that collects all the lifecycle events of the EC2 instances. The application needs to store the lifecycle events in a single Amazon Simple Queue Service (Amazon SQS) queue in the company's main AWS account for further processing.

Which solution will meet these requirements?

A. Configure Amazon EC2 to deliver the EC2 instance lifecycle events from all accounts to the Amazon EventBridge event bus of the main account. Add an EventBridge rule to the event bus of the main account that matches all EC2 instance lifecycle events. Add the SQS queue as a target of the rule.

B. Use the resource policies of the SQS queue in the main account to give each account permissions to write to that SQS queue. Add to the Amazon EventBridge event bus of each account an EventBridge rule that matches all EC2 instance lifecycle events. Add the SQS queue in the main account as a target of the rule.

C. Write an AWS Lambda function that scans through all EC2 instances in the company accounts to detect EC2 instance lifecycle changes. Configure the Lambda function to write a notification message to the SQS queue in the main account if the function detects an EC2 instance lifecycle change. Add an Amazon EventBridge scheduled rule that invokes the Lambda function every minute.

D. Configure the permissions on the main account event bus to receive events from all accounts. Create an Amazon EventBridge rule in each account to send all the EC2 instance lifecycle events to the main account event bus. Add an EventBridge rule to the main account event bus that matches all EC2 instance lifecycle events. Set the SQS queue as a target for the rule.

_____ Answer: D

Explanation:

Amazon EC2 instances can send the state-change notification events to Amazon EventBridge.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-instance-state-changes.html Amazon EventBridge can send and receive events between event buses in AWS accounts. https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-cross-account.html

Question: 3

An application is using Amazon Cognito user pools and identity pools for secure access. A developer wants to integrate the user-specific file upload and download features in the application with Amazon S3. The developer must ensure that the files are saved and retrieved in a secure manner and that users can access only their own files. The file sizes range from 3 KB to 300 MB.

Which option will meet these requirements with the HIGHEST level of security?

A. Use S3 Event Notifications to validate the file upload and download requests and update the user interface (UI).

B. Save the details of the uploaded files in a separate Amazon DynamoDB table. Filter the list of files in the user interface (UI) by comparing the current user ID with the user ID associated with the file in the table.

C. Use Amazon API Gateway and an AWS Lambda function to upload and download files. Validate each request in the Lambda function before performing the requested operation.

D. Use an IAM policy within the Amazon Cognito identity prefix to restrict users to use their own folders in Amazon S3.

_____ Answer: D

Explanation:

https://docs.aws.amazon.com/cognito/latest/developerguide/amazon-cognito-integrating-user- pools-with-identity-pools.html

Question: 4

A company is building a scalable data management solution by using AWS services to improve the speed and agility of development. The solution will ingest large volumes of data from various sources and will process this data through multiple business rules and transformations.

The solution requires business rules to run in sequence and to handle reprocessing of data if errors occur when the business rules run. The company needs the solution to be scalable and to require the least possible maintenance.

Which AWS service should the company use to manage and automate the orchestration of the data flows to meet these requirements?

A. AWS Batch

B. AWS Step Functions

C. AWS Glue

D. AWS Lambda

Answer: B

Explanation:

https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html

Question: 5

A developer has created an AWS Lambda function that is written in Python. The Lambda function reads data from objects in Amazon S3 and writes data to an Amazon DynamoDB table. The function is successfully invoked from an S3 event notification when an object is created. However, the function fails when it attempts to write to the DynamoDB table.

What is the MOST likely cause of this issue?

A. The Lambda function's concurrency limit has been exceeded.

B. DynamoDB table requires a global secondary index (GSI) to support writes.

C. The Lambda function does not have IAM permissions to write to DynamoDB.

D. The DynamoDB table is not running in the same Availability Zone as the Lambda function.

Answer: C

Explanation:

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference policies examples lambda-access- dynamodb.html

Question: 6

A developer is creating an AWS CloudFormation template to deploy Amazon EC2 instances across multiple AWS accounts. The developer must choose the EC2 instances from a list of approved instance types.

How can the developer incorporate the list of approved instance types in the CloudFormation template?

A. Create a separate CloudFormation template for each EC2 instance type in the list.

B. In the Resources section of the CloudFormation template, create resources for each EC2 instance type in the list.

C. In the CloudFormation template, create a separate parameter for each EC2 instance type in the list.

D. In the CloudFormation template, create a parameter with the list of EC2 instance types as AllowedValues.

Answer: D

Explanation:

In the CloudFormation template, the developer should create a parameter with the list of approved EC2 instance types as AllowedValues. This way, users can select the instance type they want to use when launching the CloudFormation stack, but only from the approved list.

Question: 7

A developer has an application that makes batch requests directly to Amazon DynamoDB by using the BatchGetItem low-level API operation. The responses frequently return values in the UnprocessedKeys element.

Which actions should the developer take to increase the resiliency of the application when the batch response includes values in UnprocessedKeys? (Choose two.)

A. Retry the batch operation immediately.

B. Retry the batch operation with exponential backoff and randomized delay.

C. Update the application to use an AWS software development kit (AWS SDK) to make the requests.

D. Increase the provisioned read capacity of the DynamoDB tables that the operation accesses.

E. Increase the provisioned write capacity of the DynamoDB tables that the operation accesses.

Answer: B, C

Explanation:

The UnprocessedKeys element indicates that the BatchGetItem operation did not process all of the requested items in the current

response. This can happen if the response size limit is exceeded or if the table's provisioned throughput is exceeded. To handle this situation, the developer should retry the batch operation with exponential backoff and randomized delay to avoid throttling errors and reduce the load on the table. The developer should also use an AWS SDK to make the requests, as the SDKs automatically retry requests that return UnprocessedKeys.

Reference:

[BatchGetItem - Amazon DynamoDB]

[Working with Queries and Scans - Amazon DynamoDB]

[Best Practices for Handling DynamoDB Throttling Errors]

Question: 8

A company is running a custom application on a set of on-premises Linux servers that are accessed using Amazon API Gateway. AWS X-Ray tracing has been enabled on the API test stage.

How can a developer enable X-Ray tracing on the on-premises servers with the LEAST amount of configuration?

A. Install and run the X-Ray SDK on the on-premises servers to capture and relay the data to the X- Ray service.

B. Install and run the X-Ray daemon on the on-premises servers to capture and relay the data to the X-Ray service.

C. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the PutTraceSegments API call.

D. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the PutTelemetryRecords API call.

Answer: B

Explanation:

The X-Ray daemon is a software that collects trace data from the X-Ray SDK and relays it to the X-Ray service. The X-Ray daemon can run on any platform that supports Go, including Linux, Windows, and macOS. The developer can install and run the X-Ray daemon on the on-premises servers to capture and relay the data to the X-Ray service with minimal configuration. The X-Ray SDK is used to instrument the application code, not to capture and relay data. The Lambda function solutions are more complex and require additional configuration.

Reference:

[AWS X-Ray concepts - AWS X-Ray]

[Setting up AWS X-Ray - AWS X-Ray]

Question: 9

A company wants to share information with a third party. The third party has an HTTP API endpoint that the company can use to share the information. The company has the required API key to access the HTTP API.

The company needs a way to manage the API key by using code. The integration of the API key with the application code cannot affect application performance.

Which solution will meet these requirements MOST securely?

A. Store the API credentials in AWS Secrets Manager. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.

B. Store the API credentials in a local code variable. Push the code to a secure Git repository. Use the local code variable at runtime to make the API call.

C. Store the API credentials as an object in a private Amazon S3 bucket. Restrict access to the S3 object by using IAM policies. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.

D. Store the API credentials in an Amazon DynamoDB table. Restrict access to the table by using resource-based policies. Retrieve the API credentials at runtime by using the AWS SDK. Use the credentials to make the API call.

Answer: A

Explanation:

AWS Secrets Manager is a service that helps securely store, rotate, and manage secrets such as API keys, passwords, and tokens. The developer can store the API credentials in AWS Secrets Manager and retrieve them at runtime by using the AWS SDK. This solution will meet the requirements of security, code management, and performance. Storing the API credentials in a local code variable or an S3 object is not secure, as it exposes the credentials to unauthorized access or leakage. Storing the API credentials in a DynamoDB table is also not secure, as it requires additional encryption and access control measures. Moreover, retrieving the credentials from S3 or DynamoDB may affect application performance due to network latency.

Reference:

[What Is AWS Secrets Manager? - AWS Secrets Manager]

[Retrieving a Secret - AWS Secrets Manager]

Question: 10

A developer is deploying a new application to Amazon Elastic Container Service (Amazon ECS). The developer needs to securely store and retrieve different types of variables. These variables include authentication information for a remote API, the URL for the API, and credentials. The authentication information and API URL must be available to all current and future deployed versions of the application across development, testing, and production environments.

How should the developer retrieve the variables with the FEWEST application changes?

A. Update the application to retrieve the variables from AWS Systems Manager Parameter Store. Use unique paths in Parameter Store for each variable in each environment. Store the credentials in AWS Secrets Manager in each environment.

B. Update the application to retrieve the variables from AWS Key Management Service (AWS KMS). Store the API URL and credentials as unique keys for each environment.

C. Update the application to retrieve the variables from an encrypted file that is stored with the application. Store the API URL and credentials in unique files for each environment.

D. Update the application to retrieve the variables from each of the deployed environments. Define the authentication information and API URL in the ECS task definition as unique names during the deployment process.

Answer: A

Explanation:

AWS Systems Manager Parameter Store is a service that provides secure, hierarchical storage for configuration data management and secrets management. The developer can update the application to retrieve the variables from Parameter Store by using the AWS SDK or the AWS CLI. The developer can use unique paths in Parameter Store for each variable in each environment, such as /dev/api-url, /test/api-url, and /prod/api-url. The developer can also store the credentials in AWS Secrets Manager, which is integrated with Parameter Store and provides additional features such as automatic rotation and encryption.

Reference:

[What Is AWS Systems Manager? - AWS Systems Manager]

[Parameter Store - AWS Systems Manager]

[What Is AWS Secrets Manager? - AWS Secrets Manager]

Question: 11

A company is migrating legacy internal applications to AWS. Leadership wants to rewrite the internal employee directory to use native AWS services. A developer needs to create a solution for storing employee contact details and high-resolution photos for use with the new application.

Which solution will enable the search and retrieval of each employee's individual details and high-resolution photos using AWS APIs?

A. Encode each employee's contact information and photos using Base64. Store the information in an Amazon DynamoDB table using a sort key.

B. Store each employee's contact information in an Amazon DynamoDB table along with the object keys for the photos stored in Amazon S3.

C. Use Amazon Cognito user pools to implement the employee directory in a fully managed software-as-a-service (SaaS)

method.

D. Store employee contact information in an Amazon RDS DB instance with the photos stored in Amazon Elastic File System (Amazon EFS).

Answer: B

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. The developer can store each employee's contact information in a DynamoDB table along with the object keys for the photos stored in Amazon S3. Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. The developer can use AWS APIs to search and retrieve the employee details and photos from DynamoDB and S3.

Reference:

[Amazon DynamoDB]

[Amazon Simple Storage Service (S3)]

Question: 12

A developer is creating an application that will give users the ability to store photos from their cellphones in the cloud. The application needs to support tens of thousands of users. The application uses an Amazon API Gateway REST API that is integrated with AWS Lambda functions to process the photos. The application stores details about the photos in Amazon DynamoDB.

Users need to create an account to access the application. In the application, users must be able to upload photos and retrieve previously uploaded photos. The photos will range in size from 300 KB to 5 MB.

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

A. Use Amazon Cognito user pools to manage user accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. Use the Lambda function to store the photos and details in the DynamoDB table. Retrieve previously uploaded photos directly from the DynamoDB table.

B. Use Amazon Cognito user pools to manage user accounts. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

C. Create an IAM user for each user of the application during the sign-up process. Use IAM authentication to access the API Gateway API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

D. Create a users table in DynamoDB. Use the table to manage user accounts. Create a Lambda authorizer that validates user credentials against the users table. Integrate the Lambda authorizer with API Gateway to control access to the API. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as par of the photo details in the DynamoDB table. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

Answer: B

Explanation:

Amazon Cognito user pools is a service that provides a secure user directory that scales to hundreds of millions of users. The developer can use Amazon Cognito user pools to manage user accounts and create an Amazon Cognito user pool authorizer in API Gateway to control access to the API. The developer can use the Lambda function to store the photos in Amazon S3, which is a highly scalable, durable, and secure object storage service. The developer can store the object's S3 key as part of the photo details in the DynamoDB table, which is a fast and flexible NoSQL database service. The developer can retrieve previously uploaded photos by querying DynamoDB for the S3 key and fetching the photos from S3. This solution will meet the requirements with the least operational overhead.

Reference:

[Amazon Cognito User Pools]

[Use Amazon Cognito User Pools - Amazon API Gateway]

[Amazon Simple Storage Service (S3)]

[Amazon DynamoDB]

Question: 13

A company receives food orders from multiple partners. The company has a microservices application that uses Amazon API Gateway APIs with AWS Lambda integration. Each partner sends orders by calling a customized API that is exposed through API Gateway. The API call invokes a shared Lambda function to process the orders.

Partners need to be notified after the Lambda function processes the orders. Each partner must receive updates for only the partner's own orders. The company wants to add new partners in the future with the fewest code changes possible.

Which solution will meet these requirements in the MOST scalable way?

A. Create a different Amazon Simple Notification Service (Amazon SNS) topic for each partner. Configure the Lambda function to publish messages for each partner to the partner's SNS topic.

B. Create a different Lambda function for each partner. Configure the Lambda function to notify each partner's service endpoint directly.

C. Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure the Lambda function to publish messages with specific attributes to the SNS topic. Subscribe each partner to the SNS topic. Apply the appropriate filter policy to the topic subscriptions.

D. Create one Amazon Simple Notification Service (Amazon SNS) topic. Subscribe all partners to the SNS topic.

Answer: C

Explanation:

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service that enables pub/sub communication between

distributed systems. The developer can create an SNS topic and configure the Lambda function to publish messages with specific attributes to the topic. The developer can subscribe each partner to the SNS topic and apply the appropriate filter policy to the topic subscriptions. This way, each partner will receive updates for only their own orders based on the message attributes. This solution will meet the requirements in the most scalable way and allow adding new partners in the future with minimal code changes.

Reference:

[Amazon Simple Notification Service (SNS)]

[Filtering Messages with Attributes - Amazon Simple Notification Service]

Question: 14

A financial company must store original customer records for 10 years for legal reasons. A complete record contains personally identifiable information (PII). According to local regulations, PII is available to only certain people in the company and must not be shared with third parties. The company needs to make the records available to third-party organizations for statistical analysis without sharing the PII.

A developer wants to store the original immutable record in Amazon S3. Depending on who accesses the S3 document, the document should be returned as is or with all the PII removed. The developer has written an AWS Lambda function to remove the PII from the document. The function is named removePii.

What should the developer do so that the company can meet the PII requirements while maintaining only one copy of the document?

A. Set up an S3 event notification that invokes the removePii function when an S3 GET request is made. Call Amazon S3 by using a GET request to access the object without PII.

B. Set up an S3 event notification that invokes the removePii function when an S3 PUT request is made. Call Amazon S3 by using a PUT request to access the object without PII.

C. Create an S3 Object Lambda access point from the S3 console. Select the removePii function. Use S3 Access Points to access the object without PII.

D. Create an S3 access point from the S3 console. Use the access point name to call the GetObjectLegalHold S3 API function. Pass in the removePii function name to access the object without PII.

Answer: C

Explanation:

S3 Object Lambda allows you to add your own code to process data retrieved from S3 before returning it to an application. You can use an AWS Lambda function to modify the data, such as removing PII, redacting confidential information, or resizing images. You can create an S3 Object Lambda access point and associate it with your Lambda function. Then, you can use the access point to request objects from S3 and get the modified data back. This way, you can maintain only one copy of the original document in S3 and apply different transformations depending on who accesses it. Reference: Using AWS Lambda with Amazon S3

Question: 15

A developer is deploying an AWS Lambda function The developer wants the ability to return to older versions of the function quickly and seamlessly.

How can the developer achieve this goal with the LEAST operational overhead?

A. Use AWS OpsWorks to perform blue/green deployments.

B. Use a function alias with different versions.

C. Maintain deployment packages for older versions in Amazon S3.

D. Use AWS CodePipeline for deployments and rollbacks.

Answer: B

Explanation:

A function alias is a pointer to a specific Lambda function version. You can use aliases to create different environments for your function, such as development, testing, and production. You can also use aliases to perform blue/green deployments by shifting traffic between two versions of your function gradually. This way, you can easily roll back to a previous version if something goes wrong, without having to redeploy your code or change your configuration. Reference: AWS Lambda function aliases

Question: 16

A developer has written an AWS Lambda function. The function is CPU-bound. The developer wants to ensure that the function returns responses quickly.

How can the developer improve the function's performance?

A. Increase the function's CPU core count.

B. Increase the function's memory.

C. Increase the function's reserved concurrency.

D. Increase the function's timeout.

Answer: B

Explanation:

The amount of memory you allocate to your Lambda function also determines how much CPU and network bandwidth it gets. Increasing the memory size can improve the performance of CPU-bound functions by giving them more CPU power. The CPU allocation is proportional to the memory allocation, so a function with 1 GB of memory has twice the CPU power of a function with 512 MB of memory. Reference: AWS Lambda execution environment

Question: 17

For a deployment using AWS Code Deploy, what is the run order of the hooks for in-place deployments?

A. BeforeInstall -> ApplicationStop -> ApplicationStart -> AfterInstall

B. ApplicationStop -> BeforeInstall -> AfterInstall -> ApplicationStart

C. BeforeInstall -> ApplicationStop -> ValidateService -> ApplicationStart

D. ApplicationStop -> BeforeInstall -> ValidateService -> ApplicationStart

Answer: B

Explanation:

For in-place deployments, AWS CodeDeploy uses a set of predefined hooks that run in a specific order during each deployment lifecycle event. The hooks are ApplicationStop, BeforeInstall, AfterInstall, ApplicationStart, and ValidateService. The run order of the hooks for in-place deployments is as follows:

ApplicationStop: This hook runs first on all instances and stops the current application that is running on the instances.

BeforeInstall: This hook runs after ApplicationStop on all instances and performs any tasks required before installing the new application revision.

AfterInstall: This hook runs after BeforeInstall on all instances and performs any tasks required after installing the new application revision.

ApplicationStart: This hook runs after AfterInstall on all instances and starts the new application that has been installed on the instances.

ValidateService: This hook runs last on all instances and verifies that the new application is running properly on the instances.

Reference: [AWS CodeDeploy lifecycle event hooks reference]

Question: 18

A company is building a serverless application on AWS. The application uses an AWS Lambda function to process customer orders 24 hours a day, 7 days a week. The Lambda function calls an external vendor's HTTP API to process payments.

During load tests, a developer discovers that the external vendor payment processing API occasionally times out and returns errors. The company expects that some payment processing API calls will return errors.

The company wants the support team to receive notifications in near real time only when the payment processing external API error rate exceed 5% of the total number of transactions in an hour. Developers need to use an existing Amazon Simple Notification Service (Amazon SNS) topic that is configured to notify the support team.

Which solution will meet these requirements?

A. Write the results of payment processing API calls to Amazon CloudWatch. Use Amazon CloudWatch Logs Insights to query the

CloudWatch logs. Schedule the Lambda function to check the CloudWatch logs and notify the existing SNS topic.

B. Publish custom metrics to CloudWatch that record the failures of the external payment processing API calls. Configure a CloudWatch alarm to notify the existing SNS topic when error rate exceeds the specified rate.

C. Publish the results of the external payment processing API calls to a new Amazon SNS topic. Subscribe the support team members to the new SNS topic.

D. Write the results of the external payment processing API calls to Amazon S3. Schedule an Amazon Athena query to run at regular intervals. Configure Athena to send notifications to the existing SNS topic when the error rate exceeds the specified rate.

Answer: B

Explanation:

Amazon CloudWatch is a service that monitors AWS resources and applications. The developer can publish custom metrics to CloudWatch that record the failures of the external payment processing API calls. The developer can configure a CloudWatch alarm to notify the existing SNS topic when the error rate exceeds 5% of the total number of transactions in an hour. This solution will meet the requirements in a near real-time and scalable way.

Reference:

[What Is Amazon CloudWatch? - Amazon CloudWatch]

[Publishing Custom Metrics - Amazon CloudWatch]

[Creating Amazon CloudWatch Alarms - Amazon CloudWatch]

Question: 19

A company is offering APIs as a service over the internet to provide unauthenticated read access to statistical information that is updated daily. The company uses Amazon API Gateway and AWS Lambda to develop the APIs. The service has become popular, and the company wants to enhance the responsiveness of the APIs.

Which action can help the company achieve this goal?

A. Enable API caching in API Gateway.

B. Configure API Gateway to use an interface VPC endpoint.

C. Enable cross-origin resource sharing (CORS) for the APIs.

D. Configure usage plans and API keys in API Gateway.

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The

developer can enable API caching in API Gateway to cache responses from the backend integration point for a specified time-to-live (TTL) period. This can improve the responsiveness of the APIs by reducing the number of calls made to the backend service.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Enable API Caching to Enhance Responsiveness - Amazon API Gateway]

Question: 20

A developer wants to store information about movies. Each movie has a title, release year, and genre. The movie information also can include additional properties about the cast and production crew. This additional information is inconsistent across movies. For example, one movie might have an assistant director, and another movie might have an animal trainer.

The developer needs to implement a solution to support the following use cases:

For a given title and release year, get all details about the movie that has that title and release year.

For a given title, get all details about all movies that have that title.

For a given genre, get all details about all movies in that genre.

Which data store configuration will meet these requirements?

A. Create an Amazon DynamoDB table. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort key. Create a global secondary index that uses the genre as the partition key and the title as the sort key.

B. Create an Amazon DynamoDB table. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort key. Create a global secondary index that uses the title as the partition key.

C. On an Amazon RDS DB instance, create a table that contains columns for title, release year, and genre. Configure the title as the primary key.

D. On an Amazon RDS DB instance, create a table where the primary key is the title and all other data is encoded into JSON format as one additional column.

Answer: A

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. The developer can create a DynamoDB table and configure the table with a primary key that consists of the title as the partition key and the release year as the sort key. This will enable querying for a given title and release year efficiently. The developer can also create a global secondary index that uses the genre as the partition key and the title as the sort key.

This will enable querying for a given genre efficiently. The developer can store additional properties about the cast and production crew

as attributes in the DynamoDB table. These attributes can have different data types and structures, and they do not need to be consistent across items.

Reference:

[Amazon DynamoDB]

[Working with Queries - Amazon DynamoDB]

[Working with Global Secondary Indexes - Amazon DynamoDB]

Question: 21

A developer maintains an Amazon API Gateway REST API. Customers use the API through a frontend UI and Amazon Cognito authentication.

The developer has a new version of the API that contains new endpoints and backward-incompatible interface changes. The developer needs to provide beta access to other developers on the team without affecting customers.

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

A. Define a development stage on the API Gateway API. Instruct the other developers to point the endpoints to the development stage.

B. Define a new API Gateway API that points to the new API application code. Instruct the other developers to point the endpoints to the new API.

C. Implement a query parameter in the API application code that determines which code version to call.

D. Specify new API Gateway endpoints for the API endpoints that the developer wants to add.

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The developer can define a development stage on the API Gateway API and instruct the other developers to point the endpoints to the development stage. This way, the developer can provide beta access to the new version of the API without affecting customers who use the production stage. This solution will meet the requirements with the least operational overhead.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Set up a Stage in API Gateway - Amazon API Gateway]

Question: 22

A developer is creating an application that will store personal health information (PHI). The PHI needs to be encrypted at all times. An encrypted Amazon RDS for MySQL DB instance is storing the data. The developer wants to increase the performance of the application by caching frequently accessed data while adding the ability to sort or rank the cached datasets.

Which solution will meet these requirements?

A. Create an Amazon ElastiCache for Redis instance. Enable encryption of data in transit and at rest. Store frequently accessed data in the cache.

B. Create an Amazon ElastiCache for Memcached instance. Enable encryption of data in transit and at rest. Store frequently accessed data in the cache.

C. Create an Amazon RDS for MySQL read replica. Connect to the read replica by using SSL. Configure the read replica to store frequently accessed data.

D. Create an Amazon DynamoDB table and a DynamoDB Accelerator (DAX) cluster for the table. Store frequently accessed data in the DynamoDB table.

Answer: A

Explanation:

Amazon ElastiCache is a service that offers fully managed in-memory data stores that are compatible with Redis or Memcached. The developer can create an ElastiCache for Redis instance and enable encryption of data in transit and at rest. This will ensure that the PHI is encrypted at all times. The developer can store frequently accessed data in the cache and use Redis features such as sorting and ranking to enhance the performance of the application.

Reference:

[What Is Amazon ElastiCache? - Amazon ElastiCache]

[Encryption in Transit - Amazon ElastiCache for Redis]

[Encryption at Rest - Amazon ElastiCache for Redis]

Question: 23

A company has a multi-node Windows legacy application that runs on premises. The application uses a network shared folder as a centralized configuration repository to store configuration files in .xml format. The company is migrating the application to Amazon EC2 instances. As part of the migration to AWS, a developer must identify a solution that provides high availability for the repository.

Which solution will meet this requirement MOST cost-effectively?

A. Mount an Amazon Elastic Block Store (Amazon EBS) volume onto one of the EC2 instances. Deploy a file system on the EBS volume. Use the host operating system to share a folder. Update the application code to read and write configuration files from the shared folder.

B. Deploy a micro EC2 instance with an instance store volume. Use the host operating system to share a folder. Update the application

code to read and write configuration files from the shared folder.

C. Create an Amazon S3 bucket to host the repository. Migrate the existing .xml files to the S3 bucket. Update the application code to use the AWS SDK to read and write configuration files from Amazon S3.

D. Create an Amazon S3 bucket to host the repository. Migrate the existing .xml files to the S3 bucket. Mount the S3 bucket to the EC2 instances as a local volume. Update the application code to read and write configuration files from the disk.

_____ Answer: C

Explanation:

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. The developer can create an S3 bucket to host the repository and migrate the existing .xml files to the S3 bucket. The developer can update the application code to use the AWS SDK to read and write configuration files from S3. This solution will meet the requirement of high availability for the repository in a cost-effective way.

Reference:

[Amazon Simple Storage Service (S3)]

[Using AWS SDKs with Amazon S3]

Question: 24

A company wants to deploy and maintain static websites on AWS. Each website's source code is hosted in one of several version control systems, including AWS CodeCommit, Bitbucket, and GitHub.

The company wants to implement phased releases by using development, staging, user acceptance testing, and production environments in the AWS Cloud. Deployments to each environment must be started by code merges on the relevant Git branch. The company wants to use HTTPS for all data exchange. The company needs a solution that does not require servers to run continuously.

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

A. Host each website by using AWS Amplify with a serverless backend. Conned the repository branches that correspond to each of the desired environments. Start deployments by merging code changes to a desired branch.

B. Host each website in AWS Elastic Beanstalk with multiple environments. Use the EB CLI to link each repository branch. Integrate AWS CodePipeline to automate deployments from version control code merges.

C. Host each website in different Amazon S3 buckets for each environment. Configure AWS CodePipeline to pull source code from version control. Add an AWS CodeBuild stage to copy source code to Amazon S3.

D. Host each website on its own Amazon EC2 instance. Write a custom deployment script to bundle each website's static assets. Copy the assets to Amazon EC2. Set up a workflow to run the script when code is merged.

_____ Answer: A

Explanation:

AWS Amplify is a set of tools and services that enables developers to build and deploy full-stack web and mobile applications that are powered by AWS. AWS Amplify supports hosting static websites on Amazon S3 and Amazon CloudFront, with HTTPS enabled by default. AWS Amplify also integrates with various version control systems, such as AWS CodeCommit, Bitbucket, and GitHub, and allows developers to connect different branches to different environments. AWS Amplify automatically builds and deploys the website whenever code changes are merged to a connected branch, enabling phased releases with minimal operational overhead. Reference: AWS Amplify Console

Question: 25

A company is migrating an on-premises database to Amazon RDS for MySQL. The company has readheavy workloads. The company wants to refactor the code to achieve optimum read performance for queries.

Which solution will meet this requirement with LEAST current and future effort?

A. Use a multi-AZ Amazon RDS deployment. Increase the number of connections that the code makes to the database or increase the connection pool size if a connection pool is in use.

B. Use a multi-AZ Amazon RDS deployment. Modify the code so that queries access the secondary RDS instance.

C. Deploy Amazon RDS with one or more read replicas. Modify the application code so that queries use the URL for the read replicas.

D. Use open source replication software to create a copy of the MySQL database on an Amazon EC2 instance. Modify the application code so that queries use the IP address of the EC2 instance.

Answer: C

Explanation:

Amazon RDS for MySQL supports read replicas, which are copies of the primary database instance that can handle read-only queries. Read replicas can improve the read performance of the database by ofloading the read workload from the primary instance and distributing it across multiple replicas. To use read replicas, the application code needs to be modified to direct read queries to the URL of the read replicas, while write queries still go to the URL of the primary instance. This solution requires less current and future effort than using a multi-AZ deployment, which does not provide read scaling benefits, or using open source replication software, which requires additional configuration and maintenance. Reference: Working with read replicas

Question: 26

A developer is creating an application that will be deployed on IoT devices. The application will send data to a RESTful API that is deployed as an AWS Lambda function. The application will assign each API request a unique identifier. The volume of API requests from the application can randomly increase at any given time of day.

During periods of request throttling, the application might need to retry requests. The API must be able to handle duplicate requests without inconsistencies or data loss.

Which solution will meet these requirements?

A. Create an Amazon RDS for MySQL DB instance. Store the unique identifier for each request in a database table. Modify the Lambda function to check the table for the identifier before processing the request.

B. Create an Amazon DynamoDB table. Store the unique identifier for each request in the table. Modify the Lambda function to check the table for the identifier before processing the request.

C. Create an Amazon DynamoDB table. Store the unique identifier for each request in the table. Modify the Lambda function to return a client error response when the function receives a duplicate request.

D. Create an Amazon ElastiCache for Memcached instance. Store the unique identifier for each request in the cache. Modify the Lambda function to check the cache for the identifier before processing the request.

_____ Answer: B

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that can store and retrieve any amount of data with high availability and performance. DynamoDB can handle concurrent requests from multiple IoT devices without throttling or data loss. To prevent duplicate requests from causing inconsistencies or data loss, the Lambda function can use DynamoDB conditional writes to check if the unique identifier for each request already exists in the table before processing the request. If the identifier exists, the function can skip or abort the request; otherwise, it can process the request and store the identifier in the table. Reference: Using conditional writes

Question: 27 _____

A developer wants to expand an application to run in multiple AWS Regions. The developer wants to copy Amazon Machine Images (AMIs) with the latest changes and create a new application stack in the destination Region. According to company requirements, all AMIs must be encrypted in all Regions. However, not all the AMIs that the company uses are encrypted.

How can the developer expand the application to run in the destination Region while meeting the encryption requirement?

A. Create new AMIs, and specify encryption parameters. Copy the encrypted AMIs to the destination Region. Delete the unencrypted AMIs.

B. Use AWS Key Management Service (AWS KMS) to enable encryption on the unencrypted AMIs. Copy the encrypted AMIs to the destination Region.

C. Use AWS Certificate Manager (ACM) to enable encryption on the unencrypted AMIs. Copy the encrypted AMIs to the destination Region.

D. Copy the unencrypted AMIs to the destination Region. Enable encryption by default in the destination Region.

_____ Answer: A

Explanation:

Amazon Machine Images (AMIs) are encrypted snapshots of EC2 instances that can be used to launch new instances. The developer can

create new AMIs from the existing instances and specify encryption parameters. The developer can copy the encrypted AMIs to the destination Region and use them to create a new application stack. The developer can delete the unencrypted AMIs after the encryption process is complete. This solution will meet the encryption requirement and allow the developer to expand the application to run in the destination Region.

Reference:

[Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud]

[Encrypting an Amazon EBS Snapshot - Amazon Elastic Compute Cloud]

[Copying an AMI - Amazon Elastic Compute Cloud]

Question: 28

A company hosts a client-side web application for one of its subsidiaries on Amazon S3. The web application can be accessed through Amazon CloudFront from https://www.example.com. After a successful rollout, the company wants to host three more client-side web applications for its remaining subsidiaries on three separate S3 buckets.

To achieve this goal, a developer moves all the common JavaScript files and web fonts to a central S3 bucket that serves the web applications. However, during testing, the developer notices that the browser blocks the JavaScript files and web fonts.

What should the developer do to prevent the browser from blocking the JavaScript files and web fonts?

A. Create four access points that allow access to the central S3 bucket. Assign an access point to each web application bucket.

B. Create a bucket policy that allows access to the central S3 bucket. Attach the bucket policy to the central S3 bucket.

C. Create a cross-origin resource sharing (CORS) configuration that allows access to the central S3 bucket. Add the CORS configuration to the central S3 bucket.

D. Create a Content-MD5 header that provides a message integrity check for the central S3 bucket. Insert the Content-MD5 header for each web application request.

Answer: C

Explanation:

This is a frequent trouble. Web applications cannot access the resources in other domains by default, except some exceptions. You must configure CORS on the resources to be accessed. https://docs.aws.amazon.com/AmazonS3/latest/userguide/cors.html

Question: 29

An application is processing clickstream data using Amazon Kinesis. The clickstream data feed into Kinesis experiences periodic spikes. The PutRecords API call occasionally fails and the logs show that the failed call returns the response shown below:

```
    "FailedRecordCount": 1,
    "Records": [
```

<
"SequenceNunber": "21269319989900637946712965403778482371", "Shardld": "shardId-000000000001"
},
(
"ErrorCode":"ProvisionedThroughputExceededException", "ErrorMessage": "Rate exceeded for shard shardId-000000000001 In
stream exaopleStreamNace under account 123456789."
b (
"SeguenceNumber": "21269319989999637946712965403778482985", "Shardld": "shardId-000000000002"
)
]

Which techniques will help mitigate this exception? (Choose two.)

A. Implement retries with exponential backoff.

B. Use a PutRecord API instead of PutRecords.

C. Reduce the frequency and/or size of the requests.

D. Use Amazon SNS instead of Kinesis.

E. Reduce the number of KCL consumers.

Answer: AC

Explanation:

The response from the API call indicates that the ProvisionedThroughputExceededException exception has occurred. This exception means that the rate of incoming requests exceeds the throughput limit for one or more shards in a stream. To mitigate this exception, the developer can use one or more of the following techniques:

Implement retries with exponential backoff. This will introduce randomness in the retry intervals and avoid overwhelming the shards with retries.

Reduce the frequency and/or size of the requests. This will reduce the load on the shards and avoid throttling errors.

Increase the number of shards in the stream. This will increase the throughput capacity of the stream and accommodate higher request rates.

Use a PutRecord API instead of PutRecords. This will reduce the number of records per request and avoid exceeding the payload limit.

Reference:

[ProvisionedThroughputExceededException - Amazon Kinesis Data Streams Service API Reference] [Best Practices for Handling Kinesis Data Streams Errors]

Question: 30

A company has an application that uses Amazon Cognito user pools as an identity provider. The company must secure access to user records. The company has set up multi-factor authentication (MFA). The company also wants to send a login activity notification by email every time a user logs in.

What is the MOST operationally efficient solution that meets this requirement?

A. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Add an Amazon API Gateway API to invoke the function. Call the API from the client side when login confirmation is received.

B. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Add an Amazon Cognito post authentication Lambda trigger for the function.

C. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification. Create an Amazon CloudWatch Logs log subscription filter to invoke the function based on the login status.

D. Configure Amazon Cognito to stream all logs to Amazon Kinesis Data Firehose. Create an AWS Lambda function to process the streamed logs and to send the email notification based on the login status of each user.

Answer: B

Explanation:

Amazon Cognito user pools support Lambda triggers, which are custom functions that can be executed at various stages of the user pool workflow. A post authentication Lambda trigger can be used to perform custom actions after a user is authenticated, such as sending an email notification. Amazon SES is a cloud-based email sending service that can be used to send transactional or marketing emails. A Lambda function can use the Amazon SES API to send an email to the user's email address after the user logs in successfully. Reference: Post authentication Lambda trigger

Question: 31

A developer has an application that stores data in an Amazon S3 bucket. The application uses an HTTP API to store and retrieve objects. When the PutObject API operation adds objects to the S3 bucket the developer must encrypt these objects at rest by using server-side encryption with Amazon S3 managed keys (SSE-S3).

Which solution will meet this requirement?

A. Create an AWS Key Management Service (AWS KMS) key. Assign the KMS key to the S3 bucket.

B. Set the x-amz-server-side-encryption header when invoking the PutObject API operation.

C. Provide the encryption key in the HTTP header of every request.

D. Apply TLS to encrypt the traffic to the S3 bucket.

Answer: B

Explanation:

Amazon S3 supports server-side encryption, which encrypts data at rest on the server that stores the data. One of the encryption options is SSE-S3, which uses keys managed by S3. To use SSE-S3, the x- amz-server-side-encryption header must be set to AES256 when invoking the PutObject API operation. This instructs S3 to encrypt the object data with SSE-S3 before saving it on disks in its data centers and decrypt it when it is downloaded. Reference: Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3)

Question: 32

A developer needs to perform geographic load testing of an API. The developer must deploy resources to multiple AWS Regions to support the load testing of the API.

How can the developer meet these requirements without additional application code?

A. Create and deploy an AWS Lambda function in each desired Region. Configure the Lambda function to create a stack from an AWS CloudFormation template in that Region when the function is invoked.

B. Create an AWS CloudFormation template that defines the load test resources. Use the AWS CLI create-stack-set command to create a stack set in the desired Regions.

C. Create an AWS Systems Manager document that defines the resources. Use the document to create the resources in the desired Regions.

D. Create an AWS CloudFormation template that defines the load test resources. Use the AWS CLI deploy command to create a stack from the template in each Region.

Answer: B

Explanation:

AWS CloudFormation is a service that allows developers to model and provision AWS resources using templates. A CloudFormation template can define the load test resources, such as EC2 instances, load balancers, and Auto Scaling groups. A CloudFormation stack set is a collection of stacks that can be created and managed from a single template in multiple Regions and accounts. The AWS CLI create-stack-set command can be used to create a stack set from a template and specify the Regions where the stacks should be created. Reference: Working with AWS CloudFormation stack sets

Question: 33

A developer is creating an application that includes an Amazon API Gateway REST API in the us-east- 2 Region. The developer wants to use Amazon CloudFront and a custom domain name for the API. The developer has acquired an SSL/TLS certificate for the domain from a third-party provider.

How should the developer configure the custom domain for the application?

A. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the API. Create a DNS A record for the custom domain.

B. Import the SSL/TLS certificate into CloudFront. Create a DNS CNAME record for the custom domain.

C. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the API. Create a DNS CNAME record for the custom domain.

D. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the us-east-1 Region. Create a DNS CNAME record for the custom domain.

Answer: D

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. Amazon CloudFront is a content delivery network (CDN) service that can improve the performance and security of web applications. The developer can use CloudFront and a custom domain name for the API Gateway REST API. To do so, the developer needs to import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the us-east-1 Region. This is because CloudFront requires certificates from ACM to be in this Region. The developer also needs to create a DNS CNAME record for the custom domain that points to the CloudFront distribution.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[What Is Amazon CloudFront? - Amazon CloudFront]

[Custom Domain Names for APIs - Amazon API Gateway]

Question: 34

A developer is creating a template that uses AWS CloudFormation to deploy an application. The application is serverless and uses Amazon API Gateway, Amazon DynamoDB, and AWS Lambda.

Which AWS service or tool should the developer use to define serverless resources in YAML?

A. CloudFormation serverless intrinsic functions

B. AWS Elastic Beanstalk

C. AWS Serverless Application Model (AWS SAM)

D. AWS Cloud Development Kit (AWS CDK)

Answer: C

Explanation:

AWS Serverless Application Model (AWS SAM) is an open-source framework that enables developers to build and deploy serverless applications on AWS. AWS SAM uses a template specification that extends AWS CloudFormation to simplify the definition of serverless resources such as API Gateway, DynamoDB, and Lambda. The developer can use AWS SAM to define serverless resources in YAML and deploy them using the AWS SAM CLI.

Reference:

[What Is the AWS Serverless Application Model (AWS SAM)? - AWS Serverless Application Model]

[AWS SAM Template Specification - AWS Serverless Application Model]

Question: 35

A developer wants to insert a record into an Amazon DynamoDB table as soon as a new file is added to an Amazon S3 bucket.

Which set of steps would be necessary to achieve this?

A. Create an event with Amazon EventBridge that will monitor the S3 bucket and then insert the records into DynamoDB.

B. Configure an S3 event to invoke an AWS Lambda function that inserts records into DynamoDB.

C. Create an AWS Lambda function that will poll the S3 bucket and then insert the records into DynamoDB.

D. Create a cron job that will run at a scheduled time and insert the records into DynamoDB.

Answer: B

Explanation:

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. AWS Lambda is a service that lets developers run code without provisioning or managing servers. The developer can configure an S3 event to invoke a Lambda function that inserts records into DynamoDB whenever a new file is added to the S3 bucket. This solution will meet the requirement of inserting a record into DynamoDB as soon as a new file is added to S3.

Reference:

[Amazon Simple Storage Service (S3)]

[Amazon DynamoDB]

[What Is AWS Lambda? - AWS Lambda]

[Using AWS Lambda with Amazon S3 - AWS Lambda]

Question: 36

A development team maintains a web application by using a single AWS CloudFormation template. The template defines web servers and an Amazon RDS database. The team uses the Cloud Formation template to deploy the Cloud Formation stack to different environments.

During a recent application deployment, a developer caused the primary development database to be dropped and recreated. The result of this incident was a loss of data. The team needs to avoid accidental database deletion in the future.

Which solutions will meet these requirements? (Choose two.)

A. Add a CloudFormation Deletion Policy attribute with the Retain value to the database resource.

B. Update the CloudFormation stack policy to prevent updates to the database.

C. Modify the database to use a Multi-AZ deployment.

D. Create a CloudFormation stack set for the web application and database deployments.

E. Add a Cloud Formation DeletionPolicy attribute with the Retain value to the stack.

Answer: A, B

Explanation:

AWS CloudFormation is a service that enables developers to model and provision AWS resources using templates. The developer can add a CloudFormation Deletion Policy attribute with the Retain value to the database resource. This will prevent the database from being deleted when the stack is deleted or updated. The developer can also update the CloudFormation stack policy to prevent updates to the database. This will prevent accidental changes to the database configuration or properties.

Reference:

[What Is AWS CloudFormation? - AWS CloudFormation]

[DeletionPolicy Attribute - AWS CloudFormation]

[Protecting Resources During Stack Updates - AWS CloudFormation]

Question: 37

A company has an Amazon S3 bucket that contains sensitive data. The data must be encrypted in transit and at rest. The company encrypts the data in the S3 bucket by using an AWS Key Management Service (AWS KMS) key. A developer needs to grant several other AWS accounts the permission to use the S3 GetObject operation to retrieve the data from the S3 bucket.

How can the developer enforce that all requests to retrieve the data provide encryption in transit?

A. Define a resource-based policy on the S3 bucket to deny access when a request meets the condition “aws:SecureTransport”: “false”.

B. Define a resource-based policy on the S3 bucket to allow access when a request meets the condition “aws:SecureTransport”: “false”.

C. Define a role-based policy on the other accounts' roles to deny access when a request meets the condition of

"aws:SecureTransport": "false".

D. Define a resource-based policy on the KMS key to deny access when a request meets the condition of "aws:SecureTransport": "false".

Answer: A

Explanation:

Amazon S3 supports resource-based policies, which are JSON documents that specify the permissions for accessing S3 resources. A resource-based policy can be used to enforce encryption in transit by denying access to requests that do not use HTTPS. The condition key aws:SecureTransport can be used to check if the request was sent using SSL. If the value of this key is false, the request is denied; otherwise, the request is allowed. Reference: How do I use an S3 bucket policy to require requests to use Secure Socket Layer (SSL)?

Question: 38

An application that is hosted on an Amazon EC2 instance needs access to files that are stored in an Amazon S3 bucket. The application lists the objects that are stored in the S3 bucket and displays a table to the user. During testing, a developer discovers that the application does not show any objects in the list.

What is the MOST secure way to resolve this issue?

A. Update the IAM instance profile that is attached to the EC2 instance to include the S3:* permission for the S3 bucket.

B. Update the IAM instance profile that is attached to the EC2 instance to include the S3:ListBucket permission for the S3 bucket.

C. Update the developer's user permissions to include the S3:ListBucket permission for the S3 bucket.

D. Update the S3 bucket policy by including the S3:ListBucket permission and by setting the Principal element to specify the account number of the EC2 instance.

Answer: B

Explanation:

IAM instance profiles are containers for IAM roles that can be associated with EC2 instances. An IAM role is a set of permissions that grant access to AWS resources. An IAM role can be used to allow an EC2 instance to access an S3 bucket by including the appropriate permissions in the role's policy. The S3:ListBucket permission allows listing the objects in an S3 bucket. By updating the IAM instance profile with this permission, the application on the EC2 instance can retrieve the objects from the S3 bucket and display them to the user. Reference: Using an IAM role to grant permissions to applications running on Amazon EC2 instances

Question: 39

A company is planning to securely manage one-time fixed license keys in AWS. The company's development team needs to access the license keys in automaton scripts that run in Amazon EC2 instances and in AWS CloudFormation stacks.

Which solution will meet these requirements MOST cost-effectively?

A. Amazon S3 with encrypted files prefixed with “config”

B. AWS Secrets Manager secrets with a tag that is named SecretString

C. AWS Systems Manager Parameter Store SecureString parameters

D. CloudFormation NoEcho parameters

Answer: C

Explanation:

AWS Systems Manager Parameter Store is a service that provides secure, hierarchical storage for configuration data and secrets. Parameter Store supports SecureString parameters, which are encrypted using AWS Key Management Service (AWS KMS) keys. SecureString parameters can be used to store license keys in AWS and retrieve them securely from automation scripts that run in EC2 instances or CloudFormation stacks. Parameter Store is a cost-effective solution because it does not charge for storing parameters or API calls. Reference: Working with Systems Manager parameters

Question: 40

A company has deployed infrastructure on AWS. A development team wants to create an AWS Lambda function that will retrieve data from an Amazon Aurora database. The Amazon Aurora database is in a private subnet in company's VPC. The VPC is named VPC1. The data is relational in nature. The Lambda function needs to access the data securely.

Which solution will meet these requirements?

A. Create the Lambda function. Configure VPC1 access for the function. Attach a security group named SG1 to both the Lambda function and the database. Configure the security group inbound and outbound rules to allow TCP traffic on Port 3306.

B. Create and launch a Lambda function in a new public subnet that is in a new VPC named VPC2. Create a peering connection between VPC1 and VPC2.

C. Create the Lambda function. Configure VPC1 access for the function. Assign a security group named SG1 to the Lambda function. Assign a second security group named SG2 to the database. Add an inbound rule to SG1 to allow TCP traffic from Port 3306.

D. Export the data from the Aurora database to Amazon S3. Create and launch a Lambda function in VPC1. Configure the Lambda function query the data from Amazon S3.

Answer: A

Explanation:

AWS Lambda is a service that lets you run code without provisioning or managing servers. Lambda functions can be configured to access resources in a VPC, such as an Aurora database, by specifying one or more subnets and security groups in the VPC settings of the function. A security group acts as a virtual firewall that controls inbound and outbound traffic for the resources in a VPC. To allow a Lambda function to communicate with an Aurora database, both

resources need to be associated with the same security group, and the security group rules need to allow TCP traffic on Port 3306, which is the default port for MySQL databases. Reference: [Configuring a Lambda function to access resources in a VPC]

Question: 41

A developer is building a web application that uses Amazon API Gateway to expose an AWS Lambda function to process requests from clients. During testing, the developer notices that the API Gateway times out even though the Lambda function finishes under the set time limit.

Which of the following API Gateway metrics in Amazon CloudWatch can help the developer troubleshoot the issue? (Choose two.)

A. CacheHitCount

B. IntegrationLatency

C. CacheMissCount

D. Latency

E. Count

Answer: B, D

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. Amazon CloudWatch is a service that monitors AWS resources and applications. API Gateway provides several CloudWatch metrics to help developers troubleshoot issues with their APIs. Two of the metrics that can help the developer troubleshoot the issue of API Gateway timing out are:

IntegrationLatency: This metric measures the time between when API Gateway relays a request to the backend and when it receives a response from the backend. A high value for this metric indicates that the backend is taking too long to respond and may cause API Gateway to time out.

Latency: This metric measures the time between when API Gateway receives a request from a client and when it returns a response to the client. A high value for this metric indicates that either the integration latency is high or API Gateway is taking too long to process the request or response.

Reference:

[What Is Amazon API Gateway? - Amazon API Gateway]

[Amazon API Gateway Metrics and Dimensions - Amazon CloudWatch]

[Troubleshooting API Errors - Amazon API Gateway]

Question: 42

A development team wants to build a continuous integration/continuous delivery (CI/CD) pipeline. The team is using AWS CodePipeline to automate the code build and deployment. The team wants to store the program code to prepare for the CI/CD pipeline.

Which AWS service should the team use to store the program code?

A. AWS CodeDeploy

B. AWS CodeArtifact

C. AWS CodeCommit

D. Amazon CodeGuru

Answer: C

Explanation:

AWS CodeCommit is a service that provides fully managed source control for hosting secure and scalable private Git repositories. The development team can use CodeCommit to store the program code and prepare for the CI/CD pipeline. CodeCommit integrates with other AWS services such as CodePipeline, CodeBuild, and CodeDeploy to automate the code build and deployment process.

Reference:

[What Is AWS CodeCommit? - AWS CodeCommit]

[AWS CodePipeline - AWS CodeCommit]

Question: 43

A developer is designing an AWS Lambda function that creates temporary files that are less than 10 MB during invocation. The temporary files will be accessed and modified multiple times during invocation. The developer has no need to save or retrieve these files in the future.

Where should the temporary files be stored?

A. the /tmp directory

B. Amazon Elastic File System (Amazon EFS)

C. Amazon Elastic Block Store (Amazon EBS)

D. Amazon S3

Answer: A

Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers. Lambda provides a local file system that can be used to store temporary files during invocation. The local file system is mounted under the /tmp directory and has a limit of 512 MB. The temporary files are accessible only by the Lambda function that created them and are deleted after the function execution ends.

The developer can store temporary files that are less than 10 MB in the /tmp directory and access and modify them multiple times during invocation.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Execution Environment - AWS Lambda]

Question: 44

A developer is designing a serverless application with two AWS Lambda functions to process photos. One Lambda function stores objects in an Amazon S3 bucket and stores the associated metadata in an Amazon DynamoDB table. The other Lambda function fetches the objects from the S3 bucket by using the metadata from the DynamoDB table. Both Lambda functions use the same Python library to perform complex computations and are approaching the quota for the maximum size of zipped deployment packages.

What should the developer do to reduce the size of the Lambda deployment packages with the LEAST operational overhead?

A. Package each Python library in its own .zip file archive. Deploy each Lambda function with its own copy of the library.

B. Create a Lambda layer with the required Python library. Use the Lambda layer in both Lambda functions.

C. Combine the two Lambda functions into one Lambda function. Deploy the Lambda function as a single .zip file archive.

D. Download the Python library to an S3 bucket. Program the Lambda functions to reference the object URLs.

Answer: B

Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers.

Lambda layers are a distribution mechanism for libraries, custom runtimes, and other dependencies. The developer can create a Lambda layer with the required Python library and use the layer in both Lambda functions. This will reduce the size of the Lambda deployment packages and avoid reaching the quota for the maximum size of zipped deployment packages. The developer can also benefit from using layers to manage dependencies separately from function code.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Layers - AWS Lambda]

Question: 45

A developer is writing an AWS Lambda function. The developer wants to log key events that occur while the Lambda function runs. The developer wants to include a unique identifier to associate the events with a specific function invocation. The developer adds the following code to the Lambda function:

```
function handler(event, context) {

}
```

Which solution will meet this requirement?

A. Obtain the request identifier from the AWS request ID field in the context object. Configure the application to write logs to standard output.

B. Obtain the request identifier from the AWS request ID field in the event object. Configure the application to write logs to a file.

C. Obtain the request identifier from the AWS request ID field in the event object. Configure the application to write logs to standard output.

D. Obtain the request identifier from the AWS request ID field in the context object. Configure the application to write logs to a file.

Answer: A

Explanation:

https://docs.aws.amazon.com/lambda/latest/dg/nodejs-context.htmlhttps://docs.aws.amazon.com/lambda/latest/dg/nodejs-logging.html

There is no explicit information for the runtime, the code is written in Node.js.

AWS Lambda is a service that lets developers run code without provisioning or managing servers. The developer can use the AWS request ID field in the context object to obtain a unique identifier for each function invocation. The developer can configure the application to write logs to standard output, which will be captured by Amazon CloudWatch Logs. This solution will meet the requirement of logging key events with a unique identifier.

Reference:

[What Is AWS Lambda? - AWS Lambda]

[AWS Lambda Function Handler in Node.js - AWS Lambda]

[Using Amazon CloudWatch - AWS Lambda]

Question: 46

A developer is working on a serverless application that needs to process any changes to an Amazon DynamoDB table with an AWS Lambda function.

How should the developer configure the Lambda function to detect changes to the DynamoDB table?

A. Create an Amazon Kinesis data stream, and attach it to the DynamoDB table. Create a trigger to connect the data stream to the Lambda function.

B. Create an Amazon EventBridge rule to invoke the Lambda function on a regular schedule. Conned to the DynamoDB table from the Lambda function to detect changes.

C. Enable DynamoDB Streams on the table. Create a trigger to connect the DynamoDB stream to the Lambda function.

D. Create an Amazon Kinesis Data Firehose delivery stream, and attach it to the DynamoDB table. Configure the delivery stream destination as the Lambda function.

Answer: C

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. DynamoDB Streams is a feature that captures data modification events in DynamoDB tables. The developer can enable DynamoDB Streams on the table and create a trigger to connect the DynamoDB stream to the Lambda function. This solution will enable the Lambda function to detect changes to the DynamoDB table in near real time.

Reference:

[Amazon DynamoDB]

[DynamoDB Streams - Amazon DynamoDB]

[Using AWS Lambda with Amazon DynamoDB - AWS Lambda]

Question: 47

An application uses an Amazon EC2 Auto Scaling group. A developer notices that EC2 instances are taking a long time to become available during scale-out events. The UserData script is taking a long time to run.

The developer must implement a solution to decrease the time that elapses before an EC2 instance becomes available. The solution must make the most recent version of the application available at all times and must apply all available security updates. The solution also must minimize the number of images that are created. The images must be validated.

Which combination of steps should the developer take to meet these requirements? (Choose two.)

A. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install all the patches and agents that are needed to manage and run the application. Update the Auto Scaling group launch configuration to use the AMI.

B. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install the latest version of the application and all the patches and agents that are needed to manage and run the application.
Update the Auto Scaling group launch configuration to use the AMI.

C. Set up AWS CodeDeploy to deploy the most recent version of the application at runtime.

D. Set up AWS CodePipeline to deploy the most recent version of the application at runtime.

E. Remove any commands that perform operating system patching from the UserData script.

Answer: BE

Explanation:

AWS CloudFormation is a service that enables developers to model and provision AWS resources using templates. The developer can use the following steps to avoid accidental database deletion in the future:

Set up AWS CodeDeploy to deploy the most recent version of the application at runtime. This will ensure that the application code is always up to date and does not depend on the AMI.

Remove any commands that perform operating system patching from the UserData script. This will reduce the time that the UserData script takes to run and speed up the instance launch process.

Reference:

[What Is AWS CloudFormation? - AWS CloudFormation]

[What Is AWS CodeDeploy? - AWS CodeDeploy]

[Running Commands on Your Linux Instance at Launch - Amazon Elastic Compute Cloud]

Question: 48

A developer is creating an AWS Lambda function that needs credentials to connect to an Amazon RDS for MySQL database. An Amazon S3 bucket currently stores the credentials. The developer needs to improve the existing solution by implementing credential rotation and secure storage. The developer also needs to provide integration with the Lambda function.

Which solution should the developer use to store and retrieve the credentials with the LEAST management overhead?

A. Store the credentials in AWS Systems Manager Parameter Store. Select the database that the parameter will access. Use the default AWS Key Management Service (AWS KMS) key to encrypt the parameter. Enable automatic rotation for the parameter. Use the parameter from Parameter Store on the Lambda function to connect to the database.

B. Encrypt the credentials with the default AWS Key Management Service (AWS KMS) key. Store the credentials as environment variables for the Lambda function. Create a second Lambda function to generate new credentials and to rotate the credentials by updating the environment variables of the first Lambda function. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule. Update the database to use the new credentials. On the first Lambda function, retrieve the credentials from the environment variables. Decrypt the credentials by using AWS KMS, Connect to the database.

C. Store the credentials in AWS Secrets Manager. Set the secret type to Credentials for Amazon RDS database. Select the database that the secret will access. Use the default AWS Key Management Service (AWS KMS) key to encrypt the secret. Enable automatic rotation for the secret. Use the secret from Secrets Manager on the Lambda function to connect to the database.

D. Encrypt the credentials by using AWS Key Management Service (AWS KMS). Store the credentials in an Amazon DynamoDB table.

Create a second Lambda function to rotate the credentials. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule. Update the DynamoDB table. Update the database to use the generated credentials. Retrieve the credentials from DynamoDB with the first Lambda function. Connect to the database.

Answer: C

Explanation:

AWS Secrets Manager is a service that helps you protect secrets needed to access your applications, services, and IT resources. Secrets Manager enables you to store, retrieve, and rotate secrets such as database credentials, API keys, and passwords. Secrets Manager supports a secret type for RDS databases, which allows you to select an existing RDS database instance and generate credentials for it. Secrets Manager encrypts the secret using AWS Key Management Service (AWS KMS) keys and enables automatic rotation of the secret at a specified interval. A Lambda function can use the AWS SDK or CLI to retrieve the secret from Secrets Manager and use it to connect to the database. Reference: Rotating your AWS Secrets Manager secrets

Question: 49

A developer has written the following IAM policy to provide access to an Amazon S3 bucket:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        b
        {
            "Effect": "Deny",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/secrets*"
        }
    ]
}
```

Which access does the policy allow regarding the s3:GetObject and s3:PutObject actions?

A. Access on all buckets except the “DOC-EXAMPLE-BUCKET” bucket

B. Access on all buckets that start with “DOC-EXAMPLE-BUCKET” except the “DOC-EXAMPLE- BUCKET/secrets” bucket

C. Access on all objects in the “DOC-EXAMPLE-BUCKET” bucket along with access to all S3 actions for objects in the “DOC-EXAMPLE-

BUCKET" bucket that start with "secrets"

D. Access on all objects in the "DOC-EXAMPLE-BUCKET" bucket except on objects that start with "secrets"

Answer: D

Explanation:

The IAM policy shown in the image is a resource-based policy that grants or denies access to an S3 bucket based on certain conditions. The first statement allows access to any S3 action on any object in the "DOC-EXAMPLE-BUCKET" bucket when the request is made over HTTPS (the value of aws:SecureTransport is true). The second statement denies access to the s3:GetObject and s3:PutObject actions on any object in the "DOC-EXAMPLE-BUCKET/secrets" prefix when the request is made over HTTP (the value of aws:SecureTransport is false). Therefore, the policy allows access on all objects in the "DOC-EXAMPLE-BUCKET" bucket except on objects that start with "secrets".

Reference: Using IAM policies for Amazon S3

Question: 50

A developer is creating a mobile app that calls a backend service by using an Amazon API Gateway REST API. For integration testing during the development phase, the developer wants to simulate different backend responses without invoking the backend service.

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

A. Create an AWS Lambda function. Use API Gateway proxy integration to return constant HTTP responses.

B. Create an Amazon EC2 instance that serves the backend REST API by using an AWS CloudFormation template.

C. Customize the API Gateway stage to select a response type based on the request.

D. Use a request mapping template to select the mock integration response.

Answer: D

Explanation:

Amazon API Gateway supports mock integration responses, which are predefined responses that can be returned without sending requests to a backend service. Mock integration responses can be used for testing or prototyping purposes, or for simulating different backend responses based on certain conditions. A request mapping template can be used to select a mock integration response based on an expression that evaluates some aspects of the request, such as headers, query strings, or body content. This solution does not require any additional resources or code changes and has the least operational overhead. Reference: Set up mock integrations for an API Gateway REST API https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html

Question: 51

A developer has a legacy application that is hosted on-premises. Other applications hosted on AWS depend on the on-premises application for proper functioning. In case of any application errors, the developer wants to be able to use Amazon CloudWatch to

monitor and troubleshoot all applications from one place.

How can the developer accomplish this?

A. Install an AWS SDK on the on-premises server to automatically send logs to CloudWatch.

B. Download the CloudWatch agent to the on-premises server. Configure the agent to use IAM user credentials with permissions for CloudWatch.

C. Upload log files from the on-premises server to Amazon S3 and have CloudWatch read the files.

D. Upload log files from the on-premises server to an Amazon EC2 instance and have the instance forward the logs to CloudWatch.

Answer: B

Explanation:

Amazon CloudWatch is a service that monitors AWS resources and applications. The developer can use CloudWatch to monitor and troubleshoot all applications from one place. To do so, the developer needs to download the CloudWatch agent to the on-premises server and configure the agent to use IAM user credentials with permissions for CloudWatch. The agent will collect logs and metrics from the on-premises server and send them to CloudWatch.

Reference:

[What Is Amazon CloudWatch? - Amazon CloudWatch]

[Installing and Configuring the CloudWatch Agent - Amazon CloudWatch]

Question: 52

An Amazon Kinesis Data Firehose delivery stream is receiving customer data that contains personally identifiable information. A developer needs to remove pattern-based customer identifiers from the data and store the modified data in an Amazon S3 bucket.

What should the developer do to meet these requirements?

A. Implement Kinesis Data Firehose data transformation as an AWS Lambda function. Configure the function to remove the customer identifiers. Set an Amazon S3 bucket as the destination of the delivery stream.

B. Launch an Amazon EC2 instance. Set the EC2 instance as the destination of the delivery stream. Run an application on the EC2 instance to remove the customer identifiers. Store the transformed data in an Amazon S3 bucket.

C. Create an Amazon OpenSearch Service instance. Set the OpenSearch Service instance as the destination of the delivery stream. Use search and replace to remove the customer identifiers. Export the data to an Amazon S3 bucket.

D. Create an AWS Step Functions workflow to remove the customer identifiers. As the last step in the workflow, store the transformed data in an Amazon S3 bucket. Set the workflow as the destination of the delivery stream.

Answer: A

Explanation:

Amazon Kinesis Data Firehose is a service that delivers real-time streaming data to destinations such as Amazon S3, Amazon Redshift, Amazon OpenSearch Service, and Amazon Kinesis Data Analytics. The developer can implement Kinesis Data Firehose data transformation as an AWS Lambda function. The function can remove pattern-based customer identifiers from the data and return the modified data to Kinesis Data Firehose. The developer can set an Amazon S3 bucket as the destination of the delivery stream.

Reference:

[What Is Amazon Kinesis Data Firehose? - Amazon Kinesis Data Firehose]

[Data Transformation - Amazon Kinesis Data Firehose]

Question: 53

A developer is using an AWS Lambda function to generate avatars for profile pictures that are uploaded to an Amazon S3 bucket. The Lambda function is automatically invoked for profile pictures that are saved under the /original/ S3 prefix. The developer notices that some pictures cause the Lambda function to time out. The developer wants to implement a fallback mechanism by using another Lambda function that resizes the profile picture.

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

A. Set the image resize Lambda function as a destination of the avatar generator Lambda function for the events that fail processing.

B. Create an Amazon Simple Queue Service (Amazon SQS) queue. Set the SQS queue as a destination with an on failure condition for the avatar generator Lambda function. Configure the image resize Lambda function to poll from the SQS queue.

C. Create an AWS Step Functions state machine that invokes the avatar generator Lambda function and uses the image resize Lambda function as a fallback. Create an Amazon EventBridge rule that matches events from the S3 bucket to invoke the state machine.

D. Create an Amazon Simple Notification Service (Amazon SNS) topic. Set the SNS topic as a destination with an on failure condition for the avatar generator Lambda function. Subscribe the image resize Lambda function to the SNS topic.

Answer: A

Explanation:

The solution that will meet the requirements with the least development effort is to set the image resize Lambda function as a destination of the avatar generator Lambda function for the events that fail processing. This way, the fallback mechanism is automatically triggered by the Lambda service without requiring any additional components or configuration. The other options involve creating and managing additional resources such as queues, topics, state machines, or rules, which would increase the complexity and cost of the solution.

Reference: Using AWS Lambda destinations

Question: 54

A developer needs to migrate an online retail application to AWS to handle an anticipated increase in traffic. The application currently runs on two servers: one server for the web application and another server for the database. The web server renders webpages and manages session state in memory.

The database server hosts a MySQL database that contains order details. When traffic to the application is heavy, the memory usage for the web server approaches 100% and the application slows down considerably.

The developer has found that most of the memory increase and performance decrease is related to the load of managing additional user sessions. For the web server migration, the developer will use Amazon EC2 instances with an Auto Scaling group behind an Application Load Balancer.

Which additional set of changes should the developer make to the application to improve the application's performance?

A. Use an EC2 instance to host the MySQL database. Store the session data and the application data in the MySQL database.

B. Use Amazon ElastiCache for Memcached to store and manage the session data. Use an Amazon RDS for MySQL DB instance to store the application data.

C. Use Amazon ElastiCache for Memcached to store and manage the session data and the application data.

D. Use the EC2 instance store to manage the session data. Use an Amazon RDS for MySQL DB instance to store the application data.

Answer: B

Explanation:

Using Amazon ElastiCache for Memcached to store and manage the session data will reduce the memory load and improve the performance of the web server. Using Amazon RDS for MySQL DB instance to store the application data will provide a scalable, reliable, and managed database service. Option A is not optimal because it does not address the memory issue of the web server. Option C is not optimal because it does not provide a persistent storage for the application data. Option D is not optimal because it does not provide a high availability and durability for the session data.

Reference: Amazon ElastiCache, Amazon RDS

Question: 55

An application uses Lambda functions to extract metadata from files uploaded to an S3 bucket; the metadata is stored in Amazon DynamoDB. The application starts behaving unexpectedly, and the developer wants to examine the logs of the Lambda function code for errors.

Based on this system configuration, where would the developer find the logs?

A. Amazon S3

B. AWS CloudTrail

C. Amazon CloudWatch

D. Amazon DynamoDB

Answer: C

Explanation:

Amazon CloudWatch is the service that collects and stores logs from AWS Lambda functions. The developer can use CloudWatch Logs Insights to query and analyze the logs for errors and metrics. Option A is not correct because Amazon S3 is a storage service that does not store Lambda function logs. Option B is not correct because AWS CloudTrail is a service that records API calls and events for AWS services, not Lambda function logs. Option D is not correct because Amazon DynamoDB is a database service that does not store Lambda function logs.

Reference: AWS Lambda Monitoring, [CloudWatch Logs Insights]

Question: 56

A company is using an AWS Lambda function to process records from an Amazon Kinesis data stream. The company recently observed slow processing of the records. A developer notices that the iterator age metric for the function is increasing and that the Lambda run duration is constantly above normal.

Which actions should the developer take to increase the processing speed? (Choose two.)

A. Increase the number of shards of the Kinesis data stream.

B. Decrease the timeout of the Lambda function.

C. Increase the memory that is allocated to the Lambda function.

D. Decrease the number of shards of the Kinesis data stream.

E. Increase the timeout of the Lambda function.

Answer: A, C

Explanation:

Increasing the number of shards of the Kinesis data stream will increase the throughput and parallelism of the data processing. Increasing the memory that is allocated to the Lambda function will also increase the CPU and network performance of the function, which will reduce the run duration and improve the processing speed. Option B is not correct because decreasing the timeout of the Lambda function will not affect the processing speed, but may cause some records to fail if they exceed the timeout limit. Option D is not correct because decreasing the number of shards of the Kinesis data stream will decrease the throughput and parallelism of the data processing, which will slow down the processing speed. Option E is not correct because increasing the timeout of the Lambda function will not affect the processing speed, but may increase the cost of running the function.

Reference: [Amazon Kinesis Data Streams Scaling], [AWS Lambda Performance Tuning]

Question: 57

A company needs to harden its container images before the images are in a running state. The company's application uses Amazon Elastic Container Registry (Amazon ECR) as an image registry. Amazon Elastic Kubernetes Service (Amazon EKS) for compute, and an AWS CodePipeline pipeline that orchestrates a continuous integration and continuous delivery (CI/CD) workflow.

Dynamic application security testing occurs in the final stage of the pipeline after a new image is deployed to a development namespace in the EKS cluster. A developer needs to place an analysis stage before this deployment to analyze the container image earlier in the CI/CD pipeline.

Which solution will meet these requirements with the MOST operational efficiency?

A. Build the container image and run the docker scan command locally. Mitigate any findings before pushing changes to the source code repository. Write a pre-commit hook that enforces the use of this workflow before commit.

B. Create a new CodePipeline stage that occurs after the container image is built. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.

C. Create a new CodePipeline stage that occurs after source code has been retrieved from its repository. Run a security scanner on the latest revision of the source code. Fail the pipeline if there are findings.

D. Add an action to the deployment stage of the pipeline so that the action occurs before the deployment to the EKS cluster. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.

Answer: B

Explanation:

The solution that will meet the requirements with the most operational efficiency is to create a new CodePipeline stage that occurs after the container image is built. Configure ECR basic image scanning to scan on image push. Use an AWS Lambda function as the action provider. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings. This way, the container image is analyzed earlier in the CI/CD pipeline and any vulnerabilities are detected and reported before deploying to the EKS cluster. The other options either delay the analysis until after deployment, which increases the risk of exposing insecure images, or perform analysis on the source code instead of the container image, which may not capture all the dependencies and configurations that affect the security posture of the image.

Reference: Amazon ECR image scanning

Question: 58

A developer is testing a new file storage application that uses an Amazon CloudFront distribution to serve content from an Amazon S3 bucket. The distribution accesses the S3 bucket by using an origin access identity (OAI). The S3 bucket's permissions explicitly deny access to all other users.

The application prompts users to authenticate on a login page and then uses signed cookies to allow users to access their personal

storage directories. The developer has configured the distribution to use its default cache behavior with restricted viewer access and has set the origin to point to the S3 bucket. However, when the developer tries to navigate to the login page, the developer receives a 403 Forbidden error.

The developer needs to implement a solution to allow unauthenticated access to the login page. The solution also must keep all private content secure.

Which solution will meet these requirements?

A. Add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted. Keep the default cache behavior's settings unchanged.

B. Add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to *, and make viewer access restricted. Change the default cache behavior's path pattern to the path of the login page, and make viewer access unrestricted.

C. Add a second origin as a failover origin to the default cache behavior. Point the failover origin to the S3 bucket. Set the path pattern for the primary origin to *, and make viewer access restricted. Set the path pattern for the failover origin to the path of the login page, and make viewer access unrestricted.

D. Add a bucket policy to the S3 bucket to allow read access. Set the resource on the policy to the Amazon Resource Name (ARN) of the login page object in the S3 bucket. Add a CloudFront function to the default cache behavior to redirect unauthorized requests to the login page's S3 URL.

Answer: A

Explanation:

The solution that will meet the requirements is to add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted. Keep the default cache behavior's settings unchanged. This way, the login page can be accessed without authentication, while all other content remains secure and requires signed cookies. The other options either do not allow unauthenticated access to the login page, or expose private content to unauthorized users.

Reference: Restricting Access to Amazon S3 Content by Using an Origin Access Identity

Question: 59

A developer is using AWS Amplify Hosting to build and deploy an application. The developer is receiving an increased number of bug reports from users. The developer wants to add end-to-end testing to the application to eliminate as many bugs as possible before the bugs reach production.

Which solution should the developer implement to meet these requirements?

A. Run the amplify add test command in the Amplify CLI.

B. Create unit tests in the application. Deploy the unit tests by using the amplify push command in the Amplify CLI.

C. Add a test phase to the amplify.yml build settings for the application.

D. Add a test phase to the aws-exports.js file for the application.

Answer: C

Explanation:

The solution that will meet the requirements is to add a test phase to the amplify.yml build settings for the application. This way, the developer can run end-to-end tests on every code commit and catch any bugs before deploying to production. The other options either do not support end-to-end testing, or do not run tests automatically.

Reference: End-to-end testing

Question: 60

An ecommerce company is using an AWS Lambda function behind Amazon API Gateway as its application tier. To process orders during checkout, the application calls a POST API from the frontend. The POST API invokes the Lambda function asynchronously. In rare situations, the application has not processed orders. The Lambda application logs show no errors or failures.

What should a developer do to solve this problem?

A. Inspect the frontend logs for API failures. Call the POST API manually by using the requests from the log file.

B. Create and inspect the Lambda dead-letter queue. Troubleshoot the failed functions. Reprocess the events.

C. Inspect the Lambda logs in Amazon CloudWatch for possible errors. Fix the errors.

D. Make sure that caching is disabled for the POST API in API Gateway.

Answer: B

Explanation:

The solution that will solve this problem is to create and inspect the Lambda dead-letter queue. Troubleshoot the failed functions. Reprocess the events. This way, the developer can identify and fix any issues that caused the Lambda function to fail when invoked asynchronously by API Gateway. The developer can also reprocess any orders that were not processed due to failures. The other options either do not address the root cause of the problem, or do not help recover from failures.

Reference: Asynchronous invocation

Question: 61

A company is building a web application on AWS. When a customer sends a request, the application will generate reports and then make the reports available to the customer within one hour. Reports should be accessible to the customer for 8 hours. Some reports are larger than 1 MB. Each report is unique to the customer. The application should delete all reports that are older than 2 days.

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

A. Generate the reports and then store the reports as Amazon DynamoDB items that have a specified TTL. Generate a URL that retrieves the reports from DynamoDB. Provide the URL to customers through the web application.

B. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption. Attach the reports to an Amazon Simple Notification Service (Amazon SNS) message. Subscribe the customer to email notifications from Amazon SNS.

C. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption. Generate a presigned URL that contains an expiration date Provide the URL to customers through the web application. Add S3 Lifecycle configuration rules to the S3 bucket to delete old reports.

D. Generate the reports and then store the reports in an Amazon RDS database with a date stamp. Generate an URL that retrieves the reports from the RDS database. Provide the URL to customers through the web application. Schedule an hourly AWS Lambda function to delete database records that have expired date stamps.

Answer: C

Explanation:

This solution will meet the requirements with the least operational overhead because it uses Amazon S3 as a scalable, secure, and durable storage service for the reports. The presigned URL will allow customers to access their reports for a limited time (8 hours) without requiring additional authentication. The S3 Lifecycle configuration rules will automatically delete the reports that are older than 2 days, reducing storage costs and complying with the data retention policy. Option A is not optimal because it will incur additional costs and complexity to store the reports as DynamoDB items, which have a size limit of 400 KB. Option B is not optimal because it will not provide customers with access to their reports within one hour, as Amazon SNS email delivery is not guaranteed. Option D is not optimal because it will require more operational overhead to manage an RDS database and a Lambda function for storing and deleting the reports.

Reference: Amazon S3 Presigned URLs, Amazon S3 Lifecycle

Question: 62

A company has deployed an application on AWS Elastic Beanstalk. The company has configured the Auto Scaling group that is associated with the Elastic Beanstalk environment to have five Amazon EC2 instances. If the capacity is fewer than four EC2 instances during the deployment, application performance degrades. The company is using the all-at-once deployment policy.

What is the MOST cost-effective way to solve the deployment issue?

A. Change the Auto Scaling group to six desired instances.

B. Change the deployment policy to traffic splitting. Specify an evaluation time of 1 hour.

C. Change the deployment policy to rolling with additional batch. Specify a batch size of 1.

D. Change the deployment policy to rolling. Specify a batch size of 2.

Answer: C

Explanation:

This solution will solve the deployment issue by deploying the new version of the application to one new EC2 instance at a time, while keeping the old version running on the existing instances. This way, there will always be at least four instances serving traffic during the deployment, and no downtime or performance degradation will occur. Option A is not optimal because it will increase the cost of running the Elastic Beanstalk environment without solving the deployment issue. Option B is not optimal because it will split the traffic between two versions of the application, which may cause inconsistency and confusion for the customers. Option D is not optimal because it will deploy the new version of the application to two existing instances at a time, which may reduce the capacity below four instances during the deployment.

Reference: AWS Elastic Beanstalk Deployment Policies

Question: 63

A developer is incorporating AWS X-Ray into an application that handles personal identifiable information (PII). The application is hosted on Amazon EC2 instances. The application trace messages include encrypted PII and go to Amazon CloudWatch. The developer needs to ensure that no PII goes outside of the EC2 instances.

Which solution will meet these requirements?

A. Manually instrument the X-Ray SDK in the application code.

B. Use the X-Ray auto-instrumentation agent.

C. Use Amazon Macie to detect and hide PII. Call the X-Ray API from AWS Lambda.

D. Use AWS Distro for Open Telemetry.

Answer: A

Explanation:

This solution will meet the requirements by allowing the developer to control what data is sent to X- Ray and CloudWatch from the application code. The developer can filter out any PII from the trace messages before sending them to X-Ray and CloudWatch, ensuring that no PII goes outside of the EC2 instances. Option B is not optimal because it will automatically instrument all incoming and outgoing requests from the application, which may include PII in the trace messages. Option C is not optimal because it will require additional services and costs to use Amazon Macie and AWS Lambda, which may not be able to detect and hide all PII from the trace messages. Option D is not optimal because it will use Open Telemetry instead of X-Ray, which may not be compatible with CloudWatch and other AWS services.

Reference: [AWS X-Ray SDKs]

Question: 64

A developer is migrating some features from a legacy monolithic application to use AWS Lambda functions instead. The application

currently stores data in an Amazon Aurora DB cluster that runs in private subnets in a VPC. The AWS account has one VPC deployed. The Lambda functions and the DB cluster are deployed in the same AWS Region in the same AWS account.

The developer needs to ensure that the Lambda functions can securely access the DB cluster without crossing the public internet.

Which solution will meet these requirements?

A. Configure the DB cluster's public access setting to Yes.

B. Configure an Amazon RDS database proxy for the Lambda functions.

C. Configure a NAT gateway and a security group for the Lambda functions.

D. Configure the VPC, subnets, and a security group for the Lambda functions.

Answer: D

Explanation:

This solution will meet the requirements by allowing the Lambda functions to access the DB cluster securely within the same VPC without crossing the public internet. The developer can configure a VPC endpoint for RDS in a private subnet and assign it to the Lambda functions. The developer can also configure a security group for the Lambda functions that allows inbound traffic from the DB cluster on port 3306 (MySQL). Option A is not optimal because it will expose the DB cluster to public access, which may compromise its security and data integrity. Option B is not optimal because it will introduce additional latency and complexity to use an RDS database proxy for accessing the DB cluster from Lambda functions within the same VPC. Option C is not optimal because it will require additional costs and configuration to use a NAT gateway for accessing resources in private subnets from Lambda functions.

Reference: [Configuring a Lambda Function to Access Resources in a VPC]

Question: 65

A developer is building a new application on AWS. The application uses an AWS Lambda function that retrieves information from an Amazon DynamoDB table. The developer hard coded the DynamoDB table name into the Lambda function code. The table name might change over time. The developer does not want to modify the Lambda code if the table name changes.

Which solution will meet these requirements MOST efficiently?

A. Create a Lambda environment variable to store the table name. Use the standard method for the programming language to retrieve the variable.

B. Store the table name in a file. Store the file in the /tmp folder. Use the SDK for the programming language to retrieve the table name.

C. Create a file to store the table name. Zip the file and upload the file to the Lambda layer. Use the SDK for the programming language to retrieve the table name.

D. Create a global variable that is outside the handler in the Lambda function to store the table name.

Answer: A

Explanation:

The solution that will meet the requirements most efficiently is to create a Lambda environment variable to store the table name. Use the standard method for the programming language to retrieve the variable. This way, the developer can avoid hard-coding the table name in the Lambda function code and easily change the table name by updating the environment variable. The other options either involve storing the table name in a file, which is less efficient and secure than using an environment variable, or creating a global variable, which is not recommended as it can cause concurrency issues.

Reference: Using AWS Lambda environment variables

Question: 66

A company has installed smart motes in all Its customer locations. The smart meter's measure power usage at 1minute intervals and send the usage readings to a remote endpoint tot collection. The company needs to create an endpoint that will receive the smart meter readings and store the readings in a database. The company wants to store the location ID and timestamp information.

The company wants to give Is customers low-latency access to their current usage and historical usage on demand The company expects demand to increase significantly. The solution must not impact performance or include downtime write seeing.

When solution will meet these requirements MOST cost-effectively?

A. Store the smart meter readings in an Amazon RDS database. Create an index on the location ID and timestamp columns Use the columns to filter on the customers 'data.

B. Store the smart motor readings m an Amazon DynamoDB table Croato a composite Key oy using the location ID and timestamp columns. Use the columns to filter on the customers' data.

C. Store the smart meter readings in Amazon EastCache for Reds Create a Sorted set key y using the location ID and timestamp columns. Use the columns to filter on the customers' data.

D. Store the smart meter readings m Amazon S3 Parton the data by using the location ID and timestamp columns. Use Amazon Athena lo tiler on me customers' data.

Answer: B

Explanation:

The solution that will meet the requirements most cost-effectively is to store the smart meter readings in an Amazon DynamoDB table. Create a composite key by using the location ID and timestamp columns. Use the columns to filter on the customers' data. This way, the company can leverage the scalability, performance, and low latency of DynamoDB to store and retrieve the smart meter readings. The company can also use the composite key to query the data by location ID and timestamp efficiently. The other options either involve more expensive or less scalable services, or do not provide low-latency access to the current usage.

Reference: Working with Queries in DynamoDB

Question: 67

A company's website runs on an Amazon EC2 instance and uses Auto Scaling to scale the environment during peak

times. Website users across the world ate experiencing high latency flue lo sialic content on theEC2 instance. even during non-peak hours.

When companion of steps mill resolves the latency issue? (Select TWO)

A. Double the Auto Scaling group's maximum number of servers

B. Host the application code on AWS lambda

C. Scale vertically by resizing the EC2 instances

D. Create an Amazon Cloudfront distribution to cache the static content

E. Store the application's sialic content in Amazon S3

Answer: DE

Explanation:

The combination of steps that will resolve the latency issue is to create an Amazon CloudFront distribution to cache the static content and store the application's static content in Amazon S3. This way, the company can use CloudFront to deliver the static content from edge locations that are closer to the website users, reducing latency and improving performance. The company can also use S3 to store the static content reliably and cost-effectively, and integrate it with CloudFront easily. The other options either do not address the latency issue, or are not necessary or feasible for the given scenario.

Reference: Using Amazon S3 Origins and Custom Origins for Web Distributions

Question: 68

An online food company provides an Amazon API Gateway HTTP API 1o receive orders for partners. The API is integrated with an AWS Lambda function. The Lambda function stores the orders in an Amazon DynamoDB table.

The company expects to onboard additional partners Some to me panthers require additional Lambda function to receive orders. The company has created an Amazon S3 bucket. The company needs 10 store all orders and updates m the S3 bucket for future analysis

How can the developer ensure that an orders and updates are stored to Amazon S3 with the LEAST development effort?

A. Create a new Lambda function and a new API Gateway API endpoint. Configure the new Lambda function to write to the S3 bucket. Modify the original Lambda function to post updates to the new API endpoint.

B. Use Amazon Kinesis Data Streams to create a new data stream. Modify the Lambda function to publish orders to the oats stream Configure in data stream to write to the S3 bucket.

C. Enable DynamoDB Streams on me DynamoOB table. Create a new lambda function. Associate the stream's Amazon Resource Name (ARN) with the Lambda Function Configure the Lambda function to write to the S3 bucket as records appear in the table's stream.

D. Modify the Lambda function to punish to a new Amazon. Simple Lambda function receives orders. Subscribe a new Lambda function to the topic. Configure the new Lambda function to write to the S3 bucket as updates come through the topic.

Answer: C

Explanation:

This solution will ensure that all orders and updates are stored to Amazon S3 with the least development effort because it uses DynamoDB Streams to capture changes in the DynamoDB table and trigger a Lambda function to write those changes to the S3 bucket. This way, the original Lambda function and API Gateway API endpoint do not need to be modified, and no additional services are required. Option A is not optimal because it will require more development effort to create a new Lambda function and a new API Gateway API endpoint, and to modify the original Lambda function to post updates to the new API endpoint. Option B is not optimal because it will introduce additional costs and complexity to use Amazon Kinesis Data Streams to create a new data stream, and to modify the Lambda function to publish orders to the data stream. Option D is not optimal because it will require more development effort to modify the Lambda function to publish to a new Amazon SNS topic, and to create and subscribe a new Lambda function to the topic.

Reference: Using DynamoDB Streams, Using AWS Lambda with Amazon S3

Question: 69

A company has an Amazon S3 bucket containing premier content that it intends to make available to only paid subscribers of its website. The S3 bucket currently has default permissions of all objects being private to prevent inadvertent exposure of the premier content to non-paying website visitors.

How can the company Limit the ability to download a premier content file in the S3 Bucket to paid subscribers only?

A. Apply a bucket policy that allows anonymous users to download the content from the S3 bucket.

B. Generate a pre-signed object URL for the premier content file when a pad subscriber requests a download.

C. Add a Docket policy that requires multi-factor authentication for request to access the S3 bucket objects.

D. Enable server-side encryption on the S3 bucket for data protection against the non-paying website visitors.

Answer: B

Explanation:

This solution will limit the ability to download a premier content file in the S3 bucket to paid subscribers only because it uses a pre-signed object URL that grants temporary access to an S3 object for a specified duration. The pre-signed object URL can be generated by the company's website when a paid subscriber requests a download, and can be verified by Amazon S3 using the signature in the URL. Option A is not optimal because it will allow anyone to download the content from the S3 bucket without verifying their subscription status. Option C is not optimal because it will require additional steps and costs to configure multi-factor authentication for accessing the S3 bucket objects, which may not be feasible or user-friendly for paid subscribers. Option D is not optimal because it will not prevent non-paying website visitors from accessing the S3 bucket objects, but only encrypt them at rest.

Reference: Share an Object with Others, [Using Amazon S3 Pre-Signed URLs]

Question: 70

A developer is creating an AWS Lambda function that searches for Items from an Amazon DynamoDQ table that contains customer contact information. The DynamoDB table items have the customers as the partition and additional properties such as customer -type, name, and job_title.

The Lambda function runs whenever a user types a new character into the customer_type text Input. The developer wants to search to return partial matches of all tne email_address property of a particular customer type. The developer does not want to recreate the DynamoDB table.

What should the developer do to meet these requirements?

A. Add a global secondary index (GSI) to the DynamoDB table with customer-type input, as the partition key and email_address as the sort key. Perform a query operation on the GSI by using the begins with key condition expression with the email_address property.

B. Add a global secondary index (GSI) to the DynamoDB table with email_address as the partition key and customer_type as the sort key. Perform a query operation on the GSI by using the begine_with key condition expresses with the email. Address property.

C. Add a local secondary index (LSI) to the DynemoOB table with customer_type as the partition Key and email_address as the sort Key. Perform a quick operation on the LSI by using the begine_with Key condition expression with the email-address property.

D. Add a local secondary index (LSI) to the DynamoDB table with job-title as the partition key and email_address as the sort key. Perform a query operation on the LSI by using the begins_with key condition expression with the email_address property.

Answer: A

Explanation:

The solution that will meet the requirements is to add a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property. This way, the developer can search for partial matches of the email_address property of a particular customer type without recreating the DynamoDB table. The other options either involve using a local secondary index (LSI), which requires recreating the table, or using a different partition key, which does not allow filtering by customer_type.

Reference: Using Global Secondary Indexes in DynamoDB

Question: 71

A developer is building an application that uses AWS API Gateway APIs. AWS Lambda function, and AWS Dynamic DB tables. The developer uses the AWS Serverless Application Model (AWS SAM) to build and run serverless applications on AWS. Each time the developer pushes of changes for only to the Lambda functions, all the artifacts in the application are rebuilt.

The developer wants to implement AWS SAM Accelerate by running a command to only redeploy the Lambda functions that have changed.

Which command will meet these requirements?

A. sam deploy -force-upload

B. sam deploy -no-execute-changeset

C. sam package

D. sam sync -watch

Answer: D

Explanation:

The command that will meet the requirements is sam sync -watch. This command enables AWS SAM Accelerate mode, which allows the developer to only redeploy the Lambda functions that have changed. The -watch flag enables file watching, which automatically detects changes in the source code and triggers a redeployment. The other commands either do not enable AWS SAM Accelerate mode, or do not redeploy the Lambda functions automatically.

Reference: AWS SAM Accelerate

Question: 72

A developer is building an application that gives users the ability to view bank account from multiple sources in a single dashboard. The developer has automated the process to retrieve API credentials for these sources. The process invokes an AWS Lambda function that is associated with an AWS CloudFormation cotton resource.

The developer wants a solution that will store the API credentials with minimal operational overhead.

When solution will meet these requirements?

A. Add an AWS Secrets Manager GenerateSecretString resource to the CloudFormation template. Set the value to reference new credentials to the Cloudformation resource.

B. Use the AWS SDK ssm PutParameter operation in the Lambda function from the existing, custom resource to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set ma parameter type to SecureString.

C. Add an AWS Systems Manager Parameter Store resource to the CloudFormation template. Set the CloudFormation resource value to reference the new credentials Set the resource NoEcho attribute to true.

D. Use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resources to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set the parameter NoEcho attribute to true.

Answer: B

Explanation:

The solution that will meet the requirements is to use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resource to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set the

parameter type to SecureString. This way, the developer can store the API credentials with minimal operational overhead, as AWS Systems Manager Parameter Store provides secure and scalable storage for configuration data. The SecureString parameter type encrypts the parameter value with AWS Key Management Service (AWS KMS). The other options either involve adding additional resources to the CloudFormation template, which increases complexity and cost, or do not encrypt the parameter value, which reduces security.

Reference: Creating Systems Manager parameters

Question: 73

A developer is configuring an applications deployment environment in AWS CodePipeine. The application code is stored in a GitHub repository. The developer wants to ensure that the repository package's unit tests run in the new deployment environment. The deployment has already set the pipeline's source provider to GitHub and has specified the repository and branch to use in the deployment.

When combination of steps should the developer take next to meet these requirements with the least the LEAST overhead' (Select TWO).

A. Create an AWS CodeCommt project. Add the repository package's build and test commands to the protects buildspec

B. Create an AWS CodeBuid project. Add the repository package's build and test commands to the projects buildspec

C. Create an AWS CodeDeploy protect. Add the repository package's build and test commands to the project's buildspec

D. Add an action to the source stage. Specify the newly created project as the action provider. Specify the build attract as the actions input artifact.

E. Add a new stage to the pipeline alter the source stage. Add an action to the new stage. Speedy the newly created protect as the action provider. Specify the source artifact as the action's input artifact.

Answer: B, E

Explanation:

This solution will ensure that the repository package's unit tests run in the new deployment environment with the least overhead because it uses AWS CodeBuild to build and test the code in a fully managed service, and AWS CodePipeline to orchestrate the deployment stages and actions. Option A is not optimal because it will use AWS CodeCommit instead of AWS CodeBuild, which is a source control service, not a build and test service. Option C is not optimal because it will use AWS CodeDeploy instead of AWS CodeBuild, which is a deployment service, not a build and test service. Option D is not optimal because it will add an action to the source stage instead of creating a new stage, which will not follow the best practice of separating different deployment phases.

Reference: AWS CodeBuild, AWS CodePipeline

Question: 74

A developer is trying get data from an Amazon DynamoDB table called demoman-table. The developer configured the AWS CLI to use a

specific IAM use's credentials and ran the following command.

The command returned errors and no rows were returned.

What is the MOST likely cause of these issues?

A. The command is incorrect; it should be rewritten to use put-item with a string argument

B. The developer needs to log a ticket with AWS Support to enable access to the demoman-table

C. Amazon DynamoOB cannot be accessed from the AWS CLI and needs to called via the REST API

D. The IAM user needs an associated policy with read access to demoman-table

Answer: D

Explanation:

This solution will most likely solve the issues because it will grant the IAM user the necessary permission to access the DynamoDB table using the AWS CLI command. The error message indicates that the IAM user does not have sufficient access rights to perform the scan operation on the table. Option A is not optimal because it will change the command to use put-item instead of scan, which will not achieve the desired result of getting data from the table. Option B is not optimal because it will involve contacting AWS Support, which may not be necessary or efficient for this issue. Option C is not optimal because it will state that DynamoDB cannot be accessed from the AWS CLI, which is incorrect as DynamoDB supports AWS CLI commands.

Reference: AWS CLI for DynamoDB, [IAM Policies for DynamoDB]

Question: 75

An organization is using Amazon CloudFront to ensure that its users experience low-latency access to its web application. The organization has identified a need to encrypt all traffic between users and CloudFront, and all traffic between CloudFront and the web application.

How can these requirements be met? (Select TWO)

A. Use AWS KMS t0 encrypt traffic between cloudFront and the web application.

B. Set the Origin Protocol Policy to "HTTPS Only".

C. Set the Origin's HTTP Port to 443.

D. Set the Viewer Protocol Policy to "HTTPS Only" or Redirect HTTP to HTTPS"

E. Enable the CloudFront option Restrict Viewer Access.

Answer: B, D

Explanation:

This solution will meet the requirements by ensuring that all traffic between users and CloudFront, and all traffic between CloudFront and the web application, are encrypted using HTTPS protocol. The Origin Protocol Policy determines how CloudFront communicates with the origin server (the web application), and setting it to "HTTPS Only" will force CloudFront to use HTTPS for every request to the origin server. The Viewer Protocol Policy determines how CloudFront responds to HTTP or HTTPS requests from users, and setting it to "HTTPS Only" or "Redirect HTTP to HTTPS" will force CloudFront to use HTTPS for every response to users. Option A is not optimal because it will use AWS KMS to encrypt traffic between CloudFront and the web application, which is not necessary or supported by CloudFront. Option C is not optimal because it will set the origin's HTTP port to 443, which is incorrect as port 443 is used for HTTPS protocol, not HTTP protocol. Option E is not optimal because it will enable the CloudFront option Restrict Viewer Access, which is used for controlling access to private content using signed URLs or signed cookies, not for encrypting traffic.

Reference: [Using HTTPS with CloudFront], [Restricting Access to Amazon S3 Content by Using an Origin Access Identity]

Question: 76

A company is developing an ecommerce application that uses Amazon API Gateway APIs. The application uses AWS Lambda as a backend. The company needs to test the code in a dedicated, monitored test environment before the company releases the code to the production environment.

When solution will meet these requirements?

A. Use a single stage in API Gateway. Create a Lambda function for each environment. Configure API clients to send a query parameter that indicates the endowment and the specific lambda function.

B. Use multiple stages in API Gateway. Create a single Lambda function for all environments. Add different code blocks for different environments in the Lambda function based on Lambda environments variables.

C. Use multiple stages in API Gateway. Create a Lambda function for each environment. Configure API Gateway stage variables to route traffic to the Lambda function in different environments.

D. Use a single stage in API Gateway. Configure a API client to send a query parameter that indicated the environment. Add different code blocks tor afferent environments in the Lambda Junction to match the value of the query parameter.

Answer: C

Explanation:

The solution that will meet the requirements is to use multiple stages in API Gateway. Create a Lambda function for each environment. Configure API Gateway stage variables to route traffic to the Lambda function in different environments. This way, the company can test the code in a dedicated, monitored test environment before releasing it to the production environment. The company can also use stage variables to specify the Lambda function version or alias for each stage, and avoid hard-coding the Lambda function name in the API Gateway integration. The other options either involve using a single stage in API Gateway, which does not allow testing in different environments, or adding different code blocks for different environments in the Lambda function, which increases complexity and maintenance.

Reference: Set up stage variables for a REST API in API Gateway

Question: 77

A developer is planning to migrate on-premises company data to Amazon S3. The data must be encrypted, and the encryption Keys must support automate annual rotation. The company must use AWS Key Management Service (AWS KMS) to encrypt the data.

When type of keys should the developer use to meet these requirements?

A. Amazon S3 managed keys

B. Symmetric customer managed keys with key material that is generated by AWS

C. Asymmetric customer managed keys with key material that generated by AWS

D. Symmetric customer managed keys with imported key material

Answer: B

Explanation:

The type of keys that the developer should use to meet the requirements is symmetric customer managed keys with key material that is generated by AWS. This way, the developer can use AWS Key Management Service (AWS KMS) to encrypt the data with a symmetric key that is managed by the developer. The developer can also enable automatic annual rotation for the key, which creates new key material for the key every year. The other options either involve using Amazon S3 managed keys, which do not support automatic annual rotation, or using asymmetric keys or imported key material, which are not supported by S3 encryption.

Reference: Using AWS KMS keys to encrypt S3 objects

Question: 78

A team of developed is using an AWS CodePipeline pipeline as a continuous integration and continuous delivery (CI/CD) mechanism for a web application. A developer has written unit tests to programmatically test the functionality of the application code. The unit tests produce a test report that shows the results of each individual check. The developer now wants to run these tests automatically during the CI/CD process.

A. Write a Git pre-commit hook that runs the test before every commit. Ensure that each developer who is working on the project has the pre-commit hook instated locally. Review the test report and resolve any issues before pushing changes to AWS CodeCommit.

B. Add a new stage to the pipeline. Use AWS CodeBuild as the provider. Add the new stage after the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage if any test does not pass. Use the test reports feature of Codebuild to integrate the report with the CodoBuild console. View the test results in CodeBuild Resolve any issues.

C. Add a new stage to the pipeline. Use AWS CodeBuild at the provider. Add the new stage before the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage it any test does not pass. Use the test reports feature of CodeBuild to integrate the report with the CodeBuild console. View the test results in codeBuild Resolve any issues.

D. Add a new stage to the pipeline. Use Jenkins as the provider. Configure CodePipeline to use Jenkins to run the unit tests. Write a

Jenkinsfile that fails the stage if any test does not pass. Use the test report plugin for Jenkins to integrate the repot with the Jenkins dashboard. View the test results in Jenkins. Resolve any issues.

_____ Answer: C

Explanation:

The solution that will meet the requirements is to add a new stage to the pipeline. Use AWS CodeBuild as the provider. Add the new stage before the stage that deploys code revisions to the test environment. Write a buildspec that fails the CodeBuild stage if any test does not pass. Use the test reports feature of CodeBuild to integrate the report with the CodeBuild console. View the test results in CodeBuild. Resolve any issues. This way, the developer can run the unit tests automatically during the CI/CD process and catch any bugs before deploying to the test environment. The developer can also use the test reports feature of CodeBuild to view and analyze the test results in a graphical interface. The other options either involve running the tests manually, running them after deployment, or using a different provider that requires additional configuration and integration.

Reference: Test reports for CodeBuild

Question: 79

A company has multiple Amazon VPC endpoints in the same VPC. A developer needs configure an Amazon S3 bucket policy so users can access an S3 bucket only by using these VPC endpoints.

Which solution will meet these requirements?

A. Create multiple S3 bucket polices by using each VPC endpoint ID that have the aws SourceVpce value in the StringNotEquals condition.

B. Create a single S3 bucket policy that has the aws SourceVpc value and in the StingNotEquals condition to use VPC ID.

C. Create a single S3 bucket policy that the multiple aws SourceVpce value and in the SringNotEquals condton to use vpce.

D. Create a single S3 bucket policy that has multiple aws sourceVpce value in the StingNotEquale condition. Repeat for all the VPC endpoint IDs.

_____ Answer: D

Explanation:

This solution will meet the requirements by creating a single S3 bucket policy that denies access to the S3 bucket unless the request comes from one of the specified VPC endpoints. The aws:SourceVpce condition key is used to match the ID of the VPC endpoint that is used to access the S3 bucket. The StringNotEquals condition operator is used to negate the condition, so that only requests from the listed VPC endpoints are allowed. Option A is not optimal because it will create multiple S3 bucket policies, which is not possible as only one bucket policy can be attached to an S3 bucket. Option B is not optimal because it will use the aws:SourceVpc condition key, which matches the ID of the VPC that is used to access the S3 bucket, not the VPC endpoint. Option C is not optimal because it will use the StringNotEquals condition operator with a single value, which will deny access to the S3 bucket from all VPC endpoints except one.

Reference: Using Amazon S3 Bucket Policies and User Policies, AWS Global Condition Context Keys

Question: 80

A company uses a custom root certificate authority certificate chain (Root CA Cert) that is 10 KB in size generate SSL certificates for its on-premises HTTPS endpoints. One of the company's cloud based applications has hundreds of AWS Lambda functions that pull date from these endpoints. A developer updated the trust store of the Lambda execution environment to use the Root CA Cert when the Lambda execution environment is initialized. The developer bundled the Root CA Cert as a text file in the Lambdas deployment bundle.

After 3 months of development the root CA Cert is no longer valid and must be updated. The developer needs a more efficient solution to update the Root CA Cert for all deployed Lambda functions. The solution must not include rebuilding or updating all Lambda functions that use the Root CA Cert. The solution must also work for all development, testing and production environment. Each environment is managed in a separate AWS account.

When combination of steps Would the developer take to meet these environments MOST cost- effectively? (Select TWO)

A. Store the Root CA Cert as a secret in AWS Secrets Manager. Create a resource-based policy. Add IAM users to allow access to the secret

B. Store the Root CA Cert as a Secure Sting parameter in aws Systems Manager Parameter Store Create a resource-based policy. Add IAM users to allow access to the policy.

C. Store the Root CA Cert in an Amazon S3 bucket. Create a resource- based policy to allow access to the bucket.

D. Refactor the Lambda code to load the Root CA Cert from the Root CA Certs location. Modify the runtime trust store inside the Lambda function handler.

E. Refactor the Lambda code to load the Root CA Cert from the Root CA Cert's location. Modify the runtime trust store outside the Lambda function handler.

Answer: BE

Explanation:

This solution will meet the requirements by storing the Root CA Cert as a Secure String parameter in AWS Systems Manager Parameter Store, which is a secure and scalable service for storing and managing configuration data and secrets. The resource-based policy will allow IAM users in different AWS accounts and environments to access the parameter without requiring cross-account roles or permissions. The Lambda code will be refactored to load the Root CA Cert from the parameter store and modify the runtime trust store outside the Lambda function handler, which will improve performance and reduce latency by avoiding repeated calls to Parameter Store and trust store modifications for each invocation of the Lambda function. Option A is not optimal because it will use AWS Secrets Manager instead of AWS Systems Manager Parameter Store, which will incur additional costs and complexity for storing and managing a non-secret configuration data such as Root CA Cert. Option C is not optimal because it will deactivate the application secrets and monitor the application error logs temporarily, which will cause application downtime and potential data loss. Option D is not optimal because it will modify the runtime trust store inside the Lambda function handler, which will degrade performance and increase latency by repeating unnecessary operations for each invocation of the Lambda function.

Reference: AWS Systems Manager Parameter Store, [Using SSL/TLS to Encrypt a Connection to a DB Instance]

Question: 81

A developer maintains applications that store several secrets in AWS Secrets Manager. The applications use secrets that have changed over time. The developer needs to identify required secrets that are still in use. The developer does not want to cause any application downtime.

What should the developer do to meet these requirements?

A. Configure an AWS CloudTrail log file delivery to an Amazon S3 bucket. Create an Amazon CloudWatch alarm for the GetSecretValue. Secrets Manager API operation requests

B. Create a secrets manager-secret-unused AWS Config managed rule. Create an Amazon EventBridge rule to Initiate notification when the AWS Config managed rule is met.

C. Deactivate the applications secrets and monitor the applications error logs temporarily.

D. Configure AWS X-Ray for the applications. Create a sampling rule lo match the GetSecretValue Secrets Manager API operation requests.

Answer: B

Explanation:

This solution will meet the requirements by using AWS Config to monitor and evaluate whether Secrets Manager secrets are unused or have been deleted, based on specified time periods. The secrets manager-secret-unused managed rule is a predefined rule that checks whether Secrets Manager secrets have been rotated within a specified number of days or have been deleted within a specified number of days after last accessed date. The Amazon EventBridge rule will trigger a notification when the AWS Config managed rule is met, alerting the developer about unused secrets that can be removed without causing application downtime. Option A is not optimal because it will use AWS CloudTrail log file delivery to an Amazon S3 bucket, which will incur additional costs and complexity for storing and analyzing log files that may not contain relevant information about secret usage. Option C is not optimal because it will deactivate the application secrets and monitor the application error logs temporarily, which will cause application downtime and potential data loss. Option D is not optimal because it will use AWS X-Ray to trace secret usage, which will introduce additional overhead and latency for instrumenting and sampling requests that may not be related to secret usage.

Reference: [AWS Config Managed Rules], [Amazon EventBridge]

Question: 82

A developer is writing a serverless application that requires an AWS Lambda function to be invoked every 10 minutes.

What is an automated and serverless way to invoke the function?

A. Deploy an Amazon EC2 instance based on Linux, and edit its /etc/confab file by adding a command to periodically invoke the lambda function

B. Configure an environment variable named PERIOD for the Lambda function. Set the value to 600.

C. Create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function.

D. Create an Amazon Simple Notification Service (Amazon SNS) topic that has a subscription to the Lambda function with a 600-second timer.

Answer: C

Explanation:

The solution that will meet the requirements is to create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function. This way, the developer can use an automated and serverless way to invoke the function every 10 minutes. The developer can also use a cron expression or a rate expression to specify the schedule for the rule. The other options either involve using an Amazon EC2 instance, which is not serverless, or using environment variables or query parameters, which do not trigger the function.

Reference: Schedule AWS Lambda functions using EventBridge

Question: 83

Users are reporting errors in an application. The application consists of several micro services that are deployed on Amazon Elastic Container Serves (Amazon ECS) with AWS Fargate.

When combination of steps should a developer take to fix the errors? (Select TWO)

A. Deploy AWS X-Ray as a sidecar container to the micro services. Update the task role policy to allow access to me X -Ray API.

B. Deploy AWS X-Ray as a daemon set to the Fargate cluster. Update the service role policy to allow access to the X-Ray API.

C. Instrument the application by using the AWS X-Ray SDK. Update the application to use the Put- XrayTrace API call to communicate with the X-Ray API.

D. Instrument the application by using the AWS X-Ray SDK. Update the application to communicate with the X-Ray daemon.

E. Instrument the ECS task to send the stout and spider- output to Amazon CloudWatch Logs. Update the task role policy to allow the cloudwatch Putlogs action.

Answer: A, E

Explanation:

The combination of steps that the developer should take to fix the errors is to deploy AWS X-Ray as a sidecar container to the microservices and instrument the ECS task to send the stdout and stderr output to Amazon CloudWatch Logs. This way, the developer can use AWS X-Ray to analyze and debug the performance of the microservices and identify any issues or bottlenecks. The developer can also use CloudWatch Logs to monitor and troubleshoot the logs from the ECS task and detect any errors or exceptions. The other options

either involve using AWS X-Ray as a daemon set, which is not supported by Fargate, or using the PutTraceSegments API call, which is not necessary when using a sidecar container.

Reference: Using AWS X-Ray with Amazon ECS

Question: 84

A company is using Amazon OpenSearch Service to implement an audit monitoring system. A developer needs to create an AWS Cloudformation custom resource that is associated with an AWS Lambda function to configure the OpenSearch Service domain. The Lambda function must access the OpenSearch Service domain by using Open Search Service internal master user credentials.

What is the MOST secure way to pass these credentials to the Lambdas function?

A. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable. Set the No Echo attenuate to true.

B. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and to create a parameter. In AWS Systems Manager Parameter Store. Set the No Echo attribute to true. Create an 1AM role that has the ssm GetParameter permission. Assign me role to the Lambda function. Store me parameter name as the Lambda function's environment variable. Resolve the parameter's value at runtime.

C. Use a CloudFormation parameter to pass the master uses credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment varleWe Encrypt the parameters value by using the AWS Key Management Service (AWS KMS) encrypt command.

D. Use CloudFoimalion to create an AWS Secrets Manager Secret. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions. Create an 1AM role that has the secrets manager. GetSecretvalue permission. Assign the role to the Lambda Function Store the secrets name as the Lambda function's environment variable. Resole the secret's value at runtime.

Answer: D

Explanation:

The solution that will meet the requirements is to use CloudFormation to create an AWS Secrets Manager secret. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions. Create an IAM role that has the secretsmanager:GetSecretValue permission. Assign the role to the Lambda function. Store the secret's name as the Lambda function's environment variable. Resolve the secret's value at runtime. This way, the developer can pass the credentials to the Lambda function in a secure way, as AWS Secrets Manager encrypts and manages the secrets. The developer can also use a dynamic reference to avoid exposing the secret's value in plain text in the CloudFormation template. The other options either involve passing the credentials as plain text parameters, which is not secure, or encrypting them with AWS KMS, which is less convenient than using AWS Secrets Manager.

Reference: Using dynamic references to specify template values

Question: 85

An application runs on multiple EC2 instances behind an ELB.

Where is the session data best written so that it can be served reliably across multiple requests?

A. Write data to Amazon ElastiCache

B. Write data to Amazon Elastic Block Store

C. Write data to Amazon EC2 instance Store

D. Wide data to the root filesystem

Answer: A

Explanation:

The solution that will meet the requirements is to write data to Amazon ElastiCache. This way, the application can write session data to a fast, scalable, and reliable in-memory data store that can be served reliably across multiple requests. The other options either involve writing data to persistent storage, which is slower and more expensive than in-memory storage, or writing data to the root filesystem, which is not shared among multiple EC2 instances.

Reference: Using ElastiCache for session management

Question: 86

An ecommerce application is running behind an Application Load Balancer. A developer observes some unexpected load on the application during non-peak hours. The developer wants to analyze patterns for the client IP addresses that use the application. Which HTTP header should the developer use for this analysis?

A. The X-Forwarded-Proto header

B. The X-F Forwarded-Host header

C. The X-Forwarded-For header

D. The X-Forwarded-Port header

Answer: C

Explanation:

The HTTP header that the developer should use for this analysis is the X-Forwarded-For header. This header contains the IP address of the client that made the request to the Application Load Balancer. The developer can use this header to analyze patterns for the client IP addresses that use the application. The other headers either contain information about the protocol, host, or port of the request, which are not relevant for the analysis.

Reference: How Application Load Balancer works with your applications

Question: 87

A developer migrated a legacy application to an AWS Lambda function. The function uses a third- party service to pull data with a series of API calls at the end of each month. The function than processes the data to generate the monthly reports. The function has Been working with no issues so far.

The third-party service recently issued a restriction to allow a feed number to API calls each minute and each day. If the API calls exceed the limit tor each minute or each day, then the service will produce errors. The API also provides the minute limit and daily limit in the response header. This restriction might extend the overall process to multiple days because the process is consuming more API calls than the available limit.

What is the MOST operationally efficient way to refactor the server less application to accommodate this change?

A. Use an AWS Step Functions State machine to monitor API failures. Use the Wait state to delay calling the Lambda function.

B. Use an Amazon Simple Queue Service (Amazon SQS) queue to hold the API calls. Configure the Lambda function to poll the queue within the API threshold limits.

C. Use an Amazon CloudWatch Logs metric to count the number of API calls. Configure an Amazon CloudWatch alarm flat slops the currently running instance of the Lambda function when the metric exceeds the API threshold limits.

D. Use Amazon Kinesis Data Firehose to batch me API calls and deliver them to an Amazon S3 bucket win an event notification to invoke the Lambda function.

Answer: A

Explanation:

The solution that will meet the requirements is to use an AWS Step Functions state machine to monitor API failures. Use the Wait state to delay calling the Lambda function. This way, the developer can refactor the serverless application to accommodate the change in a way that is automated and scalable. The developer can use Step Functions to orchestrate the Lambda function and handle any errors or retries. The developer can also use the Wait state to pause the execution for a specified duration or until a specified timestamp, which can help avoid exceeding the API limits. The other options either involve using additional services that are not necessary or appropriate for this scenario, or do not address the issue of API failures.

Reference: AWS Step Functions Wait state

Question: 88

A developer must analyze performance issues with production-distributed applications written as AWS Lambda functions. These distributed Lambda applications invoke other components that make up me applications. How should the developer identify and troubleshoot the root cause of the performance issues in production?

A. Add logging statements to the Lambda functions. then use Amazon CloudWatch to view the logs.

B. Use AWS CloudTrail and then examine the logs.

C. Use AWS X-Ray. then examine the segments and errors.

D. Run Amazon inspector agents and then analyze performance.

Answer: C

Explanation:

This solution will meet the requirements by using AWS X-Ray to analyze and debug the performance issues with the distributed Lambda applications. AWS X-Ray is a service that collects data about requests that the applications serve, and provides tools to view, filter, and gain insights into that data. The developer can use AWS X-Ray to identify the root cause of the performance issues by examining the segments and errors that show the details of each request and the components that make up the applications. Option A is not optimal because it will use logging statements and Amazon CloudWatch, which may not provide enough information or visibility into the distributed applications. Option B is not optimal because it will use AWS CloudTrail, which is a service that records API calls and events for AWS services, not application performance data. Option D is not optimal because it will use Amazon Inspector, which is a service that helps improve the security and compliance of applications on Amazon EC2 instances, not Lambda functions.

Reference: AWS X-Ray, Using AWS X-Ray with AWS Lambda

Question: 89

A developer wants to deploy a new version of an AWS Elastic Beanstalk application. During deployment the application must maintain full capacity and avoid service interruption. Additionally, the developer must minimize the cost of additional resources that support the deployment.

Which deployment method should the developer use to meet these requirements?

A. All at once

B. Rolling with additional batch

C. Bluegreen

D. Immutable

Answer: B

Explanation:

This solution will meet the requirements by using a rolling with additional batch deployment method, which deploys the new version of the application to a separate group of instances and then shifts traffic to those instances in batches. This way, the application maintains full capacity and avoids service interruption during deployment, as well as minimizes the cost of additional resources that support the deployment. Option A is not optimal because it will use an all at once deployment method, which deploys the new version of the application to all instances simultaneously, which may cause service interruption or downtime during deployment. Option C is not optimal because it will use a blue/green deployment method, which deploys the new version of the application to a separate environment and then swaps URLs with the original environment, which may incur more costs for additional resources that support the deployment. Option D is not optimal because it will use an immutable deployment method, which deploys the new version of the application to a fresh group of instances and then redirects traffic to those instances, which may also incur more costs for additional resources that support the deployment.

Reference: AWS Elastic Beanstalk Deployment Policies

Question: 90

A developer has observed an increase in bugs in the AWS Lambda functions that a development team has deployed in its Node is application. To minimize these bugs, the developer wants to impendent automated testing of Lambda functions in an environment that Closely simulates the Lambda environment.

The developer needs to give other developers the ability to run the tests locally. The developer also needs to integrate the tests into the team's continuous integration and continuous delivery (Ct/CO) pipeline before the AWS Cloud Development Kit (AWS COK) deployment.

Which solution will meet these requirements?

A. Create sample events based on the Lambda documentation. Create automated test scripts that use the cdk local invoke command to invoke the Lambda functions. Check the response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.

B. Install a unit testing framework that reproduces the Lambda execution environment. Create sample events based on the Lambda Documentation Invoke the handler function by using a unit testing framework. Check the response Document how to run the unit testing framework for the other developers on the team. Update the OCD pipeline to run the unit testing framework.

C. Install the AWS Serverless Application Model (AWS SAW) CLI tool Use the Sam local generateevent command to generate sample events for me automated tests. Create automated test scripts that use the Sam local invoke command to invoke the Lambda functions. Check the response

Document the test scripts tor the other developers on the team Update the CI/CD pipeline to run the test scripts.

D. Create sample events based on the Lambda documentation. Create a Docker container from the Node is base image to invoke the Lambda functions. Check the response Document how to run the Docker container for the more developers on the team update the CI/CD pipeline to run the Docker container.

Answer: C

Explanation:

This solution will meet the requirements by using AWS SAM CLI tool, which is a command line tool that lets developers locally build, test, debug, and deploy serverless applications defined by AWS SAM templates. The developer can use sam local generate-event command to generate sample events for different event sources such as API Gateway or S3. The developer can create automated test scripts that use sam local invoke command to invoke Lambda functions locally in an environment that closely simulates Lambda environment. The developer can check the response from Lambda functions and document how to run the test scripts for other developers on the team. The developer can also update CI/CD pipeline to run these test scripts before deploying with AWS CDK. Option A is not optimal because it will use cdk local invoke command, which does not exist in AWS CDK CLI tool. Option B is not optimal because it will use a unit testing framework that reproduces Lambda execution environment, which may not be accurate or consistent with Lambda environment. Option D is not optimal because it will create a Docker container from Node.js base image to invoke Lambda functions, which may introduce additional overhead and complexity for creating and running Docker containers.

Reference: [AWS Serverless Application Model (AWS SAM)], [AWS Cloud Development Kit (AWS CDK)]

Question: 91

A developer is troubleshooting an application mat uses Amazon DynamoDB in the uswest-2 Region. The application is deployed to an Amazon EC2 instance. The application requires read-only permissions to a table that is named Cars The EC2 instance has an attached IAM role that contains the following IAM policy.

```
•Statensnt": |

        "Sic": "RpaiOn-yAJ 'Actions',
        "Effect*: *Allcw[H],
         Act Lon : i
              "dysaaddt: j=tltesi".
                        : «stch£«l • Lo?i ,
              *dynainodb:Scan', •dyrjunjdb:ou»ry',
              •dynwsrfb:condxtioaCh»iklt«n"
         4 *
```

When the application tries to read from the Cars table, an Access Denied error occurs.

How can the developer resolve this error?

A. Modify the IAM policy resource to be "arn aws dynamo* us-west-2 account-id table/*"

B. Modify the IAM policy to include the dynamodb * action

C. Create a trust policy that specifies the EC2 service principal. Associate the role with the policy.

D. Create a trust relationship between the role and dynamodb Amazonas com.

Answer: C

Explanation:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/access-control-overview.html#access-control-resource-ownership

Question: 92

A developer needs to store configuration variables for an application. The developer needs to set an expiration date and time for me configuration. The developer wants to receive notifications. Before the configuration expires. Which solution will meet these requirements with the LEAST operational overhead?

A. Create a standard parameter in AWS Systems Manager Parameter Store Set Expiation and Expiration Notification policy types.

B. Create a standard parameter in AWS Systems Manager Parameter Store Create an AWS Lambda function to expire the configuration and to send Amazon Simple Notification Service (Amazon SNS) notifications.

C. Create an advanced parameter in AWS Systems Manager Parameter Store Set Expiration and Expiration Notification policy

types.

D. Create an advanced parameter in AWS Systems Manager Parameter Store Create an Amazon EC2 instance with a corn job to expire the configuration and to send notifications.

_____ Answer: C

Explanation:

This solution will meet the requirements by creating an advanced parameter in AWS Systems Manager Parameter Store, which is a secure and scalable service for storing and managing configuration data and secrets. The advanced parameter allows setting expiration and expiration notification policy types, which enable specifying an expiration date and time for the configuration and receiving notifications before the configuration expires. The Lambda code will be refactored to load the Root CA Cert from the parameter store and modify the runtime trust store outside the Lambda function handler, which will improve performance and reduce latency by avoiding repeated calls to Parameter Store and trust store modifications for each invocation of the Lambda function. Option A is not optimal because it will create a standard parameter in AWS Systems Manager Parameter Store, which does not support expiration and expiration notification policy types. Option B is not optimal because it will create a secret access key and access key ID with permission to access the S3 bucket, which will introduce additional security risks and complexity for storing and managing credentials. Option D is not optimal because it will create a Docker container from Node.js base image to invoke Lambda functions, which will incur additional costs and overhead for creating and running Docker containers.

Reference: AWS Systems Manager Parameter Store, [Using SSL/TLS to Encrypt a Connection to a DB Instance]

Question: 93 _____

When using the AWS Encryption SDK how does the developer keep track of the data encryption keys used to encrypt data?

A. The developer must manually keep Hack of the data encryption keys used for each data object.

B. The SDK encrypts the data encryption key and stores it (encrypted) as part of the resumed ophertext.

C. The SDK stores the data encryption keys automaticity in Amazon S3.

D. The data encryption key is stored m the user data for the EC2 instance.

_____ Answer: B

Explanation:

This solution will meet the requirements by using AWS Encryption SDK, which is a client-side encryption library that enables developers to encrypt and decrypt data using data encryption keys that are protected by AWS Key Management Service (AWS KMS). The SDK encrypts the data encryption key with a customer master key (CMK) that is managed by AWS KMS, and stores it (encrypted) as part of the returned ciphertext. The developer does not need to keep track of the data encryption keys used to encrypt data, as they are stored with the encrypted data and can be retrieved and decrypted by using AWS KMS when needed. Option A is not optimal because it will require manual tracking of the data encryption keys used for each data object, which is error-prone and inefficient. Option C is not optimal because it will store the data encryption keys automatically in Amazon S3, which is unnecessary and insecure as Amazon S3 is not designed for storing encryption keys. Option D is not optimal because it will store the data encryption key in the user data for the EC2 instance, which is also unnecessary and insecure as user data is not encrypted by default.

Reference: [AWS Encryption SDK], [AWS Key Management Service]

Question: 94

An application that runs on AWS Lambda requires access to specific highly confidential objects in an Amazon S3 bucket. In accordance with the principle of least privilege a company grants access to the S3 bucket by using only temporary credentials.

How can a developer configure access to the S3 bucket in the MOST secure way?

A. Hardcode the credentials that are required to access the S3 objects in the application code. Use the credentials to access me required S3 objects.

B. Create a secret access key and access key ID with permission to access the S3 bucket. Store the key and key ID in AWS Secrets Manager. Configure the application to retrieve the Secrets Manager secret and use the credentials to access me S3 objects.

C. Create a Lambda function execution role Attach a policy to the rote that grants access to specific objects in the S3 bucket.

D. Create a secret access key and access key ID with permission to access the S3 bucket Store the key and key ID as environment variables m Lambda. Use the environment variables to access the required S3 objects.

Answer: C

Explanation:

This solution will meet the requirements by creating a Lambda function execution role, which is an IAM role that grants permissions to a Lambda function to access AWS resources such as Amazon S3 objects. The developer can attach a policy to the role that grants access to specific objects in the S3 bucket that are required by the application, following the principle of least privilege. Option A is not optimal because it will hardcode the credentials that are required to access S3 objects in the application code, which is insecure and difficult to maintain. Option B is not optimal because it will create a secret access key and access key ID with permission to access the S3 bucket, which will introduce additional security risks and complexity for storing and managing credentials. Option D is not optimal because it will store the secret access key and access key ID as environment variables in Lambda, which is also insecure and difficult to maintain.

Reference: [AWS Lambda Execution Role], [Using AWS Lambda with Amazon S3]

Question: 95

A developer has code that is stored in an Amazon S3 bucket. The code must be deployed as an AWS Lambda function across multiple accounts in the same AWS Region as the S3 bucket an AWS CloudPormation template that runs for each account will deploy the Lambda function.

What is the MOST secure way to allow CloudFormaton to access the Lambda Code in the S3 bucket?

A. Grant the CloudFormation service role the S3 ListBucket and GetObject permissions. Add a bucket policy to Amazon S3 with the principal of "AWS" (account numbers)

B. Grant the CloudFormation service row the S3 GetObfect permission. Add a Bucket policy to Amazon S3 with the principal of """

C. Use a service-based link to grant the Lambda function the S3 ListBucket and GetObject permissions by explicitly adding the S3 bucket's account number in the resource.

D. Use a service-based link to grant the Lambda function the S3 GetObject permission Add a resource of "** to allow access to the S3 bucket.

Answer: B

Explanation:

This solution allows the CloudFormation service role to access the S3 bucket from any account, as long as it has the S3 GetObject permission. The bucket policy grants access to any principal with the GetObject permission, which is the least privilege needed to deploy the Lambda code. This is more secure than granting ListBucket permission, which is not required for deploying Lambda code, or using a service-based link, which is not supported for Lambda functions.

Reference: AWS CloudFormation Service Role, Using AWS Lambda with Amazon S3

Question: 96

A developer warns to add request validation to a production environment Amazon API Gateway API. The developer needs to test the changes before the API is deployed to the production environment. For the lest the developer will send test requests to the API through a testing tool.

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

A. Export the existing API to an OpenAPI file. Create a new API Import the OpenAPI file Modify the new API to add request validation. Perform the tests Modify the existing API to add request validation. Deploy the existing API to production.

B. Modify the existing API to add request validation. Deploy the updated API to a new API Gateway stage Perform the tests Deploy the updated API to the API Gateway production stage.

C. Create a new API Add the necessary resources and methods including new request validation. Perform the tests Modify the existing API to add request validation. Deploy the existing API to production.

D. Clone the exiting API Modify the new API lo add request validation. Perform the tests Modify the existing API to add request validation Deploy the existing API to production.

Answer: D

Explanation:

This solution allows the developer to test the changes without affecting the production environment. Cloning an API creates a copy of the API definition that can be modified independently. The developer can then add request validation to the new API and test it using a testing tool. After verifying that the changes work as expected, the developer can apply the same changes to the existing API and deploy it to production.

Reference: Clone an API, [Enable Request Validation for an API in API Gateway]

Question: 97

A developer at a company needs to create a small application mat makes the same API call once each flay at a designated time. The company does not have infrastructure in the AWS Cloud yet, but the company wants to implement this functionality on AWS.

Which solution meets these requirements in the MOST operationally efficient manner?

A. Use a Kubermetes cron job that runs on Amazon Elastic Kubemetes Sen/ice (Amazon EKS)

B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2

C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

D. Use an AWS Batch job that is submitted to an AWS Batch job queue.

Answer: C

Explanation:

This solution meets the requirements in the most operationally efficient manner because it does not require any infrastructure provisioning or management. The developer can create a Lambda function that makes the API call and configure an EventBridge rule that triggers the function once a day at a designated time. This is a serverless solution that scales automatically and only charges for the execution time of the function.

Reference: [Using AWS Lambda with Amazon EventBridge], [Schedule Expressions for Rules]

Question: 98

A developer is building a serverless application that is based on AWS Lambda. The developer initializes the AWS software development kit (SDK) outside of the Lambda handcar function.

What is the PRIMARY benefit of this action?

A. Improves legibility and systolic convention

B. Takes advantage of runtime environment reuse

C. Provides better error handling

D. Creates a new SDK instance for each invocation

Answer: B

Explanation:

This benefit occurs when initializing the AWS SDK outside of the Lambda handler function because it allows the SDK instance to be reused across multiple invocations of the same function. This can improve performance and reduce latency by avoiding unnecessary initialization overhead. If the SDK is initialized inside the handler function, it will create a new SDK instance for each invocation, which can increase

memory usage and execution time.

Reference: [AWS Lambda execution environment], [Best Practices for Working with AWS Lambda Functions]

Question: 99

A company is using Amazon RDS as the Backend database for its application. After a recent marketing campaign, a surge of read requests to the database increased the latency of data retrieval from the database.

The company has decided to implement a caching layer in front of the database. The cached content must be encrypted and must be highly available.

Which solution will meet these requirements?

A. Amazon Cloudfront

B. Amazon ElastiCache to Memcached

C. Amazon ElastiCache for Redis in cluster mode

D. Amazon DynamoDB Accelerate (DAX)

Answer: C

Explanation:

This solution meets the requirements because it provides a caching layer that can store and retrieve encrypted data from multiple nodes. Amazon ElastiCache for Redis supports encryption at rest and in transit, and can scale horizontally to increase the cache capacity and availability. Amazon ElastiCache for Memcached does not support encryption, Amazon CloudFront is a content delivery network that is not suitable for caching database queries, and Amazon DynamoDB Accelerator (DAX) is a caching service that only works with DynamoDB tables.

Reference: [Amazon ElastiCache for Redis Features], [Choosing a Cluster Engine]

Question: 100

A developer at a company recently created a serverless application to process and show data from business reports. The application's user interface (UI) allows users to select and start processing the files. The Ul displays a message when the result is available to view. The application uses AWS Step Functions with AWS Lambda functions to process the files. The developer used Amazon API Gateway and Lambda functions to create an API to support the UI.

The company's Ul team reports that the request to process a file is often returning timeout errors because of the see or complexity of the files. The Ul team wants the API to provide an immediate response so that the Ul can deploy a message while the files are being processed. The backend process that is invoked by the API needs to send an email message when the report processing is complete.

What should the developer do to configure the API to meet these requirements?

A. Change the API Gateway route to add an X-Amz-Invocation-Type header win a sialic value of 'Event' in the integration request Deploy

the API Gateway stage to apply the changes.

B. Change the configuration of the Lambda function that implements the request to process a file. Configure the maximum age of the event so that the Lambda function will ion asynchronously.

C. Change the API Gateway timeout value to match the Lambda function ominous value. Deploy the API Gateway stage to apply the changes.

D. Change the API Gateway route to add an X-Amz-Target header with a static value of 'A sync' in the integration request Deploy me API Gateway stage to apply the changes.

Answer: A

Explanation:

This solution allows the API to invoke the Lambda function asynchronously, which means that the API will return an immediate response without waiting for the function to complete. The X-Amz- Invocation-Type header specifies the invocation type of the Lambda function, and setting it to 'Event' means that the function will be invoked asynchronously. The function can then use Amazon Simple Email Service (SES) to send an email message when the report processing is complete.

Reference: [Asynchronous invocation], [Set up Lambda proxy integrations in API Gateway]

Question: 101

A developer has an application that is composed of many different AWS Lambda functions. The Lambda functions all use some of the same dependencies. To avoid security issues the developer is constantly updating the dependencies of all of the Lambda functions. The result is duplicated effort to reach function.

How can the developer keep the dependencies of the Lambda functions up to date with the LEAST additional complexity?

A. Define a maintenance window for the Lambda functions to ensure that the functions get updated copies of the dependencies.

B. Upgrade the Lambda functions to the most recent runtime version.

C. Define a Lambda layer that contains all of the shared dependencies.

D. Use an AWS CodeCommit repository to host the dependencies in a centralized location.

Answer: C

Explanation:

This solution allows the developer to keep the dependencies of the Lambda functions up to date with the least additional complexity because it eliminates the need to update each function individually. A Lambda layer is a ZIP archive that contains libraries, custom runtimes, or other dependencies. The developer can create a layer that contains all of the shared dependencies and attach it to multiple Lambda functions. When the developer updates the layer, all of the functions that use the layer will have access to the latest version of the dependencies.

Reference: [AWS Lambda layers]

Question: 102

A mobile app stores blog posts in an Amazon DynacnoDB table Millions of posts are added every day and each post represents a single item in the table. The mobile app requires only recent posts. Any post that is older than 48 hours can be removed.

What is the MOST cost-effective way to delete posts that are older man 48 hours?

A. For each item add a new attribute of type String that has a timestamp that is set to the blog post creation time. Create a script to find old posts with a table scan and remove posts that are order than 48 hours by using the Balch Write Item API operation. Schedule a cron job on an Amazon EC2 instance once an hour to start the script.

B. For each item add a new attribute of type. String that has a timestamp that its set to the blog post creation time. Create a script to find old posts with a table scan and remove posts that are Oder than 48 hours by using the Batch Write item API operating. Place the script in a container image. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Far gate that invokes the container every 5 minutes.

C. For each item, add a new attribute of type Date that has a timestamp that is set to 48 hours after the blog post creation time. Create a global secondary index (GSI) that uses the new attribute as a sort key. Create an AWS Lambda function that references the GSI and removes expired items by using the Batch Write item API operation Schedule me function with an Amazon CloudWatch event every minute.

D. For each item add a new attribute of type. Number that has timestamp that is set to 48 hours after the blog post. creation time Configure the DynamoDB table with a TTL that references the new attribute.

Answer: D

Explanation:

This solution will meet the requirements by using the Time to Live (TTL) feature of DynamoDB, which enables automatically deleting items from a table after a certain time period. The developer can add a new attribute of type Number that has a timestamp that is set to 48 hours after the blog post creation time, which represents the expiration time of the item. The developer can configure the DynamoDB table with a TTL that references the new attribute, which instructs DynamoDB to delete the item when the current time is greater than or equal to the expiration time. This solution is also cost-effective as it does not incur any additional charges for deleting expired items. Option A is not optimal because it will create a script to find and remove old posts with a table scan and a batch write item API operation, which may consume more read and write capacity units and incur more costs. Option B is not optimal because it will use Amazon Elastic Container Service (Amazon ECS) and AWS Fargate to run the script, which may introduce additional costs and complexity for managing and scaling containers. Option C is not optimal because it will create a global secondary index (GSI) that uses the expiration time as a sort key, which may consume more storage space and incur more costs.

Reference: Time To Live, Managing DynamoDB Time To Live (TTL)

Question: 103

A developer is modifying an existing AWS Lambda function White checking the code the developer notices hardcoded parameter various

for an Amazon RDS for SQL Server user name password database host and port. There also are hardcoded parameter values for an Amazon DynamoOB table. an Amazon S3 bucket, and an Amazon Simple Notification Service (Amazon SNS) topic.

The developer wants to securely store the parameter values outside the code m an encrypted format and wants to turn on rotation for the credentials. The developer also wants to be able to reuse the parameter values from other applications and to update the parameter values without modifying code.

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

A. Create an RDS database secret in AWS Secrets Manager. Set the user name password, database, host and port. Turn on secret rotation. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket and SNS topic.

B. Create an RDS database secret in AWS Secrets Manager. Set the user name password, database, host and port. Turn on secret rotation. Create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket and SNS topic.

C. Create RDS database parameters in AWS Systems Manager Parameter. Store for the user name password, database, host and port. Create encrypted Lambda environment variables for me DynamoDB table, S3 bucket, and SNS topic. Create a Lambda function and set the logic for the credentials rotation task Schedule the credentials rotation task in Amazon EventBridge.

D. Create RDS database parameters in AWS Systems Manager Parameter. Store for the user name password database, host, and port. Store the DynamoDB table. S3 bucket, and SNS topic in Amazon S3 Create a Lambda function and set the logic for the credentials rotation Invoke the Lambda function on a schedule.

Answer: B

Explanation:

This solution will meet the requirements by using AWS Secrets Manager and AWS Systems Manager Parameter Store to securely store the parameter values outside the code in an encrypted format. AWS Secrets Manager is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can create an RDS database secret in AWS Secrets Manager and set the user name, password, database, host, and port for accessing the RDS database. The developer can also turn on secret rotation, which will change the database credentials periodically according to a specified schedule or event. AWS Systems Manager Parameter Store is a service that provides secure and scalable storage for configuration data and secrets. The developer can create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket, and SNS topic, which will encrypt them with AWS KMS. The developer can also reuse the parameter values from other applications and update them without modifying code. Option A is not optimal because it will create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic, which may not be reusable or updatable without modifying code. Option C is not optimal because it will create RDS database parameters in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option D is not optimal because it will store the DynamoDB table, S3 bucket, and SNS topic in Amazon S3, which may introduce additional costs and complexity for accessing configuration data.

Reference: AWS Secrets Manager, [AWS Systems Manager Parameter Store]

Question: 104

A developer accesses AWS CodeCommit over SSH. The SSH keys configured to access AWS

CodeCommit are tied to a user with the following permissions:

```
{
"Version": "2012-10-17",
"Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "codecommit:BatchGetRepositories",
    "codecommit:Get*",
    "codecommit:List*",
    "codecommit:GitPull"
   ],
   "Resource": "*"
  }
]
}
```

The developer needs to create/delete branches

Which specific IAM permissions need to be added based on the principle of least privilege?

A. Option A

B. Option B

C. Option C

D. Option D

Answer: A

Explanation:

This solution allows the developer to create and delete branches in AWS CodeCommit by granting the codecommit:CreateBranch and codecommit:DeleteBranch permissions. These are the minimum permissions required for this task, following the principle of least privilege. Option B grants too many permissions, such as codecommit:Put*, which allows the developer to create, update, or delete any resource in CodeCommit. Option C grants too few permissions, such as codecommit:Update*, which does not allow the developer to create or delete branches. Option D grants all permissions, such as codecommit:*, which is not secure or recommended.

Reference: [AWS CodeCommit Permissions Reference], [Create a Branch (AWS CLI)]

Question: 105

An application that is deployed to Amazon EC2 is using Amazon DynamoDB. The app cation calls the DynamoDB REST API

Periodically the application receives a

ProvisionedThroughputExceededException error when the application writes to a DynamoDB table.

Which solutions will mitigate this error MOST cost-effectively[A] (Select TWO)

A. Modify the application code to perform exponential back off when the error is received.

B. Modify the application to use the AWS SDKs for DynamoDB.

C. Increase the read and write throughput of the DynamoDB table.

D. Create a DynamoDB Accelerator (DAX) cluster for the DynamoDB table.

E. Create a second DynamoDB table Distribute the reads and writes between the two tables.

Answer: A, B

Explanation:

These solutions will mitigate the error most cost-effectively because they do not require increasing the provisioned throughput of the DynamoDB table or creating additional resources. Exponential backoff is a retry strategy that increases the waiting time between retries to reduce the number of requests sent to DynamoDB. The AWS SDKs for DynamoDB implement exponential backoff by default and also provide other features such as automatic pagination and encryption. Increasing the read and write throughput of the DynamoDB table, creating a DynamoDB Accelerator (DAX) cluster, or creating a second DynamoDB table will incur additional costs and complexity.

Reference: [Error Retries and Exponential Backoff in AWS], [Using the AWS SDKs with DynamoDB]

Question: 106

When a developer tries to run an AWS Code Build project, it raises an error because the length of all environment variables exceeds the limit for the combined maximum of characters.

What is the recommended solution?

A. Add the export LC-_ALL" on _ US, tuft" command to the pre _ build section to ensure POSIX Localization.

B. Use Amazon Cognate to store key-value pairs for large numbers of environment variables

C. Update the settings for the build project to use an Amazon S3 bucket for large numbers of environment variables

D. Use AWS Systems Manager Parameter Store to store large numbers ot environment variables

Answer: D

Explanation:

This solution allows the developer to overcome the limit for the combined maximum of characters for environment variables in AWS CodeBuild. AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. The developer can store large numbers of environment variables as parameters in Parameter Store and reference them in the buildspec file using parameter references. Adding export LC_ALL="en_US.utf8" command to the pre_build section will not affect the environment variables limit. Using Amazon Cognito or an Amazon S3 bucket to store key-value pairs for environment variables will require additional configuration and integration.

Reference: [Build Specification Reference for AWS CodeBuild], [What Is AWS Systems Manager Parameter Store?]

Question: 107

A company is expanding the compatibility of its photo-snaring mobile app to hundreds of additional devices with unique screen dimensions and resolutions. Photos are stored in Amazon S3 in their original format and resolution. The company uses an Amazon CloudFront distribution to serve the photos The app includes the dimension and resolution of the display as GET parameters with every request.

A developer needs to implement a solution that optimizes the photos that are served to each device to reduce load time and increase photo quality.

Which solution will meet these requirements MOST cost-effective?

A. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolutions. Create a dynamic CloudFront origin that automatically maps the request of each device to the corresponding photo variant.

B. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolutions. Create a Lambda@Edge function to route requests to the corresponding photo vacant by using request headers.

C. Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response. Change the CloudFront TTL cache policy to the maximum value possible.

D. Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response. In the same function store a copy of the processed photos on Amazon S3 for subsequent requests.

Answer: D

Explanation:

This solution meets the requirements most cost-effectively because it optimizes the photos on demand and caches them for future requests. Lambda@Edge allows the developer to run Lambda functions at AWS locations closer to viewers, which can reduce latency and improve photo quality. The developer can create a Lambda@Edge function that uses the GET parameters from each request to optimize the photos with the required dimensions and resolutions and returns them as a response. The function can also store a copy of the processed photos on Amazon S3 for subsequent requests, which can reduce processing time and costs. Using S3 Batch Operations to create new variants of the photos will incur additional storage costs and may not cover all possible dimensions and resolutions. Creating a dynamic CloudFront origin or a Lambda@Edge function to route requests to corresponding photo variants will require maintaining a mapping of device types and photo variants, which can be complex and error-prone.

Reference: [Lambda@Edge Overview], [Resizing Images with Amazon CloudFront & Lambda@Edge]

Question: 108

A company is building an application for stock trading. The application needs sub-millisecond latency for processing trade requests. The company uses Amazon DynamoDB to store all the trading data that is used to process each trading request A development team performs load testing on the application and finds that the data retrieval time is higher than expected. The development team needs a solution that reduces the data retrieval time with the least possible effort.

Which solution meets these requirements'?

A. Add local secondary indexes (LSis) for the trading data.

B. Store the trading data m Amazon S3 and use S3 Transfer Acceleration.

C. Add retries with exponential back off for DynamoDB queries.

D. Use DynamoDB Accelerator (DAX) to cache the trading data.

Answer: D

Explanation:

This solution will meet the requirements by using DynamoDB Accelerator (DAX), which is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10 times performance improvement - from milliseconds to microseconds - even at millions of requests per second. The developer can use DAX to cache the trading data that is used to process each trading request, which will reduce the data retrieval time with the least possible effort. Option A is not optimal because it will add local secondary indexes (LSIs) for the trading data, which may not improve the performance or reduce the latency of data retrieval, as LSIs are stored on the same partition as the base table and share the same provisioned throughput. Option B is not optimal because it will store the trading data in Amazon S3 and use S3 Transfer Acceleration, which is a feature that enables fast, easy, and secure transfers of files over long distances between S3 buckets and clients, not between DynamoDB and clients. Option C is not optimal because it will add retries with exponential backoff for DynamoDB queries, which is a strategy to handle transient errors by retrying failed requests with increasing delays, not by reducing data retrieval time.

Reference: [DynamoDB Accelerator (DAX)], [Local Secondary Indexes]

Question: 109

A developer is working on a Python application that runs on Amazon EC2 instances. The developer wants to enable tracing of application requests to debug performance issues in the code.

Which combination of actions should the developer take to achieve this goal? (Select TWO)

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

B. Install the AWS X-Ray daemon on the EC2 instances.

C. Configure the application to write JSON-formatted togs to /var/log/cloudwatch.

D. Configure the application to write trace data to /Var/log-/xray.

E. Install and configure the AWS X-Ray SDK for Python in the application.

Answer: BE

Explanation:

This solution will meet the requirements by using AWS X-Ray to enable tracing of application requests to debug performance issues in the code. AWS X-Ray is a service that collects data about requests that the applications serve, and provides tools to view, filter, and gain insights into that data. The developer can install the AWS X-Ray daemon on the EC2 instances, which is a software that listens for traffic on UDP port 2000, gathers raw segment data, and relays it to the X-Ray API. The developer can also install and configure the AWS X-Ray SDK for Python in the application, which is a library that enables instrumenting Python code to generate and send trace data to the X-Ray daemon. Option A is not optimal because it will install the Amazon CloudWatch agent on the EC2 instances, which is a software that collects metrics and logs from EC2 instances and on-premises servers, not application performance data. Option C is not optimal because it will configure the application to write JSON-formatted logs to /var/log/cloudwatch, which is not a valid path or destination for CloudWatch logs. Option D is not optimal because it will configure the application to write trace data to /var/log/xray, which is also not a valid path or destination for X-Ray trace data.

Reference: [AWS X-Ray], [Running the X-Ray Daemon on Amazon EC2]

Question: 110

A company has an application that runs as a series of AWS Lambda functions. Each Lambda function receives data from an Amazon Simple Notification Service (Amazon SNS) topic and writes the data to an Amazon Aurora DB instance.

To comply with an information security policy, the company must ensure that the Lambda functions all use a single securely encrypted database connection string to access Aurora.

Which solution will meet these requirements'?

A. Use IAM database authentication for Aurora to enable secure database connections for ail the Lambda functions.

B. Store the credentials and read the credentials from an encrypted Amazon RDS DB instance.

C. Store the credentials in AWS Systems Manager Parameter Store as a secure string parameter.

D. Use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption.

Answer: A

Explanation:

This solution will meet the requirements by using IAM database authentication for Aurora, which enables using IAM roles or users to authenticate with Aurora databases instead of using passwords or other secrets. The developer can use IAM database authentication for Aurora to enable secure database connections for all the Lambda functions that access Aurora DB instance. The developer can create an IAM role with permission to connect to Aurora DB instance and attach it to each Lambda function. The developer can also configure

Aurora DB instance to use IAM database authentication and enable encryption in transit using SSL certificates. This way, the Lambda functions can use a single securely encrypted database connection string to access Aurora without needing any secrets or passwords. Option B is not optimal because it will store the credentials and read them from an encrypted Amazon RDS DB instance, which may introduce additional costs and complexity for managing and accessing another RDS DB instance. Option C is not optimal because it will store the credentials in AWS Systems Manager Parameter Store as a secure string parameter, which may require additional steps or permissions to retrieve and decrypt the credentials from Parameter Store. Option D is not optimal because it will use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption, which may not be secure or scalable as environment variables are stored as plain text unless encrypted with AWS KMS.

Reference: [IAM Database Authentication for MySQL and PostgreSQL], [Using SSL/TLS to Encrypt a Connection to a DB Instance]

Question: 111

A developer is troubleshooting an Amazon API Gateway API Clients are receiving HTTP 400 response errors when the clients try to access an endpoint of the API.

How can the developer determine the cause of these errors?

A. Create an Amazon Kinesis Data Firehose delivery stream to receive API call logs from API Gateway. Configure Amazon CloudWatch Logs as the delivery stream's destination.

B. Turn on AWS CloudTrail Insights and create a trail Specify the Amazon Resource Name (ARN) of the trail for the stage of the API.

C. Turn on AWS X-Ray for the API stage Create an Amazon CtoudWalch Logs log group Specify the Amazon Resource Name (ARN) of the log group for the API stage.

D. Turn on execution logging and access logging in Amazon CloudWatch Logs for the API stage. Create a CloudWatch Logs log group. Specify the Amazon Resource Name (ARN) of the log group for the API stage.

Answer: D

Explanation:

This solution will meet the requirements by using Amazon CloudWatch Logs to capture and analyze the logs from API Gateway. Amazon CloudWatch Logs is a service that monitors, stores, and accesses log files from AWS resources. The developer can turn on execution logging and access logging in Amazon CloudWatch Logs for the API stage, which enables logging information about API execution and client access to the API. The developer can create a CloudWatch Logs log group, which is a collection of log streams that share the same retention, monitoring, and access control settings. The developer can specify the Amazon Resource Name (ARN) of the log group for the API stage, which instructs API Gateway to send the logs to the specified log group. The developer can then examine the logs to determine the cause of the HTTP 400 response errors. Option A is not optimal because it will create an Amazon Kinesis Data Firehose delivery stream to receive API call logs from API Gateway, which may introduce additional costs and complexity for delivering and processing streaming data. Option B is not optimal because it will turn on AWS CloudTrail Insights and create a trail, which is a feature that helps identify and troubleshoot unusual API activity or operational issues, not HTTP response errors. Option C is not optimal because it will turn on AWS X-Ray for the API stage, which is a service that helps analyze and debug distributed applications, not HTTP response errors.

Reference: [Setting Up CloudWatch Logging for a REST API], [CloudWatch Logs Concepts]

Question: 112

A company developed an API application on AWS by using Amazon CloudFront. Amazon API Gateway, and AWS Lambda. The API has a minimum of four requests every second A developer notices that many API users run the same query by using the POST method. The developer wants to cache the POST request to optimize the API resources.

Which solution will meet these requirements'?

A. Configure the CloudFront cache Update the application to return cached content based upon the default request headers.

B. Override the cache method in me selected stage of API Gateway Select the POST method.

C. Save the latest request response in Lambda /tmp directory Update the Lambda function to check the /tmp directory

D. Save the latest request m AWS Systems Manager Parameter Store Modify the Lambda function to take the latest request response from Parameter Store

Answer: A

Explanation:

This solution will meet the requirements by using Amazon CloudFront, which is a content delivery network (CDN) service that speeds up the delivery of web content and APIs to end users. The developer can configure the CloudFront cache, which is a set of edge locations that store copies of popular or recently accessed content close to the viewers. The developer can also update the application to return cached content based upon the default request headers, which are a set of HTTP headers that CloudFront automatically forwards to the origin server and uses to determine whether an object in an edge location is still valid. By caching the POST requests, the developer can optimize the API resources and reduce the latency for repeated queries. Option B is not optimal because it will override the cache method in the selected stage of API Gateway, which is not possible or effective as API Gateway does not support caching for POST methods by default. Option C is not optimal because it will save the latest request response in Lambda /tmp directory, which is a local storage space that is available for each Lambda function invocation, not a cache that can be shared across multiple invocations or requests. Option D is not optimal because it will save the latest request in AWS Systems Manager Parameter Store, which is a service that provides secure and scalable storage for configuration data and secrets, not a cache for API responses.

Reference: [Amazon CloudFront], [Caching Content Based on Request Headers]

Question: 113

A company is building a micro services app1 cation that consists of many AWS Lambda functions. The development team wants to use AWS Serverless Application Model (AWS SAM) templates to automatically test the Lambda functions. The development team plans to test a small percentage of traffic that is directed to new updates before the team commits to a full deployment of the application.

Which combination of steps will meet these requirements in the MOST operationally efficient way? (Select TWO.)

A. Use AWS SAM CLI commands in AWS CodeDeploy lo invoke the Lambda functions lo lest the deployment

B. Declare the EventInvokeConfig on the Lambda functions in the AWS SAM templates with OnSuccess and OnFailure configurations.

C. Enable gradual deployments through AWS SAM templates.

D. Set the deployment preference type to Canary10Percen130Minutes Use hooks to test the deployment.

E. Set the deployment preference type to Linear10PefcentEvery10Minutes Use hooks to test the deployment.

Answer: C, D

Explanation:

This solution will meet the requirements by using AWS Serverless Application Model (AWS SAM) templates and gradual deployments to automatically test the Lambda functions. AWS SAM templates are configuration files that define serverless applications and resources such as Lambda functions. Gradual deployments are a feature of AWS SAM that enable deploying new versions of Lambda functions incrementally, shifting traffic gradually, and performing validation tests during deployment. The developer can enable gradual deployments through AWS SAM templates by adding a DeploymentPreference property to each Lambda function resource in the template. The developer can set the deployment preference type to Canary10Percent30Minutes, which means that 10 percent of traffic will be shifted to the new version of the Lambda function for 30 minutes before shifting 100 percent of traffic. The developer can also use hooks to test the deployment, which are custom Lambda functions that run before or after traffic shifting and perform validation tests or rollback actions.

Reference: [AWS Serverless Application Model (AWS SAM)], [Gradual Code Deployment]

Question: 114

A company is using AWS CloudFormation to deploy a two-tier application. The application will use Amazon RDS as its backend database. The company wants a solution that will randomly generate the database password during deployment. The solution also must automatically rotate the database password without requiring changes to the application.

What is the MOST operationally efficient solution that meets these requirements'?

A. Use an AWS Lambda function as a CloudFormation custom resource to generate and rotate the password.

B. Use an AWS Systems Manager Parameter Store resource with the SecureString data type to generate and rotate the password.

C. Use a cron daemon on the application s host to generate and rotate the password.

D. Use an AWS Secrets Manager resource to generate and rotate the password.

Answer: D

Explanation:

This solution will meet the requirements by using AWS Secrets Manager, which is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can use an AWS Secrets Manager resource in AWS CloudFormation template, which enables creating and managing secrets as part of a CloudFormation stack. The developer can use an AWS::SecretsManager::Secret resource type to generate and rotate the password for accessing RDS database during deployment. The developer can also specify a RotationSchedule property for the secret resource, which defines how often to rotate the secret and which Lambda function to use for rotation logic. Option A is not optimal

because it will use an AWS Lambda function as a CloudFormation custom resource, which may introduce additional complexity and overhead for creating and managing a custom resource and implementing rotation logic. Option B is not optimal because it will use an AWS Systems Manager Parameter Store resource with the SecureString data type, which does not support automatic rotation of secrets. Option C is not optimal because it will use a cron daemon on the application's host to generate and rotate the password, which may incur more costs and require more maintenance for running and securing a host.

Reference: [AWS Secrets Manager], [AWS::SecretsManager::Secret]

Question: 115

A developer has been asked to create an AWS Lambda function that is invoked any time updates are made to items in an Amazon DynamoDB table. The function has been created and appropriate permissions have been added to the Lambda execution role Amazon DynamoDB streams have been enabled for the table, but the function 15 still not being invoked.

Which option would enable DynamoDB table updates to invoke the Lambda function?

A. Change the StreamViewType parameter value to NEW_AND_OLOJMAGES for the DynamoDB table.

B. Configure event source mapping for the Lambda function.

C. Map an Amazon Simple Notification Service (Amazon SNS) topic to the DynamoDB streams.

D. Increase the maximum runtime (timeout) setting of the Lambda function.

Answer: B

Explanation:

This solution allows the Lambda function to be invoked by the DynamoDB stream whenever updates are made to items in the DynamoDB table. Event source mapping is a feature of Lambda that enables a function to be triggered by an event source, such as a DynamoDB stream, an Amazon Kinesis stream, or an Amazon Simple Queue Service (SQS) queue. The developer can configure event source mapping for the Lambda function using the AWS Management Console, the AWS CLI, or the AWS SDKs. Changing the StreamViewType parameter value to NEW_AND_OLD_IMAGES for the DynamoDB table will not affect the invocation of the Lambda function, but only change the information that is written to the stream record. Mapping an Amazon Simple Notification Service (Amazon SNS) topic to the DynamoDB stream will not invoke the Lambda function directly, but require an additional subscription from the Lambda function to the SNS topic. Increasing the maximum runtime (timeout) setting of the Lambda function will not affect the invocation of the Lambda function, but only change how long the function can run before it is terminated.

Reference: [Using AWS Lambda with Amazon DynamoDB], [Using AWS Lambda with Amazon SNS]

Question: 116

A developer needs to deploy an application running on AWS Fargate using Amazon ECS The application has environment variables that must be passed to a container for the application to initialize.

How should the environment variables be passed to the container?

A. Define an array that includes the environment variables under the environment parameter within the service definition.

B. Define an array that includes the environment variables under the environment parameter within the task definition.

C. Define an array that includes the environment variables under the entryPoint parameter within the task definition.

D. Define an array that includes the environment variables under the entryPoint parameter within the service definition.

Answer: B

Explanation:

This solution allows the environment variables to be passed to the container when it is launched by AWS Fargate using Amazon ECS. The task definition is a text file that describes one or more containers that form an application. It contains various parameters for configuring the containers, such as CPU and memory requirements, network mode, and environment variables. The environment parameter is an array of key-value pairs that specify environment variables to pass to a container. Defining an array that includes the environment variables under the entryPoint parameter within the task definition will not pass them to the container, but use them as command-line arguments for overriding the default entry point of a container. Defining an array that includes the environment variables under the environment or entryPoint parameter within the service definition will not pass them to the container, but cause an error because these parameters are not valid for a service definition.

Reference: [Task Definition Parameters], [Environment Variables]

Question: 117

A developer is storing sensitive data generated by an application in Amazon S3. The developer wants to encrypt the data at rest. A company policy requires an audit trail of when the AWS Key Management Service (AWS KMS) key was used and by whom.

Which encryption option will meet these requirements?

A. Server-side encryption with Amazon S3 managed keys (SSE-S3)

B. Server-side encryption with AWS KMS managed keys (SSE-KMS}

C. Server-side encryption with customer-provided keys (SSE-C)

D. Server-side encryption with self-managed keys

Answer: B

Explanation:

This solution meets the requirements because it encrypts data at rest using AWS KMS keys and provides an audit trail of when and by whom they were used. Server-side encryption with AWS KMS managed keys (SSE-KMS) is a feature of Amazon S3 that encrypts data using keys that are managed by AWS KMS. When SSE-KMS is enabled for an S3 bucket or object, S3 requests AWS KMS to generate data keys and encrypts data using these keys. AWS KMS logs every use of its keys in AWS CloudTrail, which records all API calls to AWS KMS as events. These events include information such as who made the request, when it was made, and which key was used. The company policy can use CloudTrail logs to audit critical events related to their data encryption and access. Server-side encryption with Amazon S3 managed keys (SSE-S3) also encrypts data at rest using keys that are managed by S3, but does not provide an audit trail

of key usage. Server-side encryption with customer-provided keys (SSE-C) and server-side encryption with self-managed keys also encrypt data at rest using keys that are provided or managed by customers, but do not provide an audit trail of key usage and require additional overhead for key management.

Reference: [Protecting Data Using Server-Side Encryption with AWS KMS-Managed Encryption Keys (SSE-KMS)], [Logging AWS KMS API calls with AWS CloudTrail]

Question: 118

A company has an ecommerce application. To track product reviews, the company's development team uses an Amazon DynamoDB table.

Every record includes the following

- A Review ID a 16-digrt universally unique identifier (UUID)
- A Product ID and User ID 16 digit UUIDs that reference other tables
- A Product Rating on a scale of 1-5
- An optional comment from the user

The table partition key is the Review ID. The most performed query against the table is to find the 10 reviews with the highest rating for a given product.

Which index will provide the FASTEST response for this query"?

A. A global secondary index (GSI) with Product ID as the partition key and Product Rating as the sort key

B. A global secondary index (GSI) with Product ID as the partition key and Review ID as the sort key

C. A local secondary index (LSI) with Product ID as the partition key and Product Rating as the sort key

D. A local secondary index (LSI) with Review ID as the partition key and Product ID as the sort key

Answer: A

Explanation:

This solution allows the fastest response for the query because it enables the query to use a single partition key value (the Product ID) and a range of sort key values (the Product Rating) to find the matching items. A global secondary index (GSI) is an index that has a partition key and an optional sort key that are different from those on the base table. A GSI can be created at any time and can be queried or scanned independently of the base table. A local secondary index (LSI) is an index that has the same partition key as the base table, but a different sort key. An LSI can only be created when the base table is created and must be queried together with the base table partition key. Using a GSI with Product ID as the partition key and Review ID as the sort key will not allow the query to use a range of sort key values to find the highest ratings. Using an LSI with Product ID as the partition key and Product Rating as the sort key will not work because Product ID is not the partition key of the base table. Using an LSI with Review ID as the

partition key and Product ID as the sort key will not allow the query to use a single partition key value to find the matching items.

Reference: [Global Secondary Indexes], [Querying]

Question: 119

A company needs to distribute firmware updates to its customers around the world.

Which service will allow easy and secure control of the access to the downloads at the lowest cost?

A. Use Amazon CloudFront with signed URLs for Amazon S3.

B. Create a dedicated Amazon CloudFront Distribution for each customer.

C. Use Amazon CloudFront with AWS Lambda@Edge.

D. Use Amazon API Gateway and AWS Lambda to control access to an S3 bucket.

Answer: A

Explanation:

This solution allows easy and secure control of access to the downloads at the lowest cost because it uses a content delivery network (CDN) that can cache and distribute firmware updates to customers around the world, and uses a mechanism that can restrict access to specific files or versions. Amazon CloudFront is a CDN that can improve performance, availability, and security of web applications by delivering content from edge locations closer to customers. Amazon S3 is a storage service that can store firmware updates in buckets and objects. Signed URLs are URLs that include additional information, such as an expiration date and time, that give users temporary access to specific objects in S3 buckets. The developer can use CloudFront to serve firmware updates from S3 buckets and use signed URLs to control who can download them and for how long. Creating a dedicated CloudFront distribution for each customer will incur unnecessary costs and complexity. Using Amazon CloudFront with AWS Lambda@Edge will require additional programming overhead to implement custom logic at the edge locations. Using Amazon API Gateway and AWS Lambda to control access to an S3 bucket will also require additional programming overhead and may not provide optimal performance or availability.

Reference: [Serving Private Content through CloudFront], [Using CloudFront with Amazon S3]

Question: 120

A developer is testing an application that invokes an AWS Lambda function asynchronously. During the testing phase the Lambda function fails to process after two retries.

How can the developer troubleshoot the failure?

A. Configure AWS CloudTrail logging to investigate the invocation failures.

B. Configure Dead Letter Queues by sending events to Amazon SQS for investigation.

C. Configure Amazon Simple Workflow Service to process any direct unprocessed events.

D. Configure AWS Config to process any direct unprocessed events.

Answer: B

Explanation:

This solution allows the developer to troubleshoot the failure by capturing unprocessed events in a queue for further analysis. Dead Letter Queues (DLQs) are queues that store messages that could not be processed by a service, such as Lambda, for various reasons, such as configuration errors, throttling limits, or permissions issues. The developer can configure DLQs for Lambda functions by sending events to either an Amazon Simple Queue Service (SQS) queue or an Amazon Simple Notification Service (SNS) topic. The developer can then inspect the messages in the queue or topic to identify and fix the root cause of the failure. Configuring AWS CloudTrail logging will not capture invocation failures for asynchronous Lambda invocations, but only record API calls made by or on behalf of Lambda. Configuring Amazon Simple Workflow Service (SWF) or AWS Config will not process any direct unprocessed events, but require additional integration and configuration.

Reference: [Using AWS Lambda with DLQs], [Asynchronous invocation]

Question: 121

A company is migrating its PostgreSQL database into the AWS Cloud. The company wants to use a database that will secure and regularly rotate database credentials. The company wants a solution that does not require additional programming overhead.

Which solution will meet these requirements?

A. Use Amazon Aurora PostgreSQL tor the database. Store the database credentials in AWS Systems Manager Parameter Store Turn on rotation.

B. Use Amazon Aurora PostgreSQL for the database. Store the database credentials in AWS Secrets Manager Turn on rotation.

C. Use Amazon DynamoDB for the database. Store the database credentials in AWS Systems Manager Parameter Store Turn on rotation.

D. Use Amazon DynamoDB for the database. Store the database credentials in AWS Secrets Manager Turn on rotation.

Answer: B

Explanation:

This solution meets the requirements because it uses a PostgreSQL-compatible database that can secure and regularly rotate database credentials without requiring additional programming overhead. Amazon Aurora PostgreSQL is a relational database service that is compatible with PostgreSQL and offers high performance, availability, and scalability. AWS Secrets Manager is a service that helps you protect secrets needed to access your applications, services, and IT resources. You can store database credentials in AWS Secrets Manager and use them to access your Aurora PostgreSQL database. You can also enable automatic rotation of your secrets according to a schedule or an event. AWS Secrets Manager handles the complexity of rotating secrets for you, such as generating new passwords and updating your database with the new credentials. Using Amazon DynamoDB for the database will not meet the requirements because it is a NoSQL database that is not compatible with PostgreSQL. Using AWS Systems Manager Parameter Store for storing and rotating

database credentials will require additional programming overhead to integrate with your database.

Reference: [What Is Amazon Aurora?], [What Is AWS Secrets Manager?]

Question: 122

A developer is creating a mobile application that will not require users to log in.

What is the MOST efficient method to grant users access to AWS resources'?

A. Use an identity provider to securely authenticate with the application.

B. Create an AWS Lambda function to create an 1AM user when a user accesses the application.

C. Create credentials using AWS KMS and apply these credentials to users when using the application.

D. Use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources.

Answer: D

Explanation:

This solution is the most efficient method to grant users access to AWS resources without requiring them to log in. Amazon Cognito is a service that provides user sign-up, sign-in, and access control for web and mobile applications. Amazon Cognito identity pools support both authenticated and unauthenticated users. Unauthenticated users receive access to your AWS resources even if they aren't logged in with any of your identity providers (IdPs). You can use Amazon Cognito to associate unauthenticated users with an IAM role that has limited access to resources, such as Amazon S3 buckets or DynamoDB tables. This degree of access is useful to display content to users before they log in or to allow them to perform certain actions without signing up. Using an identity provider to securely authenticate with the application will require users to log in, which does not meet the requirement. Creating an AWS Lambda function to create an IAM user when a user accesses the application will incur unnecessary costs and complexity, and may pose security risks if not implemented properly. Creating credentials using AWS KMS and applying them to users when using the application will also incur unnecessary costs and complexity, and may not provide fine-grained access control for resources.

Reference: Switching unauthenticated users to authenticated users (identity pools), Allow user access to your API without authentication (Anonymous user access)

Question: 123

A company has developed a new serverless application using AWS Lambda functions that will be deployed using the AWS Serverless Application Model (AWS SAM) CLI.

Which step should the developer complete prior to deploying the application?

A. Compress the application to a zip file and upload it into AWS Lambda.

B. Test the new AWS Lambda function by first tracing it m AWS X-Ray.

C. Bundle the serverless application using a SAM package.

D. Create the application environment using the eb create my-env command.

Answer: C

Explanation:

This step should be completed prior to deploying the application because it prepares the application artifacts for deployment. The AWS Serverless Application Model (AWS SAM) is a framework that simplifies building and deploying serverless applications on AWS. The AWS SAM CLI is a commandline tool that helps you create, test, and deploy serverless applications using AWS SAM templates. The sam package command bundles the application artifacts, such as Lambda function code and API definitions, and uploads them to an Amazon S3 bucket. The command also returns a CloudFormation template that is ready to be deployed with the sam deploy command. Compressing the application to a zip file and uploading it to AWS Lambda will not work because it does not use AWS SAM templates or CloudFormation. Testing the new Lambda function by first tracing it in AWS X-Ray will not prepare the application for deployment, but only monitor its performance and errors. Creating the application environment using the eb create my-env command will not work because it is a command for AWS Elastic Beanstalk, not AWS SAM.

Question: 124

A company wants to automate part of its deployment process. A developer needs to automate the process of checking for and deleting unused resources that supported previously deployed stacks but that are no longer used.

The company has a central application that uses the AWS Cloud Development Kit (AWS CDK) to manage all deployment stacks. The stacks are spread out across multiple accounts. The developer's solution must integrate as seamlessly as possible within the current deployment process.

Which solution will meet these requirements with the LEAST amount of configuration?

A. In the central AWS CDK application, write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. Create an AWS CloudPormation template from a JSON file. Use the template to attach the function code to an AWS Lambda function and lo invoke the Lambda function when the deployment slack runs.

B. In the central AWS CDK application. write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. Create an AWS CDK custom resource Use the custom resource to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.

C. In the central AWS CDK, write a handler function m the code that uses AWS SDK calls to check for and delete unused resources. Create an API in AWS Amplify Use the API to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.

D. In the AWS Lambda console write a handler function in the code that uses AWS SDK calls to check for and delete unused resources. Create an AWS CDK custom resource. Use the custom resource to import the Lambda function into the stack and to Invoke the Lambda function when the deployment stack runs.

Answer: B

Explanation:

This solution meets the requirements with the least amount of configuration because it uses a feature of AWS CDK that allows custom logic to be executed during stack deployment or deletion. The AWS Cloud Development Kit (AWS CDK) is a software development framework that allows you to define cloud infrastructure as code and provision it through CloudFormation. An AWS CDK custom resource is a construct that enables you to create resources that are not natively supported by CloudFormation or perform tasks that are not supported by CloudFormation during stack deployment or deletion. The developer can write a handler function in the code that uses AWS SDK calls to check for and delete unused resources, and create an AWS CDK custom resource that attaches the function code to a Lambda function and invokes it when the deployment stack runs. This way, the developer can automate the cleanup process without requiring additional configuration or integration. Creating a CloudFormation template from a JSON file will require additional configuration and integration with the central AWS CDK application. Creating an API in AWS Amplify will require additional configuration and integration with the central AWS CDK application and may not provide optimal performance or availability. Writing a handler function in the AWS Lambda console will require additional configuration and integration with the central AWS CDK application.

Reference: [AWS Cloud Development Kit (CDK)], [Custom Resources]

Question: 125

A company built a new application in the AWS Cloud. The company automated the bootstrapping of new resources with an Auto Scaling group by using AWS Cloudf-ormation templates. The bootstrap scripts contain sensitive data.

The company needs a solution that is integrated with CloudFormation to manage the sensitive data in the bootstrap scripts.

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

A. Put the sensitive data into a CloudFormation parameter. Encrypt the CloudFormation templates by using an AWS Key Management Service (AWS KMS) key.

B. Put the sensitive data into an Amazon S3 bucket Update the CloudFormation templates to download the object from Amazon S3 during bootslrap.

C. Put the sensitive data into AWS Systems Manager Parameter Store as a secure string parameter. Update the CloudFormation templates to use dynamic references to specify template values.

D. Put the sensitive data into Amazon Elastic File System (Amazon EPS) Enforce EFS encryption after file system creation. Update the CloudFormation templates to retrieve data from Amazon EFS.

Answer: C

Explanation:

This solution meets the requirements in the most secure way because it uses a service that is integrated with CloudFormation to manage sensitive data in encrypted form. AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. You can store sensitive data as secure string parameters, which are encrypted using an AWS Key Management Service (AWS KMS) key of your choice. You can also use dynamic references in your CloudFormation templates to specify template values that are stored in Parameter Store or Secrets Manager without having to include them in your templates. Dynamic

references are resolved only during stack creation or update operations, which reduces exposure risks for sensitive data. Putting sensitive data into a CloudFormation parameter will not encrypt them or protect them from unauthorized access. Putting sensitive data into an Amazon S3 bucket or Amazon Elastic File System (Amazon EFS) will require additional configuration and integration with CloudFormation and may not provide fine-grained access control or encryption for sensitive data.

Reference: [What Is AWS Systems Manager Parameter Store?], [Using Dynamic Reference to Specify Template Values]

Question: 126

A company needs to set up secure database credentials for all its AWS Cloud resources. The company's resources include Amazon RDS DB instances Amazon DocumentDB clusters and Amazon Aurora DB instances. The company's security policy mandates that database credentials be encrypted at rest and rotated at a regular interval.

Which solution will meet these requirements MOST securely?

A. Set up IAM database authentication for token-based access. Generate user tokens to provide centralized access to RDS DB instances. Amazon DocumentDB clusters and Aurora DB instances.

B. Create parameters for the database credentials in AWS Systems Manager Parameter Store Set the Type parameter to Secure Sting. Set up automatic rotation on the parameters.

C. Store the database access credentials as an encrypted Amazon S3 object in an S3 bucket Block all public access on the S3 bucket. Use S3 server-side encryption to set up automatic rotation on the encryption key.

D. Create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console. Create secrets for the database credentials in Secrets Manager Set up secrets rotation on a schedule.

Answer: D

Explanation:

This solution will meet the requirements by using AWS Secrets Manager, which is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console, which provides a sample code for rotating secrets for RDS DB instances, Amazon DocumentDB clusters, and Amazon Aurora DB instances. The developer can also create secrets for the database credentials in Secrets Manager, which encrypts them at rest and provides secure access to them. The developer can set up secrets rotation on a schedule, which changes the database credentials periodically according to a specified interval or event. Option A is not optimal because it will set up IAM database authentication for token-based access, which may not be compatible with all database engines and may require additional configuration and management of IAM roles or users. Option B is not optimal because it will create parameters for the database credentials in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option C is not optimal because it will store the database access credentials as an encrypted Amazon S3 object in an S3 bucket, which may introduce additional costs and complexity for accessing and securing the data.

Reference: [AWS Secrets Manager], [Rotating Your AWS Secrets Manager Secrets]

Question: 127

A developer has created an AWS Lambda function that makes queries to an Amazon Aurora MySQL DB instance. When the developer performs a test the OB instance shows an error for too many connections.

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

A. Create a read replica for the DB instance Query the replica DB instance instead of the primary DB instance.

B. Migrate the data lo an Amazon DynamoDB database.

C. Configure the Amazon Aurora MySQL DB instance tor Multi-AZ deployment.

D. Create a proxy in Amazon RDS Proxy Query the proxy instead of the DB instance.

Answer: D

Explanation:

This solution will meet the requirements by using Amazon RDS Proxy, which is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. The developer can create a proxy in Amazon RDS Proxy, which sits between the application and the DB instance and handles connection management, pooling, and routing. The developer can query the proxy instead of the DB instance, which reduces the number of open connections to the DB instance and avoids errors for too many connections. Option A is not optimal because it will create a read replica for the DB instance, which may not solve the problem of too many connections as read replicas also have connection limits and may incur additional costs. Option B is not optimal because it will migrate the data to an Amazon DynamoDB database, which may introduce additional complexity and overhead for migrating and accessing data from a different database service. Option C is not optimal because it will configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment, which may improve availability and durability of the DB instance but not reduce the number of connections.

Reference: [Amazon RDS Proxy], [Working with Amazon RDS Proxy]

Question: 128

A company uses Amazon API Gateway to expose a set of APIs to customers. The APIs have caching enabled in API Gateway. Customers need a way to invalidate the cache for each API when they test the API.

What should a developer do to give customers the ability to invalidate the API cache?

A. Ask the customers to use AWS credentials to call the InvalidateCache API operation.

B. Attach an InvalidateCache policy to the IAM execution role that the customers use to invoke the API. Ask the customers to send

a request that contains the

HTTP header when they make an API call.

C. Ask the customers to use the AWS SDK API Gateway class to invoke the InvalidateCache API operation.

D. Attach an InvalidateCache policy to the IAM execution role that the customers use to invoke the API. Ask the customers to add the INVALIDATE_CACHE query

string parameter when they make an API call.

_____ Answer: D

Explanation:

Question: 129

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions. When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment. If there are no issues, all traffic must switch over to the new version.

Which change to the AWS SAM template will meet these requirements?

A. Set the Deployment Preference Type to Canaryl OPercent10Minutes. Set the AutoPublishAlias property to the Lambda alias.

B. Set the Deployment Preference Type to Linearl OPercentEveryIOMinutes. Set AutoPublishAlias property to the Lambda alias.

C. Set the Deployment Preference Type to Canaryl OPercentIOMinutes. Set the PreTraffic and PostTraffic properties to the Lambda alias.

D. Set the Deployment Preference Type to Linearl OPercentEvery10Minutes. Set PreTraffic and PostTraffic properties to the Lambda alias.

_____ Answer: A

Explanation:

The Deployment Preference Type property specifies how traffic should be shifted between versions of a Lambda function! The Canary10Percent10Minutes option means that 10% of the traffic is immediately shifted to the new version, and after 10 minutes, the remaining 90% of the traffic is shifted!. This matches the requirement of shifting 10% of the traffic for the first 10 minutes, and then

switching all traffic to the new version.

The AutoPublishAlias property enables AWS SAM to automatically create and update a Lambda alias that points to the latest version of the function! This is required to use the Deployment Preference Type property! The alias name can be specified by the developer, and it can be used to invoke the function with the latest code.

Question: 130

A developer is preparing to begin development of a new version of an application. The previous version of the application is deployed in a production environment. The developer needs to deploy fixes and updates to the current version during the development of the new version of the application. The code for the new version of the application is stored in AWS CodeCommit.

Which solution will meet these requirements?

A. From the main branch, create a feature branch for production bug fixes. Create a second feature branch from the main branch for development of the new version.

B. Create a Git tag of the code that is currently deployed in production. Create a Git tag for the development of the new version. Push the two tags to the CodeCommit repository.

C. From the main branch, create a branch of the code that is currently deployed in production. Apply an IAM policy that ensures no other other users can push or merge to the branch.

D. Create a new CodeCommit repository for development of the new version of the application. Create a Git tag for the development of the new version.

Answer: A

Explanation:

A feature branch is a branch that is created from the main branch to work on a specific feature or task! Feature branches allow developers to isolate their work from the main branch and avoid conflicts with other changes!. Feature branches can be merged back to the main branch when the feature or task is completed and tested!.

In this scenario, the developer needs to maintain two parallel streams of work: one for fixing and updating the current version of the application that is deployed in production, and another for developing the new version of the application. The developer can use feature branches to achieve this goal.

The developer can create a feature branch from the main branch for production bug fixes. This branch will contain the code that is currently deployed in production, and any fixes or updates that need to be applied to it. The developer can push this branch to the CodeCommit repository and use it to deploy changes to the production environment.

The developer can also create a second feature branch from the main branch for development of the new version of the application. This branch will contain the code that is under development for the new version, and any changes or enhancements that are part of it. The developer can push this branch to the CodeCommit repository and use it to test and deploy the new version of the application in a separate environment.

By using feature branches, the developer can keep the main branch stable and clean, and avoid mixing code from different versions of the application. The developer can also easily switch between branches and merge them when needed.

Question: 131

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda. The development team tests the API and validates responses for the known use cases before deploying the API to the production environment.

The developer wants to make the REST API available for testing by using API Gateway locally.

Which AWS Serverless Application Model Command Line Interface (AWS SAM CLI) subcommand will meet these requirements?

A. Sam local invoke

B. Sam local generate-event

C. Sam local start-lambda

D. Sam local start-api

Answer: D

Explanation:

The sam local start-api subcommand allows you to run your serverless application locally for quick development and testing! It creates a local HTTP server that acts as a proxy for API Gateway and invokes your Lambda functions based on the AWS SAM template! You can use the sam local start- api subcommand to test your REST API locally by sending HTTP requests to the local endpoint!

Question: 132

A developer is writing an application that will retrieve sensitive data from a third-party system. The application will format the data into a PDF file. The PDF file could be more than ! MB. The application will encrypt the data to disk by using AWS Key Management Service (AWS KMS). The application will decrypt the file when a user requests to download it. The retrieval and formatting portions of the application are complete.

The developer needs to use the GenerateDataKey API to encrypt the PDF file so that the PDF file can be decrypted later. The developer needs to use an AWS KMS symmetric customer managed key for encryption.

Which solutions will meet these requirements?

A. Write the encrypted key from the GenerateDataKey API to disk for later use. Use the plaintext key from the GenerateDataKey API and a symmetric encryption algorithm to encrypt the file.

B. Write the plain text key from the GenerateDataKey API to disk for later use. Use the encrypted key from the GenerateDataKey API and a symmetric encryption algorithm to encrypt the file.

C. Write the encrypted key from the GenerateDataKey API to disk for later use. Use the plaintext key from the GenerateDataKey API to encrypt the file by using the KMS Encrypt API

D. Write the plain text key from the GenerateDataKey API to disk for later use. Use the encrypted key from the GenerateDataKey API to encrypt the file by using the KMS Encrypt API

Answer: A

Explanation:

The GenerateDataKey API returns a data key that is encrypted under a symmetric encryption KMS key that you specify, and a plaintext copy of the same data key1. The data key is a random byte string that can be used with any standard encryption algorithm, such as AES or SM42. The plaintext data key can be used to encrypt or decrypt data outside of AWS KMS, while the encrypted data key can be stored with the encrypted data and later decrypted by AWS KMS1.

In this scenario, the developer needs to use the GenerateDataKey API to encrypt the PDF file so that it can be decrypted later. The developer also needs to use an AWS KMS symmetric customer managed key for encryption. To achieve this, the developer can follow

these steps:

Call the GenerateDataKey API with the symmetric customer managed key ID and the desired length or specification of the data key. The API will return an encrypted data key and a plaintext data key.

Write the encrypted data key to disk for later use. This will allow the developer to decrypt the data key and the PDF file later by using AWS KMS.

Use the plaintext data key and a symmetric encryption algorithm to encrypt the PDF file. The developer can use any standard encryption library or tool to perform this operation, such as OpenSSL or AWS Encryption SDK.

Discard the plaintext data key from memory as soon as possible after using it. This will prevent unauthorized access or leakage of the data key.

Question: 133

A developer is optimizing an AWS Lambda function and wants to test the changes in production on a small percentage of all traffic. The Lambda function serves requests to a REST API in Amazon API Gateway. The developer needs to deploy their changes and perform a test in production without changing the API Gateway URL.

Which solution will meet these requirements?

A. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function version. Upload and publish the optimized Lambda function code. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. Update the API Gateway endpoint to use the $LATEST version of the Lambda function. Publish the API to the canary stage.

B. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function version. Upload and publish the optimized Lambda function code. Update the API Gateway endpoint to use the $LATEST version of the Lambda function.
Deploy a new API Gateway stage.

C. Define an alias on the $LATEST version of the Lambda function. Update the API Gateway endpoint to reference the new Lambda function alias. Upload and
publish the optimized Lambda function code. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. Update the API Gateway endpoint to use the SLAT EST version of the Lambda function. Publish to

the canary stage.

D. Define a function version for the currently deployed production Lambda function. Update the API Gateway endpoint to reference the new Lambda function version. Upload and publish the optimized Lambda function code. Update the API Gateway endpoint to use the $LATEST version of the Lambda function.
Deploy the API to the production API Gateway stage.

Answer: C

Explanation:

A Lambda alias is a pointer to a specific Lambda function version or another alias1. A Lambda alias allows you to invoke different versions of a function using the same name1. You can also split traffic between two aliases by assigning weights to them1.

In this scenario, the developer needs to test their changes in production on a small percentage of all traffic without changing the API Gateway URL. To achieve this, the developer can follow these steps:

Define an alias on the $LATEST version of the Lambda function. This will create a new alias that points to the latest code of the function.

Update the API Gateway endpoint to reference the new Lambda function alias. This will make the API Gateway invoke the alias instead of a specific version of the function.

Upload and publish the optimized Lambda function code. This will update the $LATEST version of the function with the new code.

On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release. This will enable API Gateway to perform a canary deployment on a new API2. A canary deployment is a software development strategy in which a new version of an API is deployed for testing purposes, and the base version remains deployed as a production release for normal operations on the same stage2. The canary release receives a small percentage of API traffic and the production release takes up the rest2.

Update the API Gateway endpoint to use the $LATEST version of the Lambda function. This will make the canary release invoke the latest code of the function, which contains the optimized changes.

Publish to the canary stage. This will deploy the changes to a subset of users for testing.

By using this solution, the developer can test their changes in production on a small percentage of all traffic without changing the API Gateway URL. The developer can also monitor and compare metrics between the canary and production releases, and promote or disable the canary as needed2.

Question: 134

A company has an application that stores data in Amazon RDS instances. The application periodically experiences surges of high

traffic that cause performance problems.

During periods of peak traffic, a developer notices a reduction in query speed in all database queries.

The team's technical lead determines that a multi-threaded and scalable caching solution should be used to ofload the heavy read traffic. The solution needs to improve performance.

Which solution will meet these requirements with the LEAST complexity?

A. Use Amazon ElastiCache for Memcached to ofload read requests from the main database.

B. Replicate the data to Amazon DynamoDB. Set up a DynamoDB Accelerator (DAX) cluster.

C. Configure the Amazon RDS instances to use Multi-AZ deployment with one standby instance. Ofload read requests from the main database to the standby instance.

D. Use Amazon ElastiCache for Redis to ofload read requests from the main database.

Answer: D

Explanation:

Amazon ElastiCache for Memcached is a fully managed, multithreaded, and scalable in-memory keyvalue store that can be used to cache frequently accessed data and improve application performance! By using Amazon ElastiCache for Memcached, the developer can reduce the load on the main database and handle high traffic surges more efficiently.

To use Amazon ElastiCache for Memcached, the developer needs to create a cache cluster with one or more nodes, and configure the application to store and retrieve data from the cache cluster2. The developer can use any of the supported Memcached clients to interact with the cache cluster3. The developer can also use Auto Discovery to dynamically discover and connect to all cache nodes in a cluster4.

Amazon ElastiCache for Memcached is compatible with the Memcached protocol, which means that the developer can use existing tools and libraries that work with Memcachedl. Amazon ElastiCache for Memcached also supports data partitioning, which allows the developer to distribute data among multiple nodes and scale out the cache cluster as needed.

Using Amazon ElastiCache for Memcached is a simple and effective solution that meets the requirements with the least complexity. The developer does not need to change the database schema, migrate data to a different service, or use a different caching model. The developer can leverage the existing Memcached ecosystem and easily integrate it with the application.

Question: 135

An application that runs on AWS receives messages from an Amazon Simple Queue Service (Amazon SQS) queue and processes the messages in batches. The

application sends the data to another SQS queue to be consumed by another legacy application. The legacy system can take up to 5 minutes to process some transaction data.

A developer wants to ensure that there are no out-of-order updates in the legacy system. The developer cannot alter the behavior of the legacy system.

Which solution will meet these requirements?

A. Use an SQS FIFO queue. Configure the visibility timeout value.

B. Use an SQS standard queue with a SendMessageBatchRequestEntry data type. Configure the DelaySeconds values.

C. Use an SQS standard queue with a SendMessageBatchRequestEntry data type. Configure the visibility timeout value.

D. Use an SQS FIFO queue. Configure the DelaySeconds value.

Answer: A

Explanation:

An SQS FIFO queue is a type of queue that preserves the order of messages and ensures that each message is delivered and processed only once1. This is suitable for the scenario where the developer wants to ensure that there are no out-of-order updates in the legacy system.

The visibility timeout value is the amount of time that a message is invisible in the queue after a consumer receives it2. This prevents other consumers from processing the same message simultaneously. If the consumer does not delete the message before the visibility timeout expires, the message becomes visible again and another consumer can receive it2.

In this scenario, the developer needs to configure the visibility timeout value to be longer than the maximum processing time of the legacy system, which is 5 minutes. This will ensure that the message remains invisible in the queue until the legacy system finishes processing it and deletes it. This will prevent duplicate or out-of-order processing of messages by the legacy system.

Question: 136

A developer is troubleshooting an application in an integration environment. In the application, an Amazon Simple Queue Service (Amazon SQS) queue consumes messages and then an AWS Lambda function processes the messages. The Lambda function transforms the messages and makes an API call to a third-party service.

There has been an increase in application usage. The third-party API frequently returns an HTTP 429 Too Many Requests error message. The error message prevents a significant number of messages from being processed successfully.
How can the developer resolve this issue?

A. Increase the SQS event source's batch size setting.

B. Configure provisioned concurrency for the Lambda function based on the third-party API's documented rate limits.

C. Increase the retry attempts and maximum event age in the Lambda function's asynchronous configuration.

D. Configure maximum concurrency on the SQS event source based on the third-party service's documented rate limits.

Answer: D

Explanation:

Maximum concurrency for SQS as an event source allows customers to control the maximum concurrent invokes by the SQS event source!. When multiple SQS event sources are configured to a function, customers can control the maximum concurrent invokes of individual SQS event source!.

In this scenario, the developer needs to resolve the issue of the third-party API frequently returning an HTTP 429 Too Many Requests error message, which prevents a significant number of messages from being processed successfully. To achieve this, the developer can follow these steps:

Find out the documented rate limits of the third-party API, which specify how many requests can be made in a given time period.

Configure maximum concurrency on the SQS event source based on the rate limits of the third-party API. This will limit the number of concurrent invokes by the SQS event source and prevent exceeding the rate limits of the third-party API.

Test and monitor the application performance and adjust the maximum concurrency value as needed.

By using this solution, the developer can reduce the frequency of HTTP 429 errors and improve the message processing success rate. The developer can also avoid throttling or blocking by the third- party API.

Question: 137

An online sales company is developing a serverless application that runs on AWS. The application uses an AWS Lambda function that calculates order success rates and stores the data in an Amazon DynamoDB table. A developer wants an efficient way to invoke the Lambda function every !5 minutes.

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

A. Create an Amazon EventBridge rule that has a rate expression that will run the rule every !5 minutes. Add the Lambda function as the target of the EventBridge rule.

B. Create an AWS Systems Manager document that has a script that will invoke the Lambda function on Amazon EC2. Use a Systems Manager Run Command task to run the shell script every !5 minutes.

C. Create an AWS Step Functions state machine. Configure the state machine to invoke the Lambda function execution role at a specified interval by using a Wait state. Set the interval to !5 minutes.

D. Provision a small Amazon EC2 instance. Set up a cron job that invokes the Lambda function every !5 minutes.

_____ Answer: A

Explanation:

The best solution for this requirement is option A. Creating an Amazon EventBridge rule that has a rate expression that will run the rule every !5 minutes and adding the Lambda function as the target of the EventBridge rule is the most efficient way to invoke the Lambda function periodically. This solution does not require any additional resources or development effort, and it leverages the built- in scheduling capabilities of EventBridgel.

Question: 138 _____

A developer is migrating an application to Amazon Elastic Kubernetes Service (Amazon EKS). The developer migrates the application to Amazon Elastic Container Registry (Amazon ECR) with an EKS cluster.

As part of the application migration to a new backend, the developer creates a new AWS account. The developer makes configuration changes to the application to point the application to the new AWS account and to use new backend resources. The developer successfully tests the changes within the application by deploying the pipeline.

The Docker image build and the pipeline deployment are successful, but the application is still connecting to the old backend. The developer finds that the application's configuration is still referencing the original EKS cluster and not referencing the new backend resources.

Which reason can explain why the application is not connecting to the new resources?

A. The developer did not successfully create the new AWS account.

B. The developer added a new tag to the Docker image.

C. The developer did not update the Docker image tag to a new version.

D. The developer pushed the changes to a new Docker image tag.

Answer: C

Explanation:

The correct answer is C. The developer did not update the Docker image tag to a new version.

C . The developer did not update the Docker image tag to a new version. This is correct. When deploying an application to Amazon EKS, the developer needs to specify the Docker image tag that contains the application code and configuration. If the developer does not update the Docker image tag to a new version after making changes to the application, the EKS cluster will continue to use the old Docker image tag that references the original backend resources. To fix this issue, the developer should update the Docker image tag to a new version and redeploy the application to the EKS cluster.

A . The developer did not successfully create the new AWS account. This is incorrect. The creation of a new AWS account is not related to the application's connection to the backend resources. The developer can use any AWS account to host the EKS cluster and the backend resources, as long as they have the proper permissions and configurations.

B . The developer added a new tag to the Docker image. This is incorrect. Adding a new tag to the Docker image is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image.

D . The developer pushed the changes to a new Docker image tag. This is incorrect. Pushing the changes to a new Docker image tag is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image.

Reference:

1: Amazon EKS User Guide, “Deploying applications to your Amazon EKS cluster”, https://docs.aws.amazon.com/eks/latest/userguide/deploying-applications.html

2: Amazon ECR User Guide, "Pushing an image", https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html

3: Amazon EKS User Guide, "Updating an Amazon EKS cluster", https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html

Question: 139

A developer at a company needs to create a small application that makes the same API call once each day at a designated time. The company does not have infrastructure in the AWS Cloud yet, but the company wants to implement this functionality on AWS.

Which solution meets these requirements in the MOST operationally efficient manner?

A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS).

B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2.

C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

D. Use an AWS Batch job that is submitted to an AWS Batch job queue.

Answer: C

Explanation:

The correct answer is C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.

C . Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event. This is correct. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging1. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources2. EventBridge can create rules that run on a schedule, either at regular intervals or at specific times and dates, and invoke targets such as Lambda functions3. This solution meets the requirements of creating a small application that makes the same API call once each day at a designated time, without requiring any infrastructure in the AWS Cloud or any operational overhead.

A . Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS). This is incorrect. Amazon EKS is a fully managed Kubernetes service that allows you to run containerized applications on AWS4. Kubernetes cron jobs are tasks that run periodically on a given schedule5. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EKS cluster, which would incur additional costs and complexity.

B . Use an Amazon Linux crontab scheduled job that runs on Amazon EC2. This is incorrect. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud6. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date7. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EC2 instance, which would incur additional costs and complexity.

D . Use an AWS Batch job that is submitted to an AWS Batch job queue. This is incorrect. AWS Batch enables you to run batch computing workloads on the AWS Cloud8. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially on compute environments9. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to configure and manage an AWS Batch environment, which would incur additional costs and complexity.

Reference:

1: What is AWS Lambda? - AWS Lambda

2: What is Amazon EventBridge? - Amazon EventBridge

3: Creating an Amazon EventBridge rule that runs on a schedule - Amazon EventBridge

4: What is Amazon EKS? - Amazon EKS

5: CronJob - Kubernetes

6: What is Amazon EC2? - Amazon EC2

7: Crontab in Linux with 20 Useful Examples to Schedule Jobs - Tecmint

8: What is AWS Batch? - AWS Batch

9: Jobs - AWS Batch

Question: 140

An developer is building a serverless application by using the AWS Serverless Application Model (AWS SAM). The developer is currently testing the application in a development environment. When the application is nearly finsihed, the developer will need to set up additional testing and staging environments for a quality assurance team.

The developer wants to use a feature of the AWS SAM to set up deployments to multiple environments.

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

A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the sam deploy command and the --config-env flag that corresponds to the each environment.

B. Create additional AWS SAM templates for each testing and staging environment. Write a custom shell script that uses the sam deploy command and the --template-file flag to deploy updates to the environments.

C. Create one AWS SAM configuration file that has default parameters. Perform updates to the testing and staging environments by using the —parameter-overrides flag in the AWS SAM CLI and the parameters that the updates will override.

D. Use the existing AWS SAM template. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment. Deploy updates to the testing and staging environments by using the sam deploy command.

Answer: A

Explanation:

The correct answer is A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the sam deploy command and the --config-env flag that corresponds to the each environment.

A . Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the sam deploy command and the --config-env flag that corresponds to the each environment. This is correct. This solution will meet the requirements with the least development effort, because it uses a feature of the AWS SAM CLI that supports a project-level configuration file that can be used to configure AWS SAM CLI command parameter valuesl. The configuration file can have multiple environments, each with its own set of parameter values, such as stack name, region, capabilities, and more2. The developer can use the --config-env option to specify which environment to use when deploying the application3. This way, the developer can avoid creating multiple templates or scripts, or manually overriding parameters for each environment.

B . Create additional AWS SAM templates for each testing and staging environment. Write a custom shell script that uses the sam deploy command and the --template-file flag to deploy updates to the environments. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires creating and maintaining multiple templates and scripts for each

environment. This can introduce duplication, inconsistency, and complexity in the deployment process.

C . Create one AWS SAM configuration file that has default parameters. Perform updates to the testing and staging environments by using the —parameter-overrides flag in the AWS SAM CLI and the parameters that the updates will override. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires manually specifying and overriding parameters for each environment every time the developer deploys the application. This can be error-prone, tedious, and inefficient.

D . Use the existing AWS SAM template. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment. Deploy updates to the testing and staging environments by using the sam deploy command. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires modifying the existing template and adding complexity to the resource definitions for each environment. This can also make it difficult to manage and track changes across different environments.

Reference:

1: AWS SAM CLI configuration file - AWS Serverless Application Model

2: Configuration file basics - AWS Serverless Application Model

3: Specify a configuration file - AWS Serverless Application Model

Question: 141

A company notices that credentials that the company uses to connect to an external software as a service (SaaS) vendor are stored in a configuration file as plaintext.

The developer needs to secure the API credentials and enforce automatic credentials rotation on a quarterly basis.

Which solution will meet these requirements MOST securely?

A. Use AWS Key Management Service (AWS KMS) to encrypt the configuration file. Decrypt the configuration file when users make API calls to the SaaS vendor. Enable rotation.

B. Retrieve temporary credentials from AWS Security Token Service (AWS STS) every 15 minutes. Use the temporary credentials when users make API calls to the SaaS vendor.

C. Store the credentials in AWS Secrets Manager and enable rotation. Configure the API to have Secrets Manager access.

D. Store the credentials in AWS Systems Manager Parameter Store and enable rotation. Retrieve the credentials when users make API calls to the SaaS vendor.

Answer: C

Explanation:

Store the credentials in AWS Secrets Manager and enable rotation. Configure the API to have Secrets Manager access. This is correct. This solution will meet the requirements most securely, because it uses a service that is designed to store and manage secrets such as API credentials. AWS Secrets Manager helps you protect access to your applications, services, and IT resources by enabling you to rotate, manage, and retrieve secrets throughout their lifecycle!. You can store secrets such as passwords, database strings, API keys, and license codes as encrypted values2. You can also configure automatic rotation of your secrets on a schedule that you specify3. You can use the AWS SDK or CLI to retrieve secrets from Secrets Manager when you need them4. This way, you can avoid storing credentials in plaintext files or hardcoding them in your code.

Question: 142

A developer wants to deploy a new version of an AWS Elastic Beanstalk application. During deployment, the application must maintain full capacity and avoid service interruption. Additionally, the developer must minimize the cost of additional resources that support the deployment.

Which deployment method should the developer use to meet these requirements?

A. All at once

B. Rolling with additional batch

C. Blue/green

D. Immutable

Answer: D

Explanation:

The immutable deployment method is the best option for this scenario, because it meets the requirements of maintaining full capacity,

avoiding service interruption, and minimizing the cost of additional resources.

The immutable deployment method creates a new set of instances in a separate Auto Scaling group and deploys the new version of the application to them. Then, it swaps the new instances with the old ones and terminates the old instances. This way, the application maintains full capacity during the deployment and avoids any downtime. The cost of additional resources is also minimized, because the new instances are only created for a short time and then replaced by the old ones.

The other deployment methods do not meet all the requirements:

The all at once method deploys the new version to all instances simultaneously, which causes a short period of downtime and reduced capacity.

The rolling with additional batch method deploys the new version in batches, but for the first batch it creates new instances instead of using the existing ones. This increases the cost of additional resources and reduces the capacity of the original environment.

The blue/green method creates a new environment with a new set of instances and deploys the new version to them. Then, it swaps the URLs between the old and new environments. This method maintains full capacity and avoids service interruption, but it also increases the cost of additional resources significantly, because it duplicates the entire environment.

Question: 143

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions.

When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment. If there are no issues, all traffic must switch over to the new version.

Which change to the AWS SAM template will meet these requirements?

A. Set the Deployment Preference Type to Canary10Percent10Minutes. Set the AutoPublishAlias property to the Lambda alias.

B. Set the Deployment Preference Type to Linear10PercentEvery10Minutes. Set AutoPublishAlias property to the Lambda alias.

C. Set the Deployment Preference Type to Canary10Percent10Minutes. Set the PreTraffic and PostTraffic properties to the Lambda alias.

D. Set the Deployment Preference Type to Linear10PercentEvery10Minutes. Set PreTraffic and Post Traffic properties to the Lambda alias.

Answer: A

Explanation:

The AWS Serverless Application Model (AWS SAM) comes built-in with CodeDeploy to provide gradual AWS Lambda deployments! The Deploymentpreference property in AWS SAM allows you to specify the type of deployment that you want. The Canary10Percent10Minutes option means that 10 percent of your customer traffic is immediately shifted to your new version. After 10 minutes, all traffic is shifted to the new version! The AutoPublishAlias property in AWS SAM allows AWS SAM to automatically create an alias that points to the updated version of the Lambda function! Therefore, option A is correct.

Question: 144

A company developed an API application on AWS by using Amazon CloudFront, Amazon API Gateway, and AWS Lambda. The API has a

minimum of four requests every second. A developer notices that many API users run the same query by using the POST method. The developer

wants to cache the POST request to optimize the API resources.

Which solution will meet these requirements?

A. Configure the CloudFront cache. Update the application to return cached content based upon the default request headers.

B. Override the cache method in the selected stage of API Gateway. Select the POST method.

C. Save the latest request response in Lambda /tmp directory. Update the Lambda function to check the /tmp directory.

D. Save the latest request in AWS Systems Manager Parameter Store. Modify the Lambda function to take the latest request response from Parameter Store.

Answer: B

Explanation:

Amazon API Gateway provides tools for creating and documenting web APIs that route HTTP requests to Lambda functions2. You can secure access to your API with authentication and authorization controls. Your APIs can serve traffic over the internet or can be accessible only within your VPC2. You can override the cache method in the selected stage of API Gateway2. Therefore, option B is correct.

Question: 145

A company is building a compute-intensive application that will run on a fleet of Amazon EC2 instances. The application uses attached Amazon

Elastic Block Store (Amazon EBS) volumes for storing data. The Amazon EBS volumes will be created at time of initial deployment. The

application will process sensitive information. All of the data must be encrypted. The solution should not impact the application's performance.

Which solution will meet these requirements?

A. Configure the fleet of EC2 instances to use encrypted EBS volumes to store data.

B. Configure the application to write all data to an encrypted Amazon S3 bucket.

C. Configure a custom encryption algorithm for the application that will encrypt and decrypt all data.

D. Configure an Amazon Machine Image (AMI) that has an encrypted root volume and store the data to ephemeral disks.

Answer: A

Explanation:

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances!. Amazon EBS encryption offers a straight-forward encryption solution for your EBS resources associated with your EC2 instances!. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted: Data at rest inside the volume, all data moving between the volume and the instance, all snapshots created from the volume, and all volumes created from those snapshots!. Therefore, option A is correct.

Question: 146

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda. The development team tests the API and validates responses for the known use cases before deploying the API to the production environment.

The developer wants to make the REST API available for testing by using API Gateway locally.

Which AWS Serverless Application Model Command Line Interface (AWS SAM CLI) subcommand will meet these requirements?

A. Sam local invoke

B. Sam local generate-event

C. Sam local start-lambda

D. Sam local start-api

Answer: D

Explanation:

The AWS Serverless Application Model Command Line Interface (AWS SAM CLI) is a command-line tool for local development and testing of Serverless applications2. The sam local start- api subcommand of AWS SAM CLI is used to simulate a REST API by starting a new local endpoint3. Therefore, option D is correct.

Question: 147

A developer is creating an AWS Lambda function that consumes messages from an Amazon Simple Queue Service (Amazon SQS) standard queue. The developer notices that the Lambda function processes some messages multiple times.

How should developer resolve this issue MOST cost-effectively?

A. Change the Amazon SQS standard queue to an Amazon SQS FIFO queue by using the Amazon SQS message deduplication ID.

B. Set up a dead-letter queue.

C. Set the maximum concurrency limit of the AWS Lambda function to 1

D. Change the message processing to use Amazon Kinesis Data Streams instead of Amazon SQS.

Answer: A

Explanation:

Amazon Simple Queue Service (Amazon SQS) is a fully managed queue service that allows you to decouple and scale for applications! Amazon SQS offers two types of queues: Standard and FIFO (First In First Out) queues! The FIFO queue uses the messageDeduplicationId property to treat messages with the same value as duplicate2. Therefore, changing the Amazon SQS standard queue to an Amazon SQS FIFO queue using the Amazon SQS message deduplication ID can help resolve the issue of the Lambda function processing some messages multiple times. Therefore, option A is correct.

Question: 148

A developer has observed an increase in bugs in the AWS Lambda functions that a development team has deployed in its Node.js application.

To minimize these bugs, the developer wants to implement automated testing of Lambda functions in an environment that closely simulates the Lambda environment.

The developer needs to give other developers the ability to run the tests locally. The developer also needs to integrate the tests into the team's continuous integration and continuous delivery (CI/CD) pipeline before the AWS Cloud Development Kit (AWS CDK) deployment.

Which solution will meet these requirements?

A. Create sample events based on the Lambda documentation. Create automated test scripts that use the cdk local invoke command to invoke the Lambda functions. Check the response. Document the test scripts for the other developers on the team. Update the CI/CD pipeline to run the test scripts.

B. Install a unit testing framework that reproduces the Lambda execution environment. Create sample events based on the Lambda documentation. Invoke the handler function by using a unit testing framework. Check the response. Document how to run the unit testing framework for the other developers on the team. Update the CI/CD pipeline to run the unit testing framework.

C. Install the AWS Serverless Application Model (AWS SAM) CLI tool. Use the sam local generateevent command to generate sample events for the automated tests. Create automated test scripts that use the sam local invoke command to invoke the Lambda functions. Check the response. Document the test scripts for the other developers on the team. Update the CI/CD pipeline to run the test scripts.

D. Create sample events based on the Lambda documentation. Create a Docker container from the Node.js base image to invoke the Lambda functions. Check the response. Document how to run the

Docker container for the other developers on the team. Update the CllCD pipeline to run the Docker container.

Answer: C

Explanation:

The AWS Serverless Application Model Command Line Interface (AWS SAM CLI) is a command-line tool for local development and testing of Serverless applications3. The sam local generateevent command of AWS SAM CLI generates sample events for automated tests3. The sam local invoke command is used to invoke Lambda functions3. Therefore, option C is correct.

Question: 149

A developer wants to add request validation to a production environment Amazon API Gateway API. The developer needs to test the changes

before the API is deployed to the production environment. For the test, the developer will send test requests to the API through a testing tool.

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

A. Export the existing API to an OpenAPI file. Create a new API. Import the OpenAPI file. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.

B. Modify the existing API to add request validation. Deploy the updated API to a new API Gateway stage. Perform the tests. Deploy the updated API to the API Gateway production stage.

C. Create a new API. Add the necessary resources and methods, including new request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.

D. Clone the existing API. Modify the new API to add request validation. Perform the tests. Modify the existing API to add request validation. Deploy the existing API to production.

Answer: B

Explanation:

Amazon API Gateway allows you to create, deploy, and manage a RESTful API to expose backend HTTP endpoints, AWS Lambda functions, or other AWS services! You can use API Gateway to perform basic validation of an API request before proceeding with the integration request! When the validation fails, API Gateway immediately fails the request, returns a 400 error response to the caller, and publishes the validation results in CloudWatch Logs!

To test changes before deploying to a production environment, you can modify the existing API to add request validation and deploy the updated API to a new API Gateway stage! This allows you to perform tests without affecting the production environment. Once testing is complete and successful, you can then deploy the updated API to the API Gateway production stage!

This approach has the least operational overhead as it avoids unnecessary creation of new APIs or exporting and importing of APIs. It leverages the existing infrastructure and only requires changes in the configuration of the existing API1.

Question: 150

A company has an existing application that has hardcoded database credentials A developer needs to modify the existing application The application is deployed in two AWS Regions with an activepassive failover configuration to meet company's disaster recovery strategy

The developer needs a solution to store the credentials outside the code. The solution must comply With the company's disaster recovery strategy

Which solution Will meet these requirements in the MOST secure way?

A. Store the credentials in AWS Secrets Manager in the primary Region. Enable secret replication to the secondary Region Update the application to use the Amazon Resource Name (ARN) based on the Region.

B. Store credentials in AWS Systems Manager Parameter Store in the primary Region. Enable parameter replication to the secondary Region. Update the application to use the Amazon Resource Name (ARN) based on the Region.

C. Store credentials in a config file. Upload the config file to an S3 bucket in me primary Region. Enable Cross-Region Replication (CRR) to an S3 bucket in the secondary region. Update the application to access the config file from the S3 bucket based on the Region.

D. Store credentials in a config file. Upload the config file to an Amazon Elastic File System (Amazon EFS) file system. Update the application to use the Amazon EFS file system Regional endpoints to access the config file in the primary and secondary Regions.

Answer: A

Explanation:

AWS Secrets Manager is a service that allows you to store and manage secrets, such as database credentials, API keys, and passwords, in a secure and centralized way. It also provides features such as automatic secret rotation, auditing, and monitoring!. By using AWS Secrets Manager, you can avoid hardcoding credentials in your code, which is a bad security practice and makes it difficult to update them. You can also replicate your secrets to another Region, which is useful for disaster recovery purposes2. To access your secrets from your application, you can use the ARN of the secret, which is a unique identifier that includes the Region name. This way, your application can use the appropriate secret based on the Region where it is deployed3.

Reference:

AWS Secrets Manager

Replicating and sharing secrets

Using your own encryption keys

Question: 151

A developer is creating an AWS Lambda function that searches for items from an Amazon DynamoDB table that contains customer contact information- The DynamoDB table items have the customer's email_address as the partition key and additional properties such as customer_type, name, and job_tltle.

The Lambda function runs whenever a user types a new character into the customer_type text input The developer wants the search to return partial matches of all the email_address property of a particular customer_type The developer does not want to recreate the DynamoDB table.

What should the developer do to meet these requirements?

A. Add a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key Perform a query operation on the GSI by using the begvns_wth key condition expression With the emad_address property

B. Add a global secondary index (GSI) to the DynamoDB table With ernail_address as the partition key and customer_type as the sort key Perform a query operation on the GSI by using the begins_wtth key condition expression With the emal_address property.

C. Add a local secondary index (LSI) to the DynamoDB table With customer_type as the partition key and email_address as the sort key

Perform a query operation on the LSI by using the begins_wlth key condition expression With the email_address property

D. Add a local secondary Index (LSI) to the DynamoDB table With job_tltle as the partition key and emad_address as the sort key

Perform a query operation on the LSI by using the begins_wrth key condition expression With the email_address property

---

Answer: A

Explanation:

Understand the Problem: The existing DynamoDB table has email_address as the partition key. Searching by customer_type requires a different data access pattern. We need an efficient way to query for partial matches on email_address based on customer_type.

Why Global Secondary Index (GSI):

GSIs allow you to define a different partition key and sort key from the main table, enabling new query patterns.

In this case, having customer_type as the GSI's partition key lets you group all emails with the same customer type together.

Using email_address as the sort key allows ordering within each customer type, facilitating the partial matching.

Querying the GSI:

You'll perform a query operation on the GSI, not the original table.

Use the begins_with key condition expression on the GSI's sort key (email_address) to find partial matches as the user types in the customer_type field.

Reference:

DynamoDB Global Secondary
Indexes: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

DynamoDB Query Operation: [invalid URL removed]

Key Condition Expressions: [invalid URL removed]

Question: 152

A developer is deploying a company's application to Amazon EC2 instances The application generates gigabytes of data files each day The files are rarely accessed but the files must be available to the application's users within minutes of a request during the first year of storage The company must retain the files for 7 years.

How can the developer implement the application to meet these requirements MOST cost- effectively?

A. Store the files in an Amazon S3 bucket Use the S3 Glacier Instant Retrieval storage class Create an S3 Lifecycle policy to transition the

files to the S3 Glacier Deep Archive storage class after ! year

B. Store the files in an Amazon S3 bucket. Use the S3 Standard storage class. Create an S3 Lifecycle policy to transition the files to the S3 Glacier Flexible Retrieval storage class after ! year.

C. Store the files on an Amazon Elastic Block Store (Amazon EBS) volume Use Amazon Data Lifecycle Manager (Amazon DLM) to create snapshots of the EBS volumes and to store those snapshots in Amazon S3

D. Store the files on an Amazon Elastic File System (Amazon EFS) mount. Configure EFS lifecycle management to transition the files to the EFS Standard-Infrequent Access (Standard-IA) storage class after ! year.

____ Answer: A

Explanation:

Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds. With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard- Infrequent Access (S3 Standard-IA) storage class, when your data is accessed once per quarter.
https://aws.amazon.com/s3/storage-classes/glacier/instant-retrieval/

Understanding Storage Requirements:

Files are large and infrequently accessed, but need to be available within minutes when requested in the first year.

Long-term (7-year) retention is required.

Cost-effectiveness is a top priority.

Why S3 Glacier Instant Retrieval:

Matches the retrieval requirements (access within minutes).

More cost-effective than S3 Standard for infrequently accessed data.

Simpler to use than traditional Glacier where retrievals take hours.

Why S3 Glacier Deep Archive:

Most cost-effective S3 storage class for long term archival.

Meets the 7-year retention requirement.

S3 Lifecycle Policy:

Automate the transition from Glacier Instant Retrieval to Glacier Deep Archive after one year.

Optimize costs by matching storage classes to access patterns.

Reference:

Amazon S3 Storage Classes: https://aws.amazon.com/s3/storage-classes/

S3 Glacier Instant Retrieval: [invalid URL removed]

S3 Glacier Deep Archive: [invalid URL removed]

S3 Lifecycle Policies: https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle- mgmt.html

Question: 153

A developer is creating a serverless application that uses an AWS Lambda function The developer will use AWS CloudFormation to deploy the application The application will write logs to Amazon CloudWatch Logs The developer has created a log group in a CloudFormation template for the application to use The developer needs to modify the CloudFormation template to make the name of the log group available to the application at runtime

Which solution will meet this requirement?

A. Use the AWS:Include transform in CloudFormation to provide the log group's name to the application

B. Pass the log group's name to the application in the user data section of the CloudFormation template.

C. Use the CloudFormation template's Mappings section to specify the log group's name for the application.

D. Pass the log group's Amazon Resource Name (ARN) as an environment variable to the Lambda function

Answer: D

Explanation:

CloudFormation and Lambda Environment Variables:

CloudFormation is an excellent tool to manage infrastructure as code, including the log group resource.

Lambda functions can access environment variables at runtime, making them a suitable way to pass configuration information like the log group ARN.

CloudFormation Template Modification:

In your CloudFormation template, define the log group resource.

In the Lambda function resource, add an Environment section:

YAML

Environment:

Variables:

LOG_GROUP_ARN: !Ref LogGroupResourceName

Use code with caution.

content_copy

The !Ref intrinsic function retrieves the log group's ARN, which CloudFormation generates during stack creation.

Using the ARN in Your Lambda Function:

Within your Lambda code, access the LOG_GROUP_ARN environment variable.

Configure your logging library (e.g., Python's logging module) to send logs to the specified log group.

Reference:

AWS Lambda Environment Variables: https://docs.aws.amazon.com/lambda/latest/dg/configuration-envvars.html

CloudFormation !Ref Intrinsic Function: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/intrinsic-function- reference-ref.html

---

Question: 154

A company has a web application that runs on Amazon EC2 instances with a custom Amazon Machine Image (AMI) The company uses AWS CloudFormation to provision the application The application runs in the us-east-1 Region, and the company needs to deploy the application to the us- west-1 Region

An attempt to create the AWS CloudFormation stack in us-west-1 fails. An error message states that the AMI ID does not exist. A developer must resolve this error with a solution that uses the least amount of operational overhead

Which solution meets these requirements?

A. Change the AWS CloudFormation templates for us-east-1 and us-west-1 to use an AWS AMI. Relaunch the stack for both Regions.

B. Copy the custom AMI from us-east-1 to us-west-1. Update the AWS CloudFormation template for us-west-1 to refer to AMI ID for the copied AMI Relaunch the stack

C. Build the custom AMI in us-west-1 Create a new AWS CloudFormation template to launch the stack in us-west-1 with the new AMI ID

D. Manually deploy the application outside AWS CloudFormation in us-west-1.

Answer: B

Explanation:

Problem: CloudFormation can't find the custom AMI in the target region (us-west-1) because AMIs are region-specific.

Copying AMIs:

AMIs can be copied across regions, maintaining their configuration.

This approach minimizes operational overhead as the existing CloudFormation template can be reused with a minor update.

Updating the Template:

Modify the CloudFormation template in us-west-1 to reference the newly copied AMI's ID in that region.

Reference:

Copying AMIs: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/CopyingAMIs.html

CloudFormation Templates and AMIs: [invalid URL removed]

Question: 155

A developer is working on a web application that uses Amazon DynamoDB as its data store The application has two DynamoDB tables one table that is named artists and one table that is named songs The artists table has artistName as the partition key. The songs table has songName as the partition key and artistName as the sort key

The table usage patterns include the retrieval of multiple songs and artists in a single database operation from the webpage. The developer needs a way to retrieve this information with minimal network traffic and optimal application performance.

Which solution will meet these requirements'?

A. Perform a BatchGetItem operation that returns items from the two tables. Use the list of songName artistName keys for the songs table and the list of artistName key for the artists table.

B. Create a local secondary index (LSI) on the songs table that uses artistName as the partition key Perform a query operation for each artistName on the songs table that filters by the list of songName Perform a query operation for each artistName on the artists table

C. Perform a BatchGetItem operation on the songs table that uses the songName/artistName keys. Perform a BatchGetItem operation on the artists table that uses artistName as the key.

D. Perform a Scan operation on each table that filters by the list of songName/artistName for the songs table and the list of artistName in the artists table.

Answer: A

Explanation:

Scenario: Application needs to fetch songs and artists efficiently in a single operation.

BatchGetItem: This DynamoDB operation retrieves multiple items across different tables based on their primary keys in a single request.

Optimized for Request Batching: This approach reduces network traffic compared to performing multiple queries individually.

Data Modeling: The songs table is designed appropriately for this access pattern using artistName as the sort key.

Reference:

Amazon DynamoDB

BatchGetItem: https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API BatchGetI tem.ht

Question: 156

A data visualization company wants to strengthen the security of its core applications The applications are deployed on AWS across its development staging, pre-production, and production environments. The company needs to encrypt all of its stored sensitive credentials The sensitive credentials need to be automatically rotated Aversion of the sensitive credentials need to be stored for each environment

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

A. Configure AWS Secrets Manager versions to store different copies of the same credentials across multiple environments

B. Create a new parameter version in AWS Systems Manager Parameter Store for each environment Store the environment-specific credentials in the parameter version.

C. Configure the environment variables in the application code Use different names for each environment type

D. Configure AWS Secrets Manager to create a new secret for each environment type. Store the environment-specific credentials in the secret

Answer: D

Explanation:

Secrets Management: AWS Secrets Manager is designed specifically for storing and managing sensitive credentials.

Environment Isolation: Creating separate secrets for each environment (development, staging, etc.) ensures clear separation and prevents accidental leaks.

Automatic Rotation: Secrets Manager provides built-in rotation capabilities, enhancing security posture.

Versioning: Tracking changes to secrets is essential for auditing and compliance.

Reference:

AWS Secrets Manager: https://aws.amazon.com/secrets-manager/

Secrets Manager

Rotation: https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

Question: 157

A company's developer has deployed an application in AWS by using AWS CloudFormation The CloudFormation stack includes parameters in AWS Systems Manager Parameter Store that the application uses as configuration settings. The application can modify the parameter values

When the developer updated the stack to create additional resources with tags, the developer noted that the parameter values were reset and that the values ignored the latest changes made by the application. The developer needs to change the way the company deploys the CloudFormation stack. The developer also needs to avoid resetting the parameter values outside the stack.

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

A. Modify the CloudFormation stack to set the deletion policy to Retain for the Parameter Store parameters.

B. Create an Amazon DynamoDB table as a resource in the CloudFormation stack to hold configuration data for the application Migrate the parameters that the application is modifying from Parameter Store to the DynamoDB table

C. Create an Amazon RDS DB instance as a resource in the CloudFormation stack. Create a table in the database for parameter configuration. Migrate the parameters that the application is modifying from Parameter Store to the configuration table

D. Modify the CloudFormation stack policy to deny updates on Parameter Store parameters

Answer: A

Explanation:

Problem: CloudFormation updates reset Parameter Store parameters, disrupting application behavior.

Deletion Policy: CloudFormation has a deletion policy that controls resource behavior when a stack is deleted or updated. The 'Retain' policy instructs CloudFormation to preserve a resource's current state.

Least Development Effort: This solution involves a simple CloudFormation template modification, requiring minimal code changes.

Reference:

CloudFormation Deletion Policies: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-attribute- deletionpolicy.html

Question: 158

A company has built an AWS Lambda function to convert large image files into output files that can be used in a third-party viewer application The company recently added a new module to the function to improve the output of the generated files However, the new module has increased the bundle size and has increased the time that is needed to deploy changes to the function code.

How can a developer increase the speed of the Lambda function deployment?

A. Use AWS CodeDeploy to deploy the function code

B. Use Lambda layers to package and load dependencies.

C. Increase the memory size of the function.

D. Use Amazon S3 to host the function dependencies

Answer: B

Explanation:

Problem: Large bundle size increases Lambda deployment time.

Lambda Layers: Layers let you package dependencies separately from your function code. This optimizes the deployment package, making updates faster.

Modularization: Breaking down dependencies into layers improves code organization and reusability.

Reference:

AWS Lambda Layers: https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html

Question: 159

A developer creates a static website for their department The developer deploys the static assets for the website to an Amazon S3 bucket and serves the assets with Amazon CloudFront The developer uses origin access control (OAC) on the CloudFront distribution to access the S3 bucket

The developer notices users can access the root URL and specific pages but cannot access directories without specifying a file name. For example, /products/index.html works, but /products returns an error The developer needs to enable accessing directories without specifying a file name without exposing the S3 bucket publicly.

Which solution will meet these requirements'?

A. Update the CloudFront distribution's settings to index.html as the default root object is set

B. Update the Amazon S3 bucket settings and enable static website hosting. Specify index html as the Index document Update the S3 bucket policy to enable access. Update the CloudFront distribution's origin to use the S3 website endpoint

C. Create a CloudFront function that examines the request URL and appends index.html when directories are being accessed Add the function as a viewer request CloudFront function to the CloudFront distribution's behavior.

D. Create a custom error response on the CloudFront distribution with the HTTP error code set to the HTTP 404 Not Found response code and the response page path to /index html Set the HTTP response code to the HTTP 200 OK response code

Answer: B

Explanation:

Problem: Directory access without file names fails.

S3 Static Website Hosting:

Configuring S3 as a static website enables automatic serving of index.html for directory requests.

Bucket policies ensure correct access permissions.

Updating the CloudFront origin simplifies routing.

Avoiding Public Exposure: The S3 website endpoint allows CloudFront to access content without making the bucket public.

Reference:

S3 Static Website Hosting: https://docs.aws.amazon.com/AmazonS3/latest/userguide/WebsiteHosting.html

Question: 160

A company needs to deploy all its cloud resources by using AWS CloudFormation templates A developer must create an Amazon Simple Notification Service (Amazon SNS) automatic notification to help enforce this rule. The developer creates an SNS topic and subscribes the email address of the company's security team to the SNS topic.

The security team must receive a notification immediately if an lAM role is created without the use of CloudFormation.

Which solution will meet this requirement?

A. Create an AWS Lambda function to filter events from CloudTrail if a role was created without CloudFormation Configure the Lambda function to publish to the SNS topic. Create an Amazon EventBridge schedule to invoke the Lambda function every l5 minutes

B. Create an AWS Fargate task in Amazon Elastic Container Service (Amazon ECS) to filter events from CloudTrail if a role was created without CloudFormation Configure the Fargate task to publish to the SNS topic Create an Amazon EventBridge schedule to run the Fargate task every l5 minutes

C. Launch an Amazon EC2 instance that includes a script to filter events from CloudTrail if a role was created without CloudFormation. Configure the script to publish to the SNS topic. Create a cron job to run the script on the EC2 instance every l5 minutes.

D. Create an Amazon EventBridge rule to filter events from CloudTrail if a role was created without CloudFormation Specify the SNS topic as the target of the EventBridge rule.

Answer: D

Explanation:

EventBridge (formerly CloudWatch Events) is the ideal service for real-time event monitoring.

CloudTrail logs IAM role creation.

EventBridge rules can filter CloudTrail events and trigger SNS notifications instantly.

Question: 161

A developer is investigating an issue in part of a company's application. In the application messages are sent to an Amazon Simple Queue Service (Amazon SQS) queue The AWS Lambda function polls messages from the SQS queue and sends email messages by using Amazon Simple Email Service (Amazon SES) Users have been receiving duplicate email messages during periods of high traffic.

Which reasons could explain the duplicate email messages? (Select TWO.)

A. Standard SQS queues support at-least-once message delivery

B. Standard SQS queues support exactly-once processing, so the duplicate email messages are because of user error.

C. Amazon SES has the DomainKeys Identified Mail (DKIM) authentication incorrectly configured

D. The SQS queue's visibility timeout is lower than or the same as the Lambda function's timeout.

E. The Amazon SES bounce rate metric is too high.

Answer: A

Explanation:

SQS Delivery Behavior: Standard SQS queues guarantee at-least-once delivery, meaning messages may be processed more than once. This can lead to duplicate emails in this scenario.

Visibility Timeout: If the visibility timeout on the SQS queue is too short, a message might become visible for another consumer before the first Lambda function finishes processing it. This can also lead to duplicates.

Reference:

Amazon SQS Delivery Semantics: [invalid URL removed]

Amazon SQS Visibility

Timeout: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs- visibility-timeout.html

Question: 162

A developer uses AWS CloudFormation to deploy an Amazon API Gateway API and an AWS Step Functions state machine The state machine must reference the API Gateway API after the CloudFormation template is deployed The developer needs a solution that uses the state machine to reference the API Gateway endpoint.

Which solution will meet these requirements MOST cost-effectively?

A. Configure the CloudFormation template to reference the API endpoint in the DefinitionSubstitutions property for the AWS StepFunctions StateMachme resource.

B. Configure the CloudFormation template to store the API endpoint in an environment variable for the AWS::StepFunctions::StateMachine resourc Configure the state machine to reference the environment variable

C. Configure the CloudFormation template to store the API endpoint in a standard AWS: SecretsManager Secret resource Configure the state machine to reference the resource

D. Configure the CloudFormation template to store the API endpoint in a standard AWS::AppConfig;:ConfigurationProfile resource Configure the state machine to reference the resource.

Answer: A

Explanation:

CloudFormation and Dynamic Reference: The DefinitionSubstitutions property in CloudFormation allows you to pass values into Step Functions state machines at runtime.

Cost-Effectiveness: This solution is cost-effective as it leverages CloudFormation's built-in capabilities, avoiding the need for additional services like Secrets Manager or AppConfig.

Reference:

AWS Step Functions State

Machine: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource- stepfunctions-statemachine.html

CloudFormation DefinitionSubstitutions: https://github.com/aws-cloudformation/aws- cloudformation-resource-providers-stepfunctions/issues/14

Question: 163

A developer created an AWS Lambda function that performs a series of operations that involve multiple AWS services. The function's duration time is higher than normal. To determine the cause of the issue, the developer must investigate traffic between the services without changing the function code

Which solution will meet these requirements?

A. Enable AWS X-Ray active tracing in the Lambda function Review the logs in X-Ray

B. Configure AWS CloudTrail View the trail logs that are associated with the Lambda function.

C. Review the AWS Config logs in Amazon Cloud Watch.

D. Review the Amazon CloudWatch logs that are associated with the Lambda function.

Answer: A

Explanation:

Tracing Distributed Systems: AWS X-Ray is designed to trace requests across services, helping identify bottlenecks in distributed applications like this one.

No Code Changes: Enabling X-Ray tracing often requires minimal code changes, meeting the requirement.

Identifying Bottlenecks: Analyzing X-Ray traces and logs will reveal latency in communications between different AWS services, leading to the high duration time.

Reference:

AWS X-Ray: https://aws.amazon.com/xray/

X- Ray and Lambda: https://docs.aws.amazon.com/xray/latest/devguide/xray-services-lambda.html

Question: 164

A developer designed an application on an Amazon EC2 instance The application makes API requests to objects in an Amazon S3 bucket

Which combination of steps will ensure that the application makes the API requests in the MOST secure manner? (Select TWO.)

A. Create an IAM user that has permissions to the S3 bucket. Add the user to an 1AM group

B. Create an IAM role that has permissions to the S3 bucket

C. Add the IAM role to an instance profile. Attach the instance profile to the EC2 instance.

D. Create an 1AM role that has permissions to the S3 bucket Assign the role to an 1AM group

E. Store the credentials of the IAM user in the environment variables on the EC2 instance

Answer: BC

Explanation:

IAM Roles for EC2: IAM roles are the recommended way to provide AWS credentials to applications running on EC2 instances.

Here's how this works:

You create an IAM role with the necessary permissions to access the target S3 bucket.

You create an instance profile and associate the IAM role with this profile.

When launching the EC2 instance, you attach this instance profile.

Temporary Security Credentials: When the application on the EC2 instance needs to access S3, it doesn't directly use access keys. Instead, the AWS SDK running on the instance retrieves temporary security credentials associated with the role. These are rotated automatically by AWS.

Reference:

IAM Roles for Amazon

EC2: https://docs.aws.amazon.com/IAM/latest/UserGuide/id roles use switch-role-ec2.html

Temporary Security

Credentials: https://docs.aws.amazon.com/IAM/latest/UserGuide/id credentials temp.html

Question: 165 _____

A developer is working on an ecommerce website The developer wants to review server logs without logging in to each of the application servers individually. The website runs on multiple Amazon EC2 instances, is written in Python, and needs to be highly available

How can the developer update the application to meet these requirements with MINIMUM changes?

A. Rewrite the application to be cloud native and to run on AWS Lambda, where the logs can be reviewed in Amazon CloudWatch

B. Set up centralized logging by using Amazon OpenSearch Service, Logstash, and OpenSearch Dashboards

C. Scale down the application to one larger EC2 instance where only one instance is recording logs

D. Install the unified Amazon CloudWatch agent on the EC2 instances Configure the agent to push the application logs to CloudWatch

---

_____ Answer: D

Explanation:

Centralized Logging Benefits: Centralized logging is essential for operational visibility in scalable systems, especially those using multiple EC2 instances like our e-commerce website. CloudWatch provides this capability, along with other monitoring features.

CloudWatch Agent: This is the best way to send custom application logs from EC2 instances to CloudWatch. Here's the process:

Install the CloudWatch agent on each EC2 instance.

Configure the agent with a configuration file, specifying:

Which log files to collect.

The format in which to send logs to CloudWatch (e.g., JSON).

The specific CloudWatch Logs log group and log stream for these logs.

Viewing and Analyzing Logs: Once the agent is pushing logs, use the CloudWatch Logs console or API:

View and search the logs across all instances.

Set up alarms based on log events.

Use CloudWatch Logs Insights for sophisticated queries and analysis.

Reference:

Amazon CloudWatch Logs: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/WhatIsCloudWatchLogs.html

Unified CloudWatch Agent: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AgentReference.html

CloudWatch Logs Insights: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/AnalyzingLogData.html

Question: 166

A company runs a batch processing application by using AWS Lambda functions and Amazon API Gateway APIs with deployment stages for development, user acceptance testing and production A development team needs to configure the APIs in the deployment stages to connect to third-party service endpoints.

Which solution will meet this requirement?

A. Store the third-party service endpoints in Lambda layers that correspond to the stage

B. Store the third-party service endpoints in API Gateway stage variables that correspond to the stage

C. Encode the third-party service endpoints as query parameters in the API Gateway request URL.

D. Store the third-party service endpoint for each environment in AWS AppConfig

---

Answer: B

Explanation:

API Gateway Stage Variables: These are designed for configuring dynamic values for your APIs in different deployment stages (dev, test, prod). Here's how to use them for third-party endpoints:

In the API Gateway console, access the "Stages" section of your API.

For each stage, create a stage variable named something like thirdPartyEndpoint.

Set the value of this variable to the actual endpoint URL for that specific environment.

When configuring API requests within your API Gateway method, reference this endpoint using ${stageVariables.thirdPartyEndpoint}.

Why Stage Variables Excel Here:

Environment Isolation: This approach keeps the endpoint configuration specific to each deployment stage, ensuring the right endpoints are used during development, testing, and production cycles.

Ease of Management: You manage the endpoints directly through the API Gateway console without additional infrastructure.

Reference:

Amazon API Gateway Stage Variables: https://docs.aws.amazon.com/apigateway/latest/developerguide/stage-variables.html

Question: 167

A company is creating an application that processes csv files from Amazon S3 A developer has created an S3 bucket The developer has also created an AWS Lambda function to process the csv files from the S3 bucket

Which combination of steps will invoke the Lambda function when a csv file is uploaded to Amazon S3? (Select TWO.)

A. Create an Amazon EventBridge rule Configure the rule with a pattern to match the S3 object created event

B. Schedule an Amazon EventBridge rule to run a new Lambda function to scan the S3 bucket.

C. Add a trigger to the existing Lambda function. Set the trigger type to EventBridge Select the Amazon EventBridge rule.

D. Create a new Lambda function to scan the S3 bucket for recently added S3 objects

E. Add S3 Lifecycle rules to invoke the existing Lambda function

Answer: AE

Explanation:

Amazon EventBridge: A service that reacts to events from various AWS sources, including S3. Rules define which events trigger actions (like invoking Lambda functions).

S3 Object Created Events: EventBridge can detect these, providing seamless integration for automated CSV processing.

S3 Lifecycle Rules: Allow for actions based on object age or prefixes. These can directly trigger Lambda functions for file processing.

Reference:

Amazon EventBridge Documentation: https://docs.aws.amazon.com/eventbridge/

Working with S3 Event Notifications: https://docs.aws.amazon.com/AmazonS3/latest/userguide/EventNotifications.html

S3 Lifecycle Configuration: https://docs.aws.amazon.com/AmazonS3/latest/userguide/object- lifecycle-mgmt.html

Question: 168

A developer is creating an AWS Lambda function in VPC mode An Amazon S3 event will invoke the Lambda function when an object is

uploaded into an S3 bucket The Lambda function will process the object and produce some analytic results that will be recorded into a file Each processed object will also generate a log entry that will be recorded into a file.

Other Lambda functions. AWS services, and on-premises resources must have access to the result files and log file. Each log entry must also be appended to the same shared log file. The developer needs a solution that can share files and append results into an existing file.

Which solution should the developer use to meet these requirements?

A. Create an Amazon Elastic File System (Amazon EFS) file system. Mount the EFS file system in Lambda. Store the result files and log file in the mount point. Append the log entries to the log file.

B. Create an Amazon Elastic Block Store (Amazon EBS) Multi-Attach enabled volume Attach the EBS volume to all Lambda functions. Update the Lambda function code to download the log file, append the log entries, and upload the modified log file to Amazon EBS

C. Create a reference to the /tmp local directory. Store the result files and log file by using the directory reference. Append the log entry to the log file.

D. Create a reference to the /opt storage directory Store the result files and log file by using the directory reference Append the log entry to the log file

---

_____ Answer: A

Explanation:

Amazon EFS: A network file system (NFS) providing shared, scalable storage across multiple Lambda functions and other AWS resources.

Lambda Mounting: EFS file systems can be mounted within Lambda functions to access a shared storage space.

Log Appending: EFS supports appending data to existing files, making it ideal for the log file scenario.

Reference:

Amazon EFS Documentation: https://docs.aws.amazon.com/efs/

Using Amazon EFS with AWS Lambda: https://docs.aws.amazon.com/lambda/latest/dg/services-efs.html

---

Question: 169

A company hosts its application on AWS. The application runs on an Amazon Elastic Container Service (Amazon ECS) cluster that uses AWS Fargate. The cluster runs behind an Application Load Balancer The application stores data in an Amazon Aurora database A developer encrypts and manages database credentials inside the application

The company wants to use a more secure credential storage method and implement periodic credential rotation.

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

A. Migrate the secret credentials to Amazon RDS parameter groups. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key Turn on secret rotation. Use 1AM policies and roles to grant AWS KMS permissions to access Amazon RDS.

B. Migrate the credentials to AWS Systems Manager Parameter Store. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use 1AM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager

C. Migrate the credentials to ECS Fargate environment variables. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key Turn on secret rotation. Use 1AM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.

D. Migrate the credentials to AWS Secrets Manager. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key Turn on secret rotation Use 1AM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager by using keys.

Answer: D

Explanation:

Secrets Management: AWS Secrets Manager is designed specifically for storing and managing sensitive credentials.

Built-in Rotation: Secrets Manager provides automatic secret rotation functionality, enhancing security posture significantly.

IAM Integration: IAM policies and roles grant fine-grained access to ECS Fargate, ensuring the principle of least privilege.

Reduced Overhead: This solution centralizes secrets management and automates rotation, reducing operational overhead compared to the other options.

Reference:

AWS Secrets Manager: https://aws.amazon.com/secrets-manager/

Secrets Manager

Rotation: https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

IAM for Secrets Manager: https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth- and-access iam-policies.html

Question: 170

A developer is testing a RESTful application that is deployed by using Amazon API Gateway and AWS Lambda When the developer tests the user login by using credentials that are not valid, the developer receives an HTTP 405 METHOD_NOT_ALLOWED error The developer has verified that the test is sending the correct request for the resource

Which HTTP error should the application return in response to the request?

A. HTTP 401

B. HTTP 404

C. HTTP 503

D. HTTP 505

Answer: A

Explanation:

HTTP Status Codes: Each HTTP status code has a specific meaning in RESTful APIs.

HTTP 405 (Method Not Allowed): Indicates that the request method (e.g., POST) is not supported for the specified resource.

HTTP 401 (Unauthorized): Represents a failure to authenticate, which is the appropriate response for invalid login credentials.

Reference:

HTTP Status Codes: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status

Question: 171

A company runs an application on AWS The application uses an AWS Lambda function that is configured with an Amazon Simple Queue Service (Amazon SQS) queue called high priority queue as the event source A developer is updating the Lambda function with another SQS queue called low priority queue as the event source The Lambda function must always read up to 10 simultaneous messages from the high priority queue before processing messages from low priority queue. The Lambda function must be limited to 100 simultaneous invocations.

Which solution will meet these requirements'?

A. Set the event source mapping batch size to 10 for the high priority queue and to 90 for the low priority queue

B. Set the delivery delay to 0 seconds for the high priority queue and to 10 seconds for the low priority queue

C. Set the event source mapping maximum concurrency to 10 for the high priority queue and to 90 for the low priority queue

D. Set the event source mapping batch window to 10 for the high priority queue and to 90 for the low priority queue

Answer: C

Explanation:

Lambda Concurrency: The 'maximum concurrency' setting in event source mappings controls the maximum number of simultaneous invocations Lambda allows for that specific source.

Prioritizing Queues: Setting a lower maximum concurrency for the 'high priority queue' ensures it's processed first while allowing more concurrent invocations from the 'low priority queue'.

Batching: Batch size settings affect the number of messages Lambda retrieves from a queue per invocation, which is less relevant to the prioritization requirement.

Reference:

Lambda Event Source Mappings: https://docs.aws.amazon.com/lambda/latest/dg/invocation- eventsourcemapping.html

Lambda Concurrency: https://docs.aws.amazon.com/lambda/latest/dg/configuration- concurrency.html

Question: 172

A developer deployed an application to an Amazon EC2 instance The application needs to know the public IPv4 address of the instance

How can the application find this information?

A. Query the instance metadata from http./M69.254.169.254. latestmeta-data/.

B. Query the instance user data from http '169 254.169 254. latest/user-data/

C. Query the Amazon Machine Image (AMI) information from http://169.254.169.254/latest/meta- data/ami/.

D. Check the hosts file of the operating system

Answer: A

Explanation:

Instance Metadata Service: EC2 instances have access to an internal metadata service. It provides instance-specific information like instance ID, security groups, and public IP address.

Accessing Metadata:

Make an HTTP GET request to the base URL: http://169.254.169.254/latest/meta-data/

You'll get a list of available categories. The public IPv4 address is under public-ipv4.

Reference:

Instance Metadata and User Data: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html

Question: 173

A company has a web application that is hosted on Amazon EC2 instances The EC2 instances are configured to stream logs to Amazon CloudWatch Logs The company needs to receive an Amazon Simple Notification Service (Amazon SNS) notification when the number of application error messages exceeds a defined threshold within a 5-minute period

Which solution will meet these requirements?

A. Rewrite the application code to stream application logs to Amazon SNS Configure an SNS topic to send a notification when the

number of errors exceeds the defined threshold within a 5-minute period

B. Configure a subscription filter on the CloudWatch Logs log group. Configure the filter to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period.

D. Install and configure the Amazon Inspector agent on the EC2 instances to monitor for errors Configure Amazon Inspector to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period

D. Create a CloudWatch metric filter to match the application error pattern in the log data. Set up a CloudWatch alarm based on the new custom metric. Configure the alarm to send an SNS notification when the number of errors exceeds the defined threshold within a 5-minute period.

Answer: D

Explanation:

CloudWatch for Log Analysis: CloudWatch is the best fit here because logs are already centralized. Here's the process:

Metric Filter: Create a metric filter on the CloudWatch Logs log group. Design a pattern to specifically identify application error messages.

Custom Metric: This filter generates a new custom CloudWatch metric (e.g., ApplicationErrors). This metric tracks the error count.

CloudWatch Alarm: Create an alarm on the ApplicationErrors metric. Configure the alarm with your desired threshold and a 5-minute evaluation period.

SNS Action: Set the alarm to trigger an SNS notification when it enters the alarm state.

Reference:

CloudWatch Metric

Filters: https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/MonitoringLogData.html

CloudWatch

Alarms: https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html

Question: 174

A developer is creating a service that uses an Amazon S3 bucket for image uploads. The service will use an AWS Lambda function to create a thumbnail of each image Each time an image is uploaded the service needs to send an email notification and create the thumbnail The developer needs to configure the image processing and email notifications setup.

Which solution will meet these requirements?

A. Create an Amazon Simple Notification Service (Amazon SNS) topic Configure S3 event notifications with a destination of the SNS topic Subscribe the Lambda function to the SNS topic Create an email notification subscription to the SNS topic

B. Create an Amazon Simple Notification Service (Amazon SNS) topic. Configure S3 event notifications with a destination of the SNS

topic. Subscribe the Lambda function to the SNS topic. Create an Amazon Simple Queue Service (Amazon SQS) queue Subscribe the SQS queue to the SNS topic Create an email notification subscription to the SQS queue.

C. Create an Amazon Simple Queue Service (Amazon SQS) queue Configure S3 event notifications with a destination of the SQS queue Subscribe the Lambda function to the SQS queue Create an email notification subscription to the SQS queue.

D. Create an Amazon Simple Queue Service (Amazon SQS) queue. Send S3 event notifications to Amazon EventBridge. Create an EventBndge rule that runs the Lambda function when images are uploaded to the S3 bucket Create an EventBridge rule that sends notifications to the SQS queue Create an email notification subscription to the SQS queue

_____ Answer: A

Explanation:

SNS as a Fan-out Mechanism: SNS is perfect for triggering multiple actions from a single event (here, the image upload).

Workflow:

SNS Topic: Create an SNS topic that will be the central notification point.

S3 Event Notification: Configure the S3 bucket to send 'Object Created' event notifications to the SNS topic.

Lambda Subscription: Subscribe your thumbnail-creating Lambda function to the SNS topic.

Email Subscription: Subscribe an email address to the SNS topic to trigger notifications.

Reference:

S3 Event Notifications: https://docs.aws.amazon.com/AmazonS3/latest/userguide/EventNotifications.html

SNS Subscriptions: https://docs.aws.amazon.com/sns/latest/dg/SNSMobilePush.html

Question: 175

A developer is building a microservices-based application by using Python on AWS and several AWS services The developer must use AWS X-Ray The developer views the service map by using the console to view the service dependencies. During testing, the developer notices that some services are missing from the service map

What can the developer do to ensure that all services appear in the X-Ray service map?

A. Modify the X-Ray Python agent configuration in each service to increase the sampling rate

B. Instrument the application by using the X-Ray SDK for Python. Install the X-Ray SDK for all the services that the application uses

C. Enable X-Ray data aggregation in Amazon CloudWatch Logs for all the services that the application uses

D. Increase the X-Ray service map timeout value in the X-Ray console

Answer: B

Explanation:

AWS X-Ray SDK: The primary way to enable X-Ray tracing within applications. The SDK sends data about requests and subsegments to the X-Ray daemon for service map generation.

Instrumenting All Services: To visualize a complete microservice architecture on the service map, each relevant service must include the X-Ray SDK.

Reference:

AWS X-Ray Documentation: https://docs.aws.amazon.com/xray/

X- Ray SDK for Python: https://docs.aws.amazon.com/xray/latest/devguide/xray-sdk-python.html

Question: 176

A company has a social media application that receives large amounts of traffic User posts and interactions are continuously updated in an Amazon RDS database The data changes frequently, and the data types can be complex The application must serve read requests with minimal latency

The application's current architecture struggles to deliver these rapid data updates efficiently The company needs a solution to improve the application's performance.

Which solution will meet these requirements'?

A. Use Amazon DynamoDB Accelerator (DAX) in front of the RDS database to provide a caching layer for the high volume of rapidly changing data

B. Set up Amazon S3 Transfer Acceleration on the RDS database to enhance the speed of data transfer from the databases to the application.

C. Add an Amazon CloudFront distribution in front of the RDS database to provide a caching layer for the high volume of rapidly changing data

D. Create an Amazon ElastiCache for Redis cluster. Update the application code to use a write- through caching strategy and read the data from Redis.

Answer: D

Explanation:

Amazon ElastiCache for Redis: An in-memory data store known for extremely low latency, ideal for caching frequently accessed,

complex data.

Write-Through Caching: Ensures that data is always consistent between the cache and the database. Writes go to both Redis and RDS.

Performance Gains: Redis handles reads with minimal latency, ofloading the RDS database and improving the application's responsiveness.

Reference:

Amazon ElastiCache for Redis

Documentation: https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/

Caching Strategies: https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Strategies.html

Question: 177

A company runs a payment application on Amazon EC2 instances behind an Application Load Balance The EC2 instances run in an Auto Scaling group across multiple Availability Zones The application needs to retrieve application secrets during the application startup and export the secrets as environment variables These secrets must be encrypted at rest and need to be rotated every month.

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

A. Save the secrets in a text file and store the text file in Amazon S3 Provision a customer managed key Use the key for secret encryption in Amazon S3 Read the contents of the text file and read the export as environment variables Configure S3 Object Lambda to rotate the text file every month

B. Save the secrets as strings in AWS Systems Manager Parameter Store and use the default AWS Key Management Service (AWS KMS) key Configure an Amazon EC2 user data script to retrieve the secrets during the startup and export as environment variables Configure an AWS Lambda function to rotate the secrets in Parameter Store every month.

C. Save the secrets as base64 encoded environment variables in the application properties. Retrieve the secrets during the application startup. Reference the secrets in the application code. Write a script to rotate the secrets saved as environment variables.

D. Store the secrets in AWS Secrets Manager Provision a new customer master key Use the key to encrypt the secrets Enable automatic rotation Configure an Amazon EC2 user data script to programmatically retrieve the secrets during the startup and export as environment variables

_____ Answer: D

Explanation:

AWS Secrets Manager: Built for managing secrets, providing encryption, automatic rotation, and access control.

Customer Master Key (CMK): Provides an extra layer of control over encryption through AWS KMS.

Automatic Rotation: Enhances security by regularly changing the secret.

User Data Script: Allows secrets retrieval at instance startup and sets them as environment variables for seamless use within the application.

Reference:

AWS Secrets Manager Documentation: https://docs.aws.amazon.com/secretsmanager/

AWS KMS Documentation: https://docs.aws.amazon.com/kms/

User Data for EC2 Instances: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user- data.html

Question: 178

A company is using Amazon API Gateway to invoke a new AWS Lambda function The company has Lambda function versions in its PROD and DEV environments. In each environment, there is a Lambda function alias pointing to the corresponding Lambda function version API Gateway has one stage that is configured to point at the PROD alias

The company wants to configure API Gateway to enable the PROD and DEV Lambda function versions to be simultaneously and distinctly available

Which solution will meet these requirements?

A. Enable a Lambda authorizer for the Lambda function alias in API Gateway Republish PROD and create a new stage for DEV Create API Gateway stage variables for the PROD and DEV stages. Point each stage variable to the PROD Lambda authorizer to the DEV Lambda authorizer.

B. Set up a gateway response in API Gateway for the Lambda function alias. Republish PROD and create a new stage for DEV. Create gateway responses in API Gateway for PROD and DEV Lambda aliases

C. Use an environment variable for the Lambda function alias in API Gateway. Republish PROD and create a new stage for development. Create API gateway environment variables for PROD and DEV stages. Point each stage variable to the PROD Lambda function alias to the DEV Lambda function alias.

D. Use an API Gateway stage variable to configure the Lambda function alias Republish PROD and create a new stage for development Create API Gateway stage variables for PROD and DEV stages Point each stage variable to the PROD Lambda function alias and to the DEV Lambda function alias

Answer: D

Explanation:

API Gateway Stages: Stages in API Gateway represent distinct environments (like PROD and DEV) allowing different configurations.

Stage Variables: Stage variables store environment-specific information, including Lambda function aliases.

Ease of Management: This solution offers a straightforward way to manage different Lambda function versions across environments.

Reference:

API Gateway Stages: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up- stages.html

API Gateway Stage Variables: https://docs.aws.amazon.com/apigateway/latest/developerguide/stage-variables.html

Question: 179

A developer is working on an ecommerce platform that communicates with several third-party payment processing APIs The third-party payment services do not provide a test environment.

The developer needs to validate the ecommerce platform's integration with the third-party payment processing APIs. The developer must test the API integration code without invoking the third-party payment processing APIs.

Which solution will meet these requirements'?

A. Set up an Amazon API Gateway REST API with a gateway response configured for status code 200 Add response templates that contain sample responses captured from the real third-party API.

B. Set up an AWS AppSync GraphQL API with a data source configured for each third-party API Specify an integration type of Mock Configure integration responses by using sample responses captured from the real third-party API.

C. Create an AWS Lambda function for each third-party API. Embed responses captured from the real third-party API. Configure Amazon Route 53 Resolver with an inbound endpoint for each Lambda function's Amazon Resource Name (ARN).

D. Set up an Amazon API Gateway REST API for each third-party API Specify an integration request type of Mock Configure integration responses by using sample responses captured from the real third-party API

Answer: D

Explanation:

Mocking API Responses: API Gateway's Mock integration type enables simulating API behavior without invoking backend services.

Testing with Sample Data: Using captured responses from the real third-party API ensures realistic testing of the integration code.

Focus on Integration Logic: This solution allows the developer to isolate and test the application's interaction with the payment APIs, even without a test environment from the third-party providers.

Reference:

Amazon API Gateway Mock Integrations: https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock- integration.html

Question: 180

A developer is creating a simple proof-of-concept demo by using AWS CloudFormation and AWS Lambda functions The demo will use a CloudFormation template to deploy an existing Lambda function The Lambda function uses deployment packages and dependencies stored in Amazon S3 The developer defined anAWS Lambda Function resource in a CloudFormation template. The developer needs to add the S3 bucket to the CloudFormation template.

What should the developer do to meet these requirements with the LEAST development effort?

A. Add the function code in the CloudFormation template inline as the code property

B. Add the function code in the CloudFormation template as the ZipFile property.

C. Find the S3 key for the Lambda function Add the S3 key as the ZipFile property in the CloudFormation template.

D. Add the relevant key and bucket to the S3Bucket and S3Key properties in the CloudFormation template

Answer: D

Explanation:

S3Bucket and S3Key: These properties in a CloudFormation AWS::Lambda::Function resource specify the location of the function's code in S3.

Least Development Effort: This solution minimizes code changes, relying on CloudFormation to reference the existing S3 deployment package.

Reference:

AWS::Lambda::Function

Resource https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource- lambda-function.html

Question: 181

A company is planning to use AWS CodeDeploy to deploy an application to Amazon Elastic Container Service (Amazon ECS) During the deployment of a new version of the application, the company initially must expose only !0% of live traffic to the new version of the deployed application. Then, after !5 minutes elapse, the company must route all the remaining live traffic to the new version of the deployed application.

Which CodeDeploy predefined configuration will meet these requirements?

A. CodeDeployDefault ECSCanary!0Percent!5Minutes

B. CodeDeployDefault LambdaCanary!0Percent5Minutes

C. CodeDeployDefault LambdaCanary!0Percent!5Minutes

D. CodeDeployDefault ECSLinear!0PercentEvery! Minutes

Answer: A

Explanation:

CodeDeploy Predefined Configurations: CodeDeploy offers built-in deployment configurations for common scenarios.

Canary Deployment: Canary deployments gradually shift traffic to a new version, ideal for controlled rollouts like this requirement.

CodeDeployDefault.ECSCanary!0Percent!5Minutes: This configuration matches the company's requirements, shifting !0% of traffic initially and then completing the rollout after !5 minutes.

Reference:

AWS CodeDeploy Deployment

Configurations: https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment- configurations-create.html

Question: 182

A developer is using AWS Step Functions to automate a workflow The workflow defines each step as an AWS Lambda function task The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an UlegalArgumentException error or a TooManyRequestsException error

The developer wants the state machine to stop running when the state machine encounters an UlegalArgumentException error. The state machine needs to retry the GetResource task one additional time after l0 seconds if the state machine encounters a TooManyRequestsException error. If the second attempt fails, the developer wants the state machine to stop running.

How can the developer implement the Lambda retry functionality without adding unnecessary complexity to the state machine'?

A. Add a Delay task after the GetResource task. Add a catcher to the GetResource task. Configure the catcher with an error type of TooManyRequestsException. Configure the next step to be the Delay task Configure the Delay task to wait for an interval of l0 seconds Configure the next step to be the GetResource task.

B. Add a catcher to the GetResource task Configure the catcher with an error type of TooManyRequestsException. an interval of l0 seconds, and a maximum attempts value of l. Configure the next step to be the GetResource task.

C. Add a retrier to the GetResource task Configure the retrier with an error type of TooManyRequestsException, an interval of l0 seconds, and a maximum attempts value of l.

D. Duplicate the GetResource task Rename the new GetResource task to TryAgain Add a catcher to the original GetResource task Configure the catcher with an error type of TooManyRequestsException. Configure the next step to be TryAgain.

Answer: C

Explanation:

Step Functions Retriers: Retriers provide a built-in way to gracefully handle transient errors within State Machines. Here's how to use them:

Directly attach a retrier to the problematic 'GetResource' task.

Configure the retrier:

ErrorEquals: Set this to ['TooManyRequestsException'] to target the specific error.

IntervalSeconds: Set to l0 for the desired retry delay.

MaxAttempts: Set to l, as you want only one retry attempt.

Error Handling:

Upon 'TooManyRequestsException', the retrier triggers the task again after l0 seconds.

On a second failure, Step Functions moves to the next state or fails the workflow, as per your design.

'IllegalArgumentException' causes error propagation as intended.

Reference:

Error Handling in Step Functions: https://docs.aws.amazon.com/step-functions/latest/dg/concepts- error-handling.html

Question: 183

An Amazon Simple Queue Service (Amazon SQS) queue serves as an event source for an AWS Lambda function In the SQS queue, each item corresponds to a video file that the Lambda function must convert to a smaller resolution The Lambda function is timing out on longer video files, but the Lambda function's timeout is already configured to its maximum value

What should a developer do to avoid the timeouts without additional code changes'?

A. Increase the memory configuration of the Lambda function

B. Increase the visibility timeout on the SQS queue

C. Increase the instance size of the host that runs the Lambda function.

D. Use multi-threading for the conversion.

Answer: B

Explanation:

Visibility Timeout: When an SQS message is processed by a consumer (here, the Lambda function), it's temporarily hidden from other consumers. Visibility timeout controls this duration.

How It Helps:

Increase the visibility timeout beyond the maximum processing time your Lambda might typically take for long videos.

This prevents the message from reappearing in the queue while Lambda is still working, avoiding premature timeouts.

Reference:

SQS Visibility

Timeout: https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs- visibility-timeout.html

Question: 184

A developer is creating an Amazon DynamoDB table by using the AWS CLI The DynamoDB table must use server-side encryption with an AWS owned encryption key

How should the developer create the DynamoDB table to meet these requirements?

A. Create an AWS Key Management Service (AWS KMS) customer managed key. Provide the key's Amazon Resource Name (ARN) in the KMSMasterKeyId parameter during creation of the DynamoDB table

B. Create an AWS Key Management Service (AWS KMS) AWS managed key Provide the key's Amazon Resource Name (ARN) in the KMSMasterKeyId parameter during creation of the DynamoDB table

C. Create an AWS owned key Provide the key's Amazon Resource Name (ARN) in the KMSMasterKeyId parameter during creation of the DynamoDB table.

D. Create the DynamoDB table with the default encryption options

_____ Answer: D

Explanation:

Default SSE in DynamoDB: DynamoDB tables are encrypted at rest by default using an AWS owned key (SSE-S3).

No Additional Action Needed: Creating a table without explicitly specifying a KMS key will use this default encryption.

Reference:

DynamoDB Server-Side

Encryption: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Encryption

Question: 185

A developer is creating an AWS Lambda function. The Lambda function needs an external library to connect to a third-party solution The external library is a collection of files with a total size of 100 MB The developer needs to make the external library available to the Lambda execution environment and reduce the Lambda package space

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

A. Create a Lambda layer to store the external library Configure the Lambda function to use the layer

B. Create an Amazon S3 bucket Upload the external library into the S3 bucket. Mount the S3 bucket folder in the Lambda function Import the library by using the proper folder in the mount point.

C. Load the external library to the Lambda function's /tmp directory during deployment of the Lambda package. Import the library from the /tmp directory.

D. Create an Amazon Elastic File System (Amazon EFS) volume. Upload the external library to the EFS volume Mount the EFS volume in the Lambda function. Import the library by using the proper folder in the mount point.

_____ Answer: A

Explanation:

Lambda Layers: These are designed to package dependencies that you can share across functions.

How to Use:

Create a layer, upload your 100MB library as a zip.

Attach the layer to your function.

In your function code, import the library from the standard layer path.

Reference:

Lambda Layers: https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html

Question: 186

A company built an online event platform For each event the company organizes quizzes and generates leaderboards that are based on the quiz scores. The company stores the leaderboard data in Amazon DynamoDB and retains the data for 30 days after an event is complete The company then uses a scheduled job to delete the old leaderboard data

The DynamoDB table is configured with a fixed write capacity. During the months when many events occur, the DynamoDB write API requests are throttled when the scheduled delete job runs.

A developer must create a long-term solution that deletes the old leaderboard data and optimizes write throughput

Which solution meets these requirements?

A. Configure a TTL attribute for the leaderboard data

B. Use DynamoDB Streams to schedule and delete the leaderboard data

C. Use AWS Step Functions to schedule and delete the leaderboard data.

D. Set a higher write capacity when the scheduled delete job runs

Answer: A

Explanation:

DynamoDB TTL (Time-to-Live): A native feature that automatically deletes items after a specified expiration time.

Efficiency: Eliminates the need for scheduled deletion jobs, optimizing write throughput by avoiding potential throttling conflicts.

Seamless Integration: TTL works directly within DynamoDB, requiring minimal development overhead.

Reference:

DynamoDB TTL

Documentation: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html

Question: 187

A developer must use multi-factor authentication (MFA) to access data in an Amazon S3 bucket that is in another AWS account. Which AWS Security Token Service (AWS STS) API operation should the developer use with the MFA information to meet this requirement?

A. AssumeRoleWithWebidentity

B. GetFederationToken

C. AssumeRoleWithSAML

D. AssumeRole

Answer: D

Explanation:

AWS STS AssumeRole: The central operation for assuming temporary security credentials, commonly used for cross-account access.

MFA Integration: The AssumeRole call can include MFA information to enforce multi-factor authentication.

Credentials for S3 Access: The returned temporary credentials would provide the necessary permissions to access the S3 bucket in the other account.

Reference:

AWS STS AssumeRole

Documentation: https://docs.aws.amazon.com/STS/latest/APIReference/API AssumeRole.html

Question: 188

A company has an analytics application that uses an AWS Lambda function to process transaction data asynchronously A developer

notices that asynchronous invocations of the Lambda function sometimes fail When failed Lambda function invocations occur, the developer wants to invoke a second Lambda function to handle errors and log details.

Which solution will meet these requirements?

A. Configure a Lambda function destination with a failure condition Specify Lambda function as the destination type Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource

B. Enable AWS X-Ray active tracing on the initial Lambda function. Configure X-Ray to capture stack traces of the failed invocations. Invoke the error-handling Lambda function by including the stack traces in the event object.

C. Configure a Lambda function trigger with a failure condition Specify Lambda function as the destination type Specify the error-handling Lambda function's Amazon Resource Name (ARN) as the resource

D. Create a status check alarm on the initial Lambda function. Configure the alarm to invoke the error-handling Lambda function when the alarm is initiated. Ensure that the alarm passes the stack trace in the event object.

Answer: A

Explanation:

Lambda Destinations on Failure: Allow routing asynchronous function invocations to specified resources (like another Lambda function) upon failure.

Error Handling: The error-handling Lambda receives details about the failure, enabling logging and custom actions.

Direct Integration: This solution leverages native Lambda functionality for a simpler implementation.

Question: 189

A company is preparing to migrate an application to the company's first AWS environment Before this migration, a developer is creating a proof-of-concept application to validate a model for building and deploying container-based applications on AWS.

Which combination of steps should the developer take to deploy the containerized proof-of-concept application with the LEAST operational effort? (Select TWO.)

A. Package the application into a zip file by using a command line tool Upload the package to Amazon S3

B. Package the application into a container image by using the Docker CLI. Upload the image to Amazon Elastic Container Registry (Amazon ECR)

C. Deploy the application to an Amazon EC2 instance by using AWS CodeDeploy.

D. Deploy the application to Amazon Elastic Kubernetes Service (Amazon EKS) on AWS Fargate

E. Deploy the application to Amazon Elastic Container Service (Amazon ECS) on AWS Fargate

Answer: BE

Explanation:

Containerization: Packaging the application as a container image promotes portability and standardization. Docker is the standard tool for containerization.

Amazon ECR: ECR is a managed container registry designed to work seamlessly with AWS container services.

Fargate: ECS Fargate provides serverless container orchestration, minimizing operational overhead for this proof-of-concept.

Reference:

Docker: https://www.docker.com/

Amazon ECR: https://aws.amazon.com/ecr/

Question: 190

A company runs an application on AWS The application stores data in an Amazon DynamoDB table Some queries are taking a long time to run These slow queries involve an attribute that is not the table's partition key or sort key

The amount of data that the application stores in the DynamoDB table is expected to increase significantly. A developer must increase the performance of the queries.

Which solution will meet these requirements'?

A. Increase the page size for each request by setting the Limit parameter to be higher than the default value Configure the application to retry any request that exceeds the provisioned throughput.

B. Create a global secondary index (GSI). Set query attribute to be the partition key of the index

C. Perform a parallel scan operation by issuing individual scan requests in the parameters specify the segment for the scan requests and the total number of segments for the parallel scan.

D. Turn on read capacity auto scaling for the DynamoDB table. Increase the maximum read capacity units (RCUs).

Answer: B

Explanation:

Global Secondary Index (GSI): GSIs enable alternative query patterns on a DynamoDB table by using different partition and sort keys.

Addressing Query Bottleneck: By making the slow-query attribute the GSI's partition key, you optimize queries on that attribute.

Scalability: GSIs automatically scale to handle increasing data volumes.

Reference:

Amazon DynamoDB Global Secondary Indexes: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

Question: 191

A developer maintains a critical business application that uses Amazon DynamoDB as the primary data store The DynamoDB table contains millions of documents and receives 30-60 requests each minute The developer needs to perform processing in near-real time on the documents when they are added or updated in the DynamoDB table

How can the developer implement this feature with the LEAST amount of change to the existing application code?

A. Set up a cron job on an Amazon EC2 instance Run a script every hour to query the table for changes and process the documents

B. Enable a DynamoDB stream on the table Invoke an AWS Lambda function to process the documents.

C. Update the application to send a PutEvents request to Amazon EventBridge. Create an EventBridge rule to invoke an AWS Lambda function to process the documents.

D. Update the application to synchronously process the documents directly after the DynamoDB write

_____ Answer: B

Explanation:

DynamoDB Streams: Capture near real-time changes to DynamoDB tables, triggering downstream actions.

Lambda for Processing: Lambda functions provide a serverless way to execute code in response to events like DynamoDB Stream updates.

Minimal Code Changes: This solution requires the least modifications to the existing application.

Reference:

DynamoDB Streams: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

AWS Lambda: https://aws.amazon.com/lambda/

Question: 192

A developer needs to build an AWS CloudFormation template that self-populates the AWS Region variable that deploys the CloudFormation template

What is the MOST operationally efficient way to determine the Region in which the template is being deployed?

A. Use the AWS:.Region pseudo parameter

B. Require the Region as a CloudFormation parameter

C. Find the Region from the AWS::Stackld pseudo parameter by using the Fn::Split intrinsic function

D. Dynamically import the Region by referencing the relevant parameter in AWS Systems Manager Parameter Store

Answer: A

Explanation:

Pseudo Parameters: CloudFormation provides pseudo parameters that reference runtime context, including the current AWS Region.

Operational Efficiency: The AWS::Region pseudo parameter offers the most direct and self-contained way to obtain the Region dynamically within the template.

Reference:

CloudFormation Pseudo Parameters: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo- parameter-reference.html

Question: 193

A company has an application that runs across multiple AWS Regions. The application is experiencing performance issues at irregular intervals. A developer must use AWS X-Ray to implement distributed tracing for the application to troubleshoot the root cause of the performance issues.

What should the developer do to meet this requirement?

A. Use the X-Ray console to add annotations for AWS services and user-defined services

B. Use Region annotation that X-Ray adds automatically for AWS services Add Region annotation for user-defined services

C. Use the X-Ray daemon to add annotations for AWS services and user-defined services

D. Use Region annotation that X-Ray adds automatically for user-defined services Configure X-Ray to add Region annotation for AWS services

Answer: B

Explanation:

Distributed Tracing with X-Ray: X-Ray helps visualize request paths and identify bottlenecks in applications distributed across Regions.

Region Annotations (Automatic for AWS Services): X-Ray automatically adds a Region annotation to segments representing calls to AWS services. This aids in tracing cross-Region traffic.

Region Annotations (Manual for User-Defined): For segments representing calls to user-defined services in different Regions, the developer needs to add the Region annotation manually to enable comprehensive tracing.

Reference:

AWS X-Ray: https://aws.amazon.com/xray/

Question: 194

A company is building a new application that runs on AWS and uses Amazon API Gateway to expose APIs Teams of developers are working on separate components of the application in parallel The company wants to publish an API without an integrated backend so that teams that depend on the application backend can continue the development work before the API backend development is complete.

Which solution will meet these requirements?

A. Create API Gateway resources and set the integration type value to MOCK Configure the method integration request and integration response to associate a response with an HTTP status code Create an API Gateway stage and deploy the API.

B. Create an AWS Lambda function that returns mocked responses and various HTTP status codes.

Create API Gateway resources and set the integration type value to AWS_PROXY Deploy the API.

C. Create an EC2 application that returns mocked HTTP responses Create API Gateway resources and set the integration type value to AWS Create an API Gateway stage and deploy the API.

D. Create API Gateway resources and set the integration type value set to HTTP_PROXY. Add mapping templates and deploy the API.

Create an AWS Lambda layer that returns various HTTP status codes Associate the Lambda layer with the API deployment

---

Answer: A

Explanation:

API Gateway Mocking: This feature is built for decoupling development dependencies. Here's the process:

Create resources and methods in your API Gateway.

Set the integration type to 'MOCK'.

Define Integration Responses, mapping HTTP status codes to desired mocked responses (JSON, etc.).

Deployment and Use:

Create a deployment stage for the API.

Frontend teams can call this API and get the mocked responses without a real backend.

Reference:

Mocking API Gateway APIs: https://docs.aws.amazon.com/apigateway/latest/developerguide/how- to-mock-integration.html

Question: 195

A company has an application that is hosted on Amazon EC2 instances The application stores objects in an Amazon S3 bucket and allows users to download objects from the S3 bucket A developer turns on S3 Block Public Access for the S3 bucket After this change, users report errors when they attempt to download objects The developer needs to implement a solution so that only users who are signed in to the application can access objects in the S3 bucket.

Which combination of steps will meet these requirements in the MOST secure way? (Select TWO.)

A. Create an EC2 instance profile and role with an appropriate policy Associate the role with the EC2 instances

B. Create an 1AM user with an appropriate policy. Store the access key ID and secret access key on the EC2 instances

C. Modify the application to use the S3 GeneratePresignedUrl API call

D. Modify the application to use the S3 GetObject API call and to return the object handle to the user

E. Modify the application to delegate requests to the S3 bucket.

Answer: AC

Explanation:

IAM Roles for EC2 (A): The most secure way to provide AWS permissions from EC2.

Create a role with a policy allowing s3:GetObject on the specific bucket.

Attach the role to an instance profile and associate that profile with your instances.

Pre-signed URLs (C): Temporary, authenticated URLs for specific S3 actions.

Modify the app to use the AWS SDK to call GeneratePresignedUrl.

Embed these URLs when a user is properly logged in, allowing download access.

Reference:

IAM Roles for EC2: https://docs.aws.amazon.com/IAM/latest/UserGuide/id roles use switch-role- ec2.html

Generating Presigned

URLs: https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.htm

Question: 196

An AWS Lambda function requires read access to an Amazon S3 bucket and requires read/write access to an Amazon DynamoDB table The correct 1AM policy already exists

What is the MOST secure way to grant the Lambda function access to the S3 bucket and the DynamoDB table?

A. Attach the existing 1AM policy to the Lambda function.

B. Create an 1AM role for the Lambda function Attach the existing 1AM policy to the role Attach the role to the Lambda function

C. Create an 1AM user with programmatic access Attach the existing 1AM policy to the user. Add the user access key ID and secret access key as environment variables in the Lambda function.

D. Add the AWS account root user access key ID and secret access key as encrypted environment variables in the Lambda function

Answer: B

Explanation:

Principle of Least Privilege: Granting specific permissions through an IAM role is more secure than directly attaching policies to a function or using root user credentials.

IAM Roles for Lambda: Designed to provide temporary credentials to Lambda functions, enhancing security.

Reusability: The existing IAM policy ensures the correct S3 and DynamoDB access is granted.

Reference:

IAM Roles for Lambda Documentation: https://docs.aws.amazon.com/lambda/latest/dg/lambda- intro-execution-role.html

IAM Best Practices: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

Question: 197

A developer is designing a serverless application for a game in which users register and log in through a web browser The application makes requests on behalf of users to a set of AWS Lambda functions that run behind an Amazon API Gateway HTTP API

The developer needs to implement a solution to register and log in users on the application's sign-in page. The solution must minimize operational overhead and must minimize ongoing management of user identities.

Which solution will meet these requirements'?

A. Create Amazon Cognito user pools for external social identity providers Configure 1AM roles for the identity pools.

B. Program the sign-in page to create users' 1AM groups with the 1AM roles attached to the groups

C. Create an Amazon RDS for SQL Server DB instance to store the users and manage the permissions to the backend resources in AWS

D. Configure the sign-in page to register and store the users and their passwords in an Amazon DynamoDB table with an attached IAM policy.

Answer: A

Explanation:

Amazon Cognito User Pools: A managed user directory service, simplifying user registration and login.

Social Identity Providers: Cognito supports integration with external providers (e.g., Google, Facebook), reducing development effort.

IAM Roles for Authorization: Cognito-managed IAM roles grant fine-grained access to AWS resources (like Lambda functions).

Operational Overhead: Cognito minimizes the need to manage user identities and credentials independently.

Reference:

Amazon Cognito Documentation https://docs.aws.amazon.com/cognito/

Cognito User Pools for Web

Applications: https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-app- integration.html

Question: 198

A developer supports an application that accesses data in an Amazon DynamoDB table. One of the item attributes is expirationDate in the timestamp format. The application uses this attribute to find items, archive them, and remove them from the table based on the timestamp value

The application will be decommissioned soon, and the developer must find another way to implement this functionality. The developer needs a solution that will require the least amount of code to write.

Which solution will meet these requirements?

A. Enable TTL on the expirationDate attribute in the table. Create a DynamoDB stream. Create an AWS Lambda function to process the deleted items. Create a DynamoDB trigger for the Lambda function.

B. Create two AWS Lambda functions one to delete the items and one to process the items Create a DynamoDB stream Use the DeleteItem API operation to delete the items based on the expirationDate attribute Use the GetRecords API operation to get the items from the DynamoDB stream and process them

C. Create two AWS Lambda functions, one to delete the items and one to process the items. Create an Amazon EventBndge scheduled rule to invoke the Lambda Functions Use the DeleteItem API operation to delete the items based on the expirationDate attribute. Use the GetRecords API operation to get the items from the DynamoDB table and process them.

D. Enable TTL on the expirationDate attribute in the table Specify an Amazon Simple Queue Service (Amazon SQS> dead-letter queue as the target to delete the items Create an AWS Lambda function to process the items

Answer: A

Explanation:

TTL for Automatic Deletion: DynamoDB's Time-to-Live effortlessly deletes expired items without manual intervention.

DynamoDB Stream: Captures changes to the table, including deletions of expired items, triggering downstream actions.

Lambda for Processing: A Lambda function connected to the stream provides custom logic for handling the deleted items.

Code Efficiency: This solution leverages native DynamoDB features and stream-based processing, minimizing the need for custom code.

Reference:

DynamoDB TTL

Documentation: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html

DynamoDB Streams

Documentation: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html

Question: 199

A developer is building an ecommerce application that uses multiple AWS Lambda functions. Each function performs a specific step in a customer order workflow, such as order processing and inventory management.

The developer must ensure that the Lambda functions run in a specific order.

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

A. Configure an Amazon Simple Queue Service (Amazon SQS) queue to contain messages about each step a function must perform. Configure the Lambda functions to run sequentially based on the order of messages in the SQS queue.

B. Configure an Amazon Simple Notification Service (Amazon SNS) topic to contain notifications about each step a function must perform. Subscribe the Lambda functions to the SNS topic. Use subscription filters based on the step each function must perform.

C. Configure an AWS Step Functions state machine to invoke the Lambda functions in a specific order.

D. Configure Amazon EventBridge Scheduler schedules to invoke the Lambda functions in a specific order.

Answer: C

Explanation:

The requirement here is to ensure that Lambda functions are executed in a specific order. AWS Step Functions is a low-code workflow orchestration service that enables you to sequence AWS services, such as AWS Lambda, into workflows. It is purpose-built for situations like this, where different steps need to be executed in a strict sequence.

AWS Step Functions: Step Functions allows developers to design workflows as state machines, where each state corresponds to a particular function. In this case, the developer can create a Step Functions state machine where each step (order processing, inventory management, etc.) is represented by a Lambda function.

Operational Overhead: Step Functions have very low operational overhead because it natively handles retries, error handling, and function sequencing.

Alternatives:

Amazon SQS (Option A): While SQS can manage message ordering, it requires more manual handling of each step and the logic to sequentially invoke the Lambda functions.

Amazon SNS (Option B): SNS is a pub/sub service and is not designed to handle sequences of Lambda executions.

EventBridge (Option D): EventBridge Scheduler allows you to invoke Lambda functions based on scheduled times, but it doesn't directly support sequencing based on workflow logic.

Therefore, AWS Step Functions is the most appropriate solution due to its native orchestration capabilities and minimal operational complexity.

Reference:

AWS Step Functions documentation

Question: 200

A company hosts a client-side web application for one of its subsidiaries on Amazon S3. The web application can be accessed through Amazon CloudFront from https://www.example.com. After a successful rollout, the company wants to host three more client-side web applications for its remaining subsidiaries on three separate S3 buckets.

To achieve this goal, a developer moves all the common JavaScript files and web fonts to a central S3 bucket that serves the web applications. However, during testing, the developer notices that the browser blocks the JavaScript files and web fonts.

What should the developer do to prevent the browser from blocking the JavaScript files and web fonts?

A. Create four access points that allow access to the central S3 bucket. Assign an access point to each web application bucket.

B. Create a bucket policy that allows access to the central S3 bucket. Attach the bucket policy to the central S3 bucket.

C. Create a cross-origin resource sharing (CORS) configuration that allows access to the central S3 bucket. Add the CORS configuration to the central S3 bucket.

D. Create a Content-MD5 header that provides a message integrity check for the central S3 bucket. Insert the Content-MD5

header for each web application request.

Answer: C

Explanation:

Question: 201

A company runs a new application on AWS Elastic Beanstalk. The company needs to deploy updates to the application. The updates must not cause any downtime for application users. The deployment must forward a specified percentage of incoming client traffic to a new application version during an evaluation period.

Which deployment type will meet these requirements?

A. Rolling

B. Traffic-splitting

C. In-place

D. Immutable

Answer: B

Explanation:

AWS Elastic Beanstalk supports several deployment policies, and in this case, the requirement is to forward a specific percentage of traffic to the new version without causing downtime. The Traffic-splitting deployment policy is the most appropriate choice.

Traffic-splitting Deployment: This deployment method allows you to gradually shift a specified percentage of incoming traffic from the old environment version to the new one. During the evaluation period, if any issues are detected, the traffic can be redirected back to the old version.

No Downtime: This method ensures no downtime since both versions of the application run concurrently, and traffic is split between them.

Alternatives:

Rolling deployments (Option A): These gradually replace instances but may result in partial downtime if some instances fail during deployment.

In-place deployments (Option C): In-place deployments replace instances without creating new ones, which can lead to downtime.

Immutable deployments (Option D): While this ensures no downtime by creating entirely new instances, it doesn't provide traffic splitting during the evaluation phase.

Reference:

Elastic Beanstalk Deployment Policies

Question: 202

A developer needs to retrieve all data from an Amazon DynamoDB table that matches a particular partition key.

Which solutions will meet this requirement in the MOST operationally efficient way? (Select TWO.)

A. Use the Scan API and a filter expression to match on the key.

B. Use the GetItem API with a request parameter for key that contains the partition key name and specific key value.

C. Use the ExecuteStatement API and a filter expression to match on the key.

D. Use the GetItem API and a PartiQL statement to match on the key.

E. Use the ExecuteStatement API and a PartiQL statement to match on the key.

Answer: B, E

Explanation:

Question: 203

A company has a web application that contains an Amazon API Gateway REST API. A developer has created an AWS CloudFormation template for the initial deployment of the application. The developer has deployed the application successfully as part of an AWS CodePipeline continuous integration and continuous delivery (CI/CD) process. All resources and methods are available through the deployed stage endpoint.

The CloudFormation template contains the following resource types:

- AWS::ApiGateway::RestApi
- AWS::ApiGateway::Resource
- AWS::ApiGateway::Method
- AWS:ApiGateway::Stage
- AWS::ApiGateway:;Deployment

The developer adds a new resource to the REST API with additional methods and redeploys the template. CloudFormation reports that the deployment is successful and that the stack is in the UPDATE_COMPLETE state. However, calls to all new methods are returning 404 (Not Found) errors.

What should the developer do to make the new methods available?

A. Specify the disable-rollback option during the update-stack operation.

B. Unset the Cloud Forma lion stack failure options.

C. Add an AWS CodeBuild stage lo CodePipeline to run the aws apigateway create-deployment AWS CLI command.

D. Add an action to CodePipeline to run the aws cloudfront create-invalidation AWS CLI command.

Answer: C

Explanation:

Question: 204

A company is developing an application that will be accessed through the Amazon API Gateway REST API. Registered users should be the only ones who can access certain resources of this API. The token being used should expire automatically and needs to be refreshed periodically.

How can a developer meet these requirements?

A. Create an Amazon Cognito identity pool, configure the Amazon Cognito Authorizer in API Gateway, and use the temporary credentials generated by the identity pool.

B. Create and maintain a database record for each user with a corresponding token and use an AWS Lambda authorizer in API Gateway.

C. Create an Amazon Cognito user pool, configure the Cognito Authorizer in API Gateway, and use the identity or access token.

D. Create an 1AM user for each API user, attach an invoke permissions policy to the API. and use an I AM authorizer in API Gateway.

Answer: C

Explanation:

Question: 205

A developer manages a website that distributes its content by using Amazon CloudFront. The website's static artifacts are stored in an Amazon S3 bucket.

The developer deploys some changes and can see the new artifacts in the S3 bucket. However, the changes do not appear on the webpage that the CloudFront distribution delivers.

How should the developer resolve this issue?

A. Configure S3 Object Lock to update to the latest version of the files every time an S3 object is updated.

B. Configure the S3 bucket to clear all old objects from the bucket before new artifacts are uploaded.

C. Set CloudFront to invalidate the cache after the artifacts have been deployed to Amazon S3.

D. Set CloudFront to modify the distribution origin after the artifacts have been deployed to Amazon S3.

Answer: C

Explanation:

Question: 206

A company had an Amazon RDS for MySQL DB instance that was named mysql-db. The DB instance was deleted within the past 90 days. A developer needs to find which 1AM user or role deleted the DB instance in the AWS environment. Which solution will provide this information?

A. Retrieve the AWS CloudTrail events for the resource mysql-db where the event name is DeleteDBInstance. Inspect each event.

B. Retrieve the Amazon CloudWatch log events from the most recent log stream within the rds/mysql-db log group. Inspect the log events.

C. Retrieve the AWS X-Ray trace summaries. Filter by services with the name mysql-db. Inspect the ErrorRootCauses values within each summary.

D. Retrieve the AWS Systems Manager deletions inventory Filter the inventory by deletions that have a TypeName value of RDS. Inspect the deletion details.

Answer: A

Explanation:

Question: 207

A company is using an Amazon API Gateway REST API endpoint as a webhook to publish events from an on-premises source control management (SCM) system to Amazon EventBridge. The company has configured an EventBridge rule to listen for the events and to control application deployment in a central AWS account. The company needs to receive the same events across multiple receiver AWS accounts.

How can a developer meet these requirements without changing the configuration of the SCM system?

A. Deploy the API Gateway REST API to all the required AWS accounts. Use the same custom domain name for all the gateway endpoints so that a single SCM webhook can be used for all events from all accounts.

B. Deploy the API Gateway REST API to all the receiver AWS accounts. Create as many SCM webhooks as the number of AWS accounts.

C. Grant permission to the central AWS account for EventBridge to access the receiver AWS accounts. Add an EventBridge event bus on the receiver AWS accounts as the targets to the existing EventBridge rule.

D. Convert the API Gateway type from REST API to HTTP API.

Answer: C

Explanation:

Question: 208

A developer is creating AWS CloudFormation templates to manage an application's deployment in Amazon Elastic Container Service (Amazon ECS) through AWS CodeDeploy. The developer wants to automatically deploy new versions of the application to a percentage of users before the new version becomes available for all users.

How should the developer manage the deployment of the new version?

A. Modify the CloudFormation template to include a Transform section and the AWS::CodeDeploy::BlueGreen hook.

B. Deploy the new version in a new CloudFormation stack. After testing is complete, update the application's DNS records for the new stack.

C. Run CloudFormation stack updates on the application stack to deploy new application versions when they are available.

D. Create a nested stack for the new version. Include a Transform section and the AWS::CodeDeploy::BlueGreen hook.

Answer: A

Explanation:

Question: 209

A developer is building a highly secure healthcare application using serverless components. This application requires writing temporary data to /Imp storage on an AWS Lambda function.

How should the developer encrypt this data?

A. Enable Amazon EBS volume encryption with an AWS KMS key in the Lambda function configuration so that all storage attached to the Lambda function is encrypted.

B. Set up the Lambda function with a role and key policy to access an AWS KMS key. Use the key to generate a data key used to encrypt all data prior to writing to Amp storage.

C. Use OpenSSL to generate a symmetric encryption key on Lambda startup. Use this key to encrypt the data prior to writing to /tmp.

D. Use an on-premises hardware security module (HSM) to generate keys, where the Lambda function requests a data key from the HSM and uses that to encrypt data on all requests to the function.

Answer: B

Explanation:

Question: 210

A developer is creating an AWS Serverless Application Model (AWS SAM) template. The AWS SAM template contains the definition of multiple AWS Lambda functions, an Amazon S3 bucket, and an Amazon CtoudFront distribution. One of the Lambda functions runs on Lambda@Edge in the

CloudFront distribution. The S3 bucket is configured as an origin for the CloudFront distribution.

When the developer deploys the AWS SAM template in the eu-west-1 Region, the creation of the stack fails.

Which of the following could be the reason for this issue?

A. CloudFront distributions can be created only in the us-east-1 Region.

B. Lambda@Edge functions can be created only in the us-east-1 Region.

C. A single AWS SAM template cannot contain multiple Lambda functions.

D. The CloudFront distribution and the S3 bucket cannot be created in the same Region.

Answer: B

Explanation:

Question: 211

A developer has written a distributed application that uses micro services. The microservices are running on Amazon EC2 instances. Because of message volume, the developer is unable to match log output from each microservice to a specific transaction. The developer needs to analyze the message flow to debug the application.

Which combination of steps should the developer take to meet this requirement? (Select TWO.)

A. Download the AWS X-Ray daemon. Install the daemon on an EC2 instance. Ensure that the EC2 instance allows UDP traffic on port 2000.

B. Configure an interface VPC endpoint to allow traffic to reach the global AWS X-Ray daemon on TCP port 2000.

C. Enable AWS X-Ray. Configure Amazon CloudWatch to push logs to X-Ray.

D. Add the AWS X-Ray software development kit (SDK) to the microservices. Use X-Ray to trace requests that each microservice makes.

E. Set up Amazon CloudWatch metric streams to collect streaming data from the microservices.

Answer: A, D

Explanation:

Question: 212

A developer is building an application that uses Amazon DynamoDB. The developer wants to retrieve multiple specific items from the database with a single API call. Which DynamoDB API call will meet these requirements with the MINIMUM impact on the database?

A. BatchGetItem

B. GetItem

C. Scan

D. Query

Answer: A

Explanation:

Question: 213

An application stores user data in Amazon S3 buckets in multiple AWS Regions. A developer needs to implement a solution that analyzes the user data in the S3 buckets to find sensitive information. The analysis findings from all the S3 buckets must be available in the eu-west-2 Region.

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

A. Create an AWS Lambda function to generate findings. Program the Lambda function to send the findings to another S3 bucket in eu-west-2.

B. Configure Amazon Made to generate findings. Use Amazon EventBridge to create rules that copy the findings to eu-west-2.

C. Configure Amazon Inspector to generate findings. Use Amazon EventBridge to create rules that copy the findings to eu-west-2.

D. Configure Amazon Macie to generate findings and to publish the findings to AWS CloudTrail. Use a CloudTrail trail to copy the results to eu-west-2.

Answer: B

Explanation:

Question: 214

A company uses Amazon DynamoDB as a data store for its order management system. The company frontend application stores orders in a DynamoDB table. The DynamoDB table is configured to send change events to a DynamoDB stream. The company uses an AWS Lambda

function to log and process the incoming orders based on data from the DynamoDB stream.

An operational review reveals that the order quantity of incoming orders is sometimes set to 0. A developer needs to create a dashboard that will show how many unique customers this problem affects each day.

What should the developer do to implement the dashboard?

A. Grant the Lambda function's execution role permissions to upload logs to Amazon CloudWatch Logs. Implement a CloudWatch Logs Insights query that selects the number of unique customers for orders with order quantity equal to 0 and groups the results in 1-day periods. Add the CloudWatch Logs Insights query to a CloudWatch dashboard.

B. Use Amazon Athena to query AWS CtoudTrail API logs for API calls. Implement an Athena query that selects the number of unique customers for orders with order quantity equal to 0 and groups the results in 1-day periods. Add the Athena query to an Amazon CloudWatch dashboard.

C. Configure the Lambda function to send events to Amazon EventBridge. Create an EventBridge rule that groups the number of unique customers for orders with order quantity equal to 0 in 1-day periods. Add a CloudWatch dashboard as the target of the rule.

D. Turn on custom Amazon CloudWatch metrics for the DynamoDB stream of the DynamoOB table. Create a CloudWatch alarm that groups the number of unique customers for orders with order quantity equal to 0 in 1-day periods. Add the CloudWatch alarm to a CloudWatch dashboard.

_____ Answer: A

Explanation:

Question: 215

A developer has created an AWS Lambda function to provide notification through Amazon Simple Notification Service (Amazon SNS) whenever a file is uploaded to Amazon S3 that is larger than 50 MB. The developer has deployed and tested the Lambda function by using the CLI. However, when the event notification is added to the S3 bucket and a 3.000 MB file is uploaded, the Lambda function does not launch.

Which of the following Is a possible reason for the Lambda function's inability to launch?

A. The S3 event notification does not activate for files that are larger than 1.000 MB.

B. The resource-based policy for the Lambda function does not have the required permissions to be invoked by Amazon S3.

C. Lambda functions cannot be invoked directly from an S3 event.

D. The S3 bucket needs to be made public.

_____ Answer: B

Explanation:

Question: 216 _____

A developer is creating an application that must transfer expired items from Amazon DynamoDB to Amazon S3. The developer sets up the DynamoDB table to automatically delete items after a specific TTL. The application must process the items in DynamoDB and then must store the expired items in Amazon S3. The entire process, including item processing and storage in Amazon S3, will take 5 minutes.

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

A. Configure DynamoDB Accelerator (DAX) lo query for expired items based on the TTL Save the results to Amazon S3.

B. Configure DynamoDB Streams to invoke an AWS Lambda function. Program the Lambda function to process the items and to store the expired items in Amazon S3.

C. Deploy a custom application on an Amazon Elastic Container Service (Amazon ECS) cluster on Amazon EC2 instances. Program the custom application to process the items and to store the expired items in Amazon S3.

D. Create an Amazon EventBridge rule to invoke an AWS Lambda function. Program the Lambda function to process The items and to store the expired items in Amazon S3.

Answer: B

Explanation:

Question: 217

An application ingests data from an Amazon Kinesis data stream. The shards in the data stream are set for normal traffic.

During tests for peak traffic, the application ingests data slowly. A developer needs to adjust the data stream to handle the peak traffic.

What should the developer do to meet this requirement MOST cost-effectively?

A. Install the Kinesis Producer Library {KPL) to ingest data into the data stream.

B. Switch to on-demand capacity mode for the data stream. Specify a partition key when writing data to the data stream.

C. Decrease the amount of time that data is kept in the data stream by using the DecreaseStreamRetention Period API operation.

D. Increase the shard count in the data stream by using the UpdateShardCount API operation.

Answer: D

Explanation:

Question: 218

A company has an application that is deployed on AWS Elastic Beanstalk. The application generates user-specific PDFs and stores the

PDFs in an Amazon S3 bucket. The application then uses Amazon Simple Email Service (Amazon SES) to send the PDFs by email to subscribers.

Users no longer access the PDFs 90 days after the PDFs are generated. The S3 bucket is not versioned and contains many obsolete PDFs.

A developer must reduce the number of files in the S3 bucket by removing PDFs that are older than 90 days.

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

A. Update the application code. In the code, add a rule to scan all the objects in the S3 bucket every day and to delete objects after 90 days.

B. Create an AWS Lambda function. Program the Lambda function to scan all the objects in the S3 bucket every day and to delete objects after 90 days.

C. Create an S3 Lifecycle rule for the S3 bucket to expire objects after 90 days.

D. Partition the S3 objects with a <year>/<month>/<day> key prefix. Create an AWS Lambda function to remove objects that have prefixes that have reached the expiration date.

_____ Answer: C

Explanation:

Question: 219

A company runs an ecommerce application on AWS. The application stores data in an Amazon Aurora database.

A developer is adding a caching layer to the application. The caching strategy must ensure that the application always uses the most recent value for each data item.

Which caching strategy will meet these requirements?

A. Implement a TTL strategy for every item that is saved in the cache.

B. Implement a write-through strategy for every item that is created and updated.

C. Implement a lazy loading strategy for every item that is loaded.

D. Implement a read-through strategy for every item that is loaded.

_____ Answer: B

Explanation:

Question: 220

A developer is monitoring an application that runs on an Amazon EC2 Instance. The developer has configured a custom Amazon CloudWatch metric with data granularity of 1 second. It any issues occur, the developer wants to be notified within 30 seconds by Amazon Simple Notification Service (Amazon SNS).

What should the developer do to meet this requirement?

A. Configure a high-resolution CloudWatch alarm.

B. Set up a custom CloudWatch dashboard.

C. Use Amazon CloudWatch Logs Insights.

D. Change to a default CloudWatch metric.

Answer: A

Explanation:

Question: 221

A company is developing a serverless application that requires storage of sensitive API keys as environment variables for various services. The application requires the automatic rotation of the encryption keys every year.

Which solution will meet these requirements with no development effort?

A. Encrypt the environment variables by using AWS Secrets Manager. Set up automatic rotation in Secrets Manager.

B. Encrypt the environment variables by using AWS Key Management Service (AWS KMS) customer managed keys. Enable automatic key rotation.

C. Encrypt the environment variables by using AWS Key Management Service (AWS KMS) AWS managed keys. Configure a custom AWS Lambda function to automate key rotation.

D. Encrypt the environment variables by using AWS Systems Manager Parameter Store. Set up automatic rotation in Parameter Store.

Answer: A

Explanation:

Question: 222

A developer needs to use a code template to create an automated deployment of an application onto Amazon EC2 instances. The template must be configured to repeat deployment, installation, and updates of resources for the application. The template must be able to create identical environments and roll back to previous versions.

Which solution will meet these requirements?

A. Use AWS Amplify for automatic deployment templates. Use a traffic-splitting deployment to copy any deployments. Modify any resources created by Amplify, if necessary.

B. Use AWS CodeBuild for automatic deployment. Upload the required AppSpec file template. Save the appspec.yml file in the root directory folder of the revision. Specify the deployment group that includes the EC2 instances for the deployment.

C. Use AWS CloudFormation to create an infrastructure template in JSON format to deploy the EC2 instances. Use Cloud Formation helper scripts to install the necessary software and to start the application. Call the scripts directly from the template.

D. Use AWS AppSync to deploy the application. Upload the template as a GraphQL schema. Specify the EC2 instances for deployment of the application. Use resolvers as a version control mechanism and to make any updates to the deployments.

_____ Answer: C

Explanation:

Question: 223

A company uses AWS X-Ray to monitor a serverless application. The components of the application have different request rates. The user interactions and transactions are important to trace, but they are low in volume. The background processes such as application health checks, polling, and connection maintenance generate high volumes of read-only requests.

Currently, the default X-Ray sampling rules are universal for all requests. Only the first request per second and some additional requests are recorded. This setup is not helping the company review the requests based on service or request type.

A developer must configure rules to trace requests based on service or request properties. The developer must trace the user interactions and transactions without wasting effort recording minor background tasks.

Which solution will meet these requirements?

A. Disable sampling for high-volume read-only requests. Sample at a lower rate for all requests that handle user interactions or transactions.

B. Disable sampling and trace all requests for requests that handle user interactions or transactions. Sample high-volume read-only requests at a higher rate.

C. Disable sampling and trace all requests for requests that handle user interactions or transactions. Sample high-volume read-only requests at a lower rate.

D. Disable sampling for high-volume read-only requests. Sample at a higher rate for all requests that handle user interactions or transactions.

_____ Answer: C

Explanation:

Question: 224

A company has a serverless application that uses Amazon API Gateway backed by AWS Lambda proxy integration. The company is developing several backend APIs. The company needs a landing page to provide an overview of navigation to the APIs.

A developer creates a new /LandingPage resource and a new GET method that uses mock integration.

What should the developer do next to meet these requirements?

A. Configure the integration request mapping template with Content-Type of text/html and statusCode of 200. Configure the integration response mapping template with Content-Type of application/json. In the integration response mapping template, include the LandingPage HTML code that references the APIs.

B. Configure the Integration request mapping template with Content-Type of application/json. In the integration request mapping template, include the LandingPage HMTL code that references the APIs. Configure the integration response mapping template with Content-Type of text/html and statusCode of 200.

C. Configure the integration request mapping template with Content-Type of application/json and statusCode of 200. Configure the integration response mapping template with Content-Type of text/html. In the integration response mapping template, include the LandingPage HTML code that references the APIs.

D. Configure the integration request mapping template with Content-Type of text/html. In the integration request mapping template, include the LandingPage HTML code that references the APIs. Configure the integration response mapping template with Content-Type of application/json and statusCode of 200.

Answer: C

Explanation:

Question: 225

A developer is setting up infrastructure by using AWS Cloud Formation. If an error occurs when the resources described in the CloudFormation template are provisioned, successfully provisioned resources must be preserved. The developer must provision and update the CloudFormation stack by using the AWS CLI.

Which solution will meet these requirements?

A. Add an --enable-terminal ion-protection command line option to the create-stack command and the update-stack command.

B. Add a -disable-roll back command line option to the create-stack command and the update-stack command

C. Add a —parameters ParameterKey=P reserve Resources. ParameterValue=True command line option to the create-stack command and the update-stack command.

D. Add a -tags Key=PreserveResources.Value=True command line option to the create-stack command and the update-stack command.

Answer: B

Explanation:

Question: 226

A developer is receiving HTTP 400: ThrottlingException errors intermittently when calling the Amazon CloudWatch API. When a call fails, no data is retrieved.

What best practice should first be applied to address this issue?

A. Contact AWS Support for a limit increase.

B. Use the AWS CLI to get the metrics.

C. Analyze the applications and remove the API call.

D. Retry the call with exponential backoff.

Answer: D

Explanation:

Question: 227

A developer needs to troubleshoot an AWS Lambda function in a development environment. The Lambda function is configured in VPC mode and needs to connect to an existing Amazon RDS for SOL Server DB instance. The DB instance is deployed in a private subnet and accepts connections by using port 1433.

When the developer tests the function, the function reports an error when it tries to connect to the database.

Which combination of steps should the developer take to diagnose this issue? (Select TWO.)

A. Check that the function's security group has outbound access on port 1433 to the DB instance's security group. Check that the DB instance's security group has inbound access on port 1433 from the function's security group.

B. Check that the function's security group has Inbound access on port 1433 from the DB Instance's security group. Check that the DB instance's security group has outbound access on port 1433 to the function's security group.

C. Check that the VPC is set up for a NAT gateway. Check that the DB instance has the public access option turned on.

D. Check that the function's execution role permissions include rds:DescribeDBInstances, rds: ModifyDB Instance, and rds:DescribeDBSecurityGroups for the DB instance.

E. Check that the function's execution rote permissions include ec2: CreateNetworkInterface. ec2: DescribeNetworkInterfaces. and ec2: DeleteNetworkInterface.

Answer: A, E

Explanation:

Question: 228

A developer is building an application on AWS. The application has an Amazon API Gateway API that sends requests to an AWS Lambda function. The API is experiencing increased latency because the Lambda function has limited available CPU to fulfill the requests.

Before the developer deploys the API into production, the developer must configure the Lambda function to have more CPU.

Which solution will meet this requirement?

A. Increase the virtual CPU (vCPU) cores quota of the Lambda function.

B. Increase the amount of memory that is allocated to the Lambda function.

C. Increase the ephemeral storage size of the Lambda function.

D. Increase the timeout value of the Lambda function.

Answer: B

Explanation:

Question: 229

A company caches session information for a web application in an Amazon DynamoDB table. The company wants an automated way to delete old items from the table.

What is the simplest way to do this?

A. Write a script that deletes old records; schedule the script as a cron job on an Amazon EC2 instance.

B. Add an attribute with the expiration time; enable the Time To Live feature based on that attribute.

C. Each day, create a new table to hold session data; delete the previous day's table.

D. Add an attribute with the expiration time; name the attribute ItemExpiration.

Answer: B

Explanation:

Question: 230

A company requires that all applications running on Amazon EC2 use 1AM roles to gam access to AWS services. A developer is modifying an application that currently relies on 1AM user access keys stored in environment variables to access Amazon DynamoDB tables using boto, the AWS SDK for Python.

The developer associated a role with the same permissions as the 1AM user to the EC2 instance, then deleted the 1AM user. When the application was restarted, the AWS

Access Denied Exception messages started appearing in the application logs. The developer was able to use their personal account on the server to run DynamoDB API commands using the AWS CLI.

What is the MOST likely cause of the exception?

A. 1AM policies might take a few minutes to propagate to resources.

B. Disabled environment variable credentials are still being used by the application.

C. The AWS SDK does not support credentials obtained using an instance role.

D. The instance's security group does not allow access to http://169.254.169.254.

_____ Answer: B

Explanation:

Question: 231

A developer wants the ability to roll back to a previous version of an AWS Lambda function in the event of errors caused by a new deployment. How can the developer achieve this with MINIMAL impact on users?

A. Change the application to use an alias that points to the current version. Deploy the new version of the code Update the alias to use the newly deployed version. If too many errors are encountered, point the alias back to the previous version.

B. Change the application to use an alias that points to the current version. Deploy the new version of the code. Update the alias to direct 10% of users to the newly deployed version. If too many errors are encountered, send 100% of traffic to the previous version

C. Do not make any changes to the application. Deploy the new version of the code. If too many errors are encountered, point the application back to the previous version using the version number in the Amazon Resource Name (ARN).

D. Create three aliases: new, existing, and router. Point the existing alias to the current version. Have the router alias direct 100% of users to the existing alias. Update the application to use the router alias. Deploy the new version of the code. Point the new alias to this version. Update the router alias to direct 10% of users to the new alias. If too many errors are encountered, send 100% of traffic to the existing alias.

_____ Answer: A

Explanation:

Question: 232

A company has an online web application that includes a product catalog. The catalog is stored in an Amazon S3 bucket that is named DOC-EXAMPLE-BUCKET. The application must be able to list the objects in the S3 bucket and must be able to download objects through an 1AM policy.

Which policy allows MINIMUM access to meet these requirements?

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
  b
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject"
    J,
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*" )
]
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::DOC EXAMPLE BUCKET" b
```

B.

```
{
  "Effect": "Allow",
  "Action": [
    "s3:»"
  b
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
  )
1
```

```
"Version": "2012-10-17",
"Statement": [
  (
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
  h
  (
    "Effect": "Allow",
    "Action":[
      "s3:GetObject",
      "s3:PutObject",
      "s3:DeleteObject"
```

```
            1,
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*" )
        1


    "Version": "2012-10-17",
    "Statement": [
      (
        "Effect": "Allow",
        "Action":"s3:ListBucket",
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET" b


    "Version": "2012-10-17",
    "Statement": [
      (
        "Effect": "Allow", "Action":"s3:ListBucket",
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET" b
D. {
        "Effect": "Deny",
        "Action": (
          "s3:GetObject"
        J,
                  "Resource": "arn:aws:s3:::DOC EXAMPLE-BUCKET/*"
      )
    J
```

A. Option A
B. Option B

C. Option C

D. Option D

Answer: A

Explanation:

Question: 233

A developer is building an application that uses an AWS Lambda function to process data. The application requires minimum latency. The Lambda function must have predictable function start times. All setup activities for the execution environment must happen before invocation of the Lambda function.

Which solution will meet these requirements?

A. Increase the memory of the Lambda function to the maximum amount. Configure an Amazon EventBridge rule to schedule invocations of the Lambda function every minute to keep the execution environment active.

B. Optimize the static initialization code that runs when a new execution environment is prepared for the first time. Decrease and compress the size of the Lambda function package and the imported libraries and dependencies.

C. Increase the reserved concurrency of the Lambda function to the maximum value for unreserved account concurrency. Run any setup activities manually before the initial invocation of the Lambda function.

D. Publish a new version of the Lambda function. Configure provisioned concurrency for the Lambda function with the required minimum number of execution environments.

Answer: D

Explanation:

Question: 234

A company uses an AWS Lambda function to transfer files from an Amazon S3 bucket to the company's SFTP server. The Lambda function connects to the SFTP server by using credentials such as username and password. The company uses Lambda environment variables to store these credentials.

A developer needs to implement encrypted username and password credentials.

Which solution will meet these requirements?

A. Remove the user credentials from the Lambda environment. Implement 1AM database authentication.

B. Move the user credentials from Lambda environment variables to AWS Systems Manager Parameter Store.

C. Move the user credentials from Lambda environment variables to AWS Key Management Service (AWS KMS).

D. Move the user credentials from the Lambda environment to an encrypted .txt file. Store the file in an S3 bucket.

Answer: B

Explanation:

Question: 235

A developer is building a microservice that uses AWS Lambda to process messages from an Amazon Simple Queue Service (Amazon SQS) standard queue. The Lambda function calls external APIs to enrich the SOS message data before loading the data into an Amazon Redshift

data warehouse. The SOS queue must handle a maximum of 1.000 messages per second.

During initial testing, the Lambda function repeatedly inserted duplicate data into the Amazon Redshift table. The duplicate data led to a problem with data analysis. All duplicate messages were submitted to the queue within 1 minute of each other.

How should the developer resolve this issue?

A. Create an SOS FIFO queue. Enable message deduplication on the SOS FIFO queue.

B. Reduce the maximum Lambda concurrency that the SOS queue can invoke.

C. Use Lambda's temporary storage to keep track of processed message identifiers.

D. Configure a message group ID for every sent message. Enable message deduplication on the SQS standard queue.

Answer: A

Explanation:

Question: 236

A company launched an online portal to announce a new product that the company will release in 6 months. The portal requests that users enter an email address to receive communications about the product. The company needs to create a REST API that will store the email addresses in Amazon DynamoDB.

A developer has created an AWS Lambda function that can store the email addresses. The developer will deploy the Lambda function by using the AWS Serverless Application Model (AWS SAM). The developer must provide access to the Lambda function over HTTP.

Which solutions will meet these requirements with the LEAST additional configuration? (Select TWO.)

A. Expose the Lambda function by using function URLs.

B. Expose the Lambda function by using a Gateway Load Balancer.

C. Expose the Lambda function by using a Network Load Balancer.

D. Expose the Lambda function by using AWS Global Accelerator

E. Expose the Lambda function by using Amazon API Gateway.

Answer: A, E

Explanation:

Question: 237

A developer has created a large AWS Lambda function. Deployment of the function Is failing because of an InvalidParameterValueException error. The error message indicates that the unzipped size of the function exceeds the maximum supported value.

Which actions can the developer take to resolve this error? (Select TWO.)

A. Submit a quota increase request to AWS Support to increase the function to the required size.

B. Use a compression algorithm that is more efficient than ZIP.

C. Break up the function into multiple smaller functions.

D. Zip the .zip file twice to compress the file more.

E. Move common libraries, function dependencies, and custom runtimes into Lambda layers.

Answer: C, E

Explanation:

Question: 238

A developer is creating a new batch application that will run on an Amazon EC2 instance. The application requires read access to an Amazon S3 bucket. The developer needs to follow security best practices to grant S3 read access to the application.

Which solution meets these requirements?

A. Add the permissions to an 1AM policy. Attach the policy to a role. Attach the role to the EC2 instance profile.

B. Add the permissions inline to an 1AM group. Attach the group to the EC2 instance profile.

C. Add the permissions to an 1AM policy. Attach the policy to a user. Attach the user to the EC2 instance profile.

D. Add the permissions to an 1AM policy. Use 1AM web identity federation to access the S3 bucket with the policy.

Answer: A

Explanation:

Question: 239

An 1AM role is attached to an Amazon EC2 instance that explicitly denies access to all Amazon S3 API actions. The EC2 instance credentials file specifies the 1AM access key and secret access key, which allow full administrative access.

Given that multiple modes of 1AM access are present for this EC2 instance, which of the following is correct?

A. The EC2 instance will only be able to list the S3 buckets.

B. The EC2 instance will only be able to list the contents of one S3 bucket at a time.

C. The EC2 instance will be able to perform all actions on any S3 bucket.

D. The EC2 instance will not be able to perform any S3 action on any S3 bucket.

_____ Answer: D

Explanation:

Question: 240 _____

A developer is writing unit tests tor a new application that will be deployed on AWS. The developer wants to validate all pull requests with unit tests and merge the code with the main branch only when all tests pass

The developer stores the code in AWS CodeCommit and sets up AWS CodeBuild to run the unit tests. The developer creates an AWS Lambda function to start the CodeBuild task. The developer needs to identify the CodeCommit events in an Amazon EventBridge event that can invoke the Lambda function when a pull request is created or updated.

Which CodeCommit event will meet these requirements?

```
"source": ("aws.codecommit"],
■derail": {
      "event": ("pullRequestMergeStatusUpdated"],
»
```

```
"source": ["aws.codecommit"j,
"detail": {
      "event": ["pullRequestApprovalRuleCreated"] )
```

```
"source": ["aws.codecommit"],
"detail": (
                              "event": ["pullRequestSource3ranch(Jpdated", "pullRequestCreated"]
```

```
"source": ["aws.codeconrnit"),
"detail": (
      "event": ("pullRequestUpdated", "pullRequestSource3ranchCreated") )
```

A. Option A

B. Option B

C. Option C

D. Option D

Answer: C

Explanation:

https://aws.amazon.com/blogs/devops/automated-code-review-on-pull-requests-using-aws- codecommit-and-aws-codebuild/

Question: 241

A company has a website that displays a daily newsletter. When a user visits the website, an AWS Lambda function processes the browser's request and queries the company's on-premises database to obtain the current newsletter. The newsletters are stored in English. The Lambda function uses the Amazon Translate TranslateText API operation to translate the newsletters, and the translation is displayed to the user.

Due to an increase in popularity, the website's response time has slowed. The database is overloaded. The company cannot change the database and needs a solution that improves the response time of the Lambda function.

Which solution meets these requirements?

A. Change to asynchronous Lambda function invocation.

B. Cache the translated newsletters in the Lambda /tmp directory.

C. Enable TranslateText API caching.

D. Change the Lambda function to use parallel processing.

Answer: B

Explanation:

Question: 242

A developer creates an AWS Lambda function that is written in Java. During testing, the Lambda function does not work how the developer expected. The developer wants to use tracing capabilities to troubleshoot the problem.

Which AWS service should the developer use to accomplish this goal?

A. AWS Trusted Advisor

B. Amazon CloudWatch

C. AWS X-Ray

D. AWS CloudTrail

Answer: C

Explanation:

Question: 243

A developer is designing a fault-tolerant environment where client sessions will be saved.

How can the developer ensure that no sessions are lost if an Amazon EC2 instance fails?

A. Use sticky sessions with an Elastic Load Balancer target group.

B. Use Amazon SOS to save session data.

C. Use Amazon DynamoDB to perform scalable session handling.

D. Use Elastic Load Balancer connection draining to stop sending requests to failing instances.

Answer: C

Explanation:

Question: 244

A software company is launching a multimedia application. The application will allow guest users to access sample content before the users decide if they want to create an account to gain full access. The company wants to implement an authentication process that can identify users who have already created an account. The company also needs to keep track of the number of guest users who eventually create an account.

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

A. Create an Amazon Cognito user pool. Configure the user pool to allow unauthenticated users. Exchange user tokens for temporary credentials that allow authenticated users to assume a role.

B. Create an Amazon Cognito identity pool. Configure the identity pool to allow unauthenticated users. Exchange unique identity for temporary credentials that allow all users to assume a role.

C. Create an Amazon CloudFront distribution. Configure the distribution to allow unauthenticated users. Exchange user tokens for temporary credentials that allow all users to assume a role.

D. Create a role for authenticated users that allows access to all content. Create a role for unauthenticated users that allows access to only the sample content.

E. Allow all users to access the sample content by default. Create a role for authenticated users that allows access to the other

content.

Answer: B, D

Explanation:

Question: 245

A developer received the following error message during an AWS CloudFormation deployment:

Which action should the developer take to resolve this error?

A. Contact AWS Support to report an issue with the Auto Scaling Groups (ASG> service.

B. Add a DependsOn attribute to the ASGInstanceRole12345678 resource in the CloudFormation template. Then delete the stack.

C. Modify the CloudFormation template to retain the ASGInstanceRolet 2345678 resource. Then manually delete the resource after deployment.

D. Add a force parameter when calling CloudFormation with the role-am of ASGInstanceRole12345678.

Answer: C

Explanation:

Question: 246

A company has implemented a pipeline in AWS CodePipeline. The company Is using a single AWS account and does not use AWS Organizations. The company needs to test its AWS CloudFormation templates in its primary AWS Region and a disaster recovery Region.

Which solution will meet these requirements with the MOST operational efficiency?

A. In the CodePipeline pipeline, implement an AWS CodeDeploy action for each Region to deploy and test the Cloud Formation templates. Update CodePipeline and AWS CodeBuild with appropriate permissions.

B. Configure CodePipeline to deploy and test the Cloud Formation templates. Use CloudFormation StackSets to start deployment across both Regions.

C. Configure CodePipeline to invoke AWS CodeBuild to deploy and test the CloudFormation templates in each Region. Update CodeBuild and CloudFormation with appropriate permissions.

D. Use the Snyk action in CodePipeline to deploy and test the CloudFormation templates in each Region.

Answer: B

Explanation:

Question: 247

A developer needs to modify an application architecture to meet new functional requirements. Application data is stored in Amazon DynamoDB and processed tor analysis in a nightly batch. The system analysts do not want to wait until the next day to view the processed data and have asked to have it available in near-real time.

Which application architecture pattern would enable the data to be processed as it is received?

A. Event driven

B. Client-server d riven

C. Fan-out driven

D. Schedule driven

Answer: A

Explanation:

Question: 248

A company is working on a new serverless application. A developer needs to find an automated way to deploy AWS Lambda functions and the dependent Infrastructure with minimum coding effort. The application also needs to be reliable.

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

A. Build the application by using shell scripts to create .zip files for each Lambda function. Manually upload the .zip files to the AWS Management Console.

B. Build the application by using the AWS Serverless Application Model (AWS SAM). Use a continuous integration and continuous delivery (CI/CD) pipeline and the SAM CLI to deploy the Lambda functions.

C. Build the application by using shell scripts to create .zip files for each Lambda function. Upload the .zip files. Deploy the .zip files as Lambda functions by using the AWS CLI in a continuous integration and continuous delivery (CI/CD) pipeline.

D. Build a container for each Lambda function. Store the container images in AWS CodeArtifact.
Deploy the containers as Lambda functions by using the AWS CLI in a continuous integration and continuous delivery (CI/CD) pipeline.

Answer: B

Explanation:

Question: 249

A company maintains a REST service using Amazon API Gateway and the API Gateway native API key validation. The company recently launched a new registration page, which allows users to sign up for the service. The registration page creates a new API key using CreateApiKey and sends the new key to the user. When the user attempts to call the API using this key. the user receives a 403 Forbidden error. Existing users are unaffected and can still call the API.

What code updates will grant these new users access to the API?

A. The createDeploymer.t method must be called so the API can be redeployed to include the newly created API key.

B. The updateAuthorizer method must be called to update the API's authorizer to include the newly created API key

C. The importApiKeys method must be called to import all newly created API keys into the current stage of the API.

D. The createUsagePlanKey method must be called to associate the newly created API key with the correct usage plan.

_____ Answer: D

Explanation:

Question: 250

A web application is using Amazon Kinesis Data Streams for clickstream data that may not be consumed for up to 12 hours.

How can the developer implement encryption at rest for data within the Kinesis Data Streams?

A. Enable SSL connections to Kinesis.

B. Use Amazon Kinesis Consumer Library.

C. Encrypt the data once it is at rest with a Lambda function.

D. Enable server-side encryption in Kinesis Data Streams.

_____ Answer: B

Explanation:

Question: 251

A developer created an AWS Lambda function that accesses resources in a VPC. The Lambda function polls an Amazon Simple Queue Service (Amazon SOS) queue for new messages through a VPC endpoint. Then the function calculates a rolling average of the numeric values that are contained in the messages. After initial tests of the Lambda function, the developer found that the value of the rolling average that the function returned was not accurate.

How can the developer ensure that the function calculates an accurate rolling average?

A. Set the function's reserved concurrency to 1. Calculate the rolling average in the function. Store the calculated rolling average in Amazon ElastiCache.

B. Modify the function to store the values in Amazon ElastiCache. When the function initializes, use the previous values from the cache to calculate the rolling average.

C. Set the function's provisioned concurrency to 1. Calculate the rolling average in the function. Store the calculated rolling average in Amazon ElastiCache.

D. Modify the function to store the values in the function's layers. When the function initializes, use the previously stored values to calculate the rolling average.

Answer: B

Explanation:

Question: 252

A company wants to migrate applications from its on-premises servers to AWS. As a first step, the company is modifying and migrating a non-critical application to a single Amazon EC2 instance. The application will store information in an Amazon S3 bucket. The company needs to follow security best practices when deploying the application on AWS.

Which approach should the company take to allow the application to interact with Amazon S3?

A. Create an 1AM role that has administrative access to AWS. Attach the role to the EC2 instance.

B. Create an 1AM user. Attach the AdministratorAccess policy. Copy the generated access key and secret key. Within the application code, use the access key and secret key along with the AWS SDK to communicate with Amazon S3.

C. Create an 1AM role that has the necessary access to Amazon S3. Attach the role to the EC2 instance.

D. Create an 1AM user. Attach a policy that provides the necessary access to Amazon S3. Copy the generated access key and secret key. Within the application code, use the access key and secret key along with the AWS SDK to communicate with Amazon S3.

Answer: C

Explanation:

Question: 253

A developer is creating an AWS Lambda function that will connect to an Amazon RDS for MySQL instance. The developer wants to store the database credentials. The database credentials need to be encrypted and the database password needs to be automatically rotated.

Which solution will meet these requirements?

A. Store the database credentials as environment variables for the Lambda function. Set the environment variables to rotate automatically.

B. Store the database credentials in AWS Secrets Manager. Set up managed rotation on the database credentials.

C. Store the database credentials in AWS Systems Manager Parameter Store as secure string parameters. Set up managed rotation on the parameters.

D. Store the database credentials in the X-Amz-Security-Token parameter. Set up managed rotation on the parameter.

_____ Answer: B

Explanation:

Question: 254 _____

A company is building an application to accept data from customers. The data must be encrypted at rest and in transit.

The application uses an Amazon API Gateway API that resolves to AWS Lambda functions. The Lambda functions store the data in an Amazon Aurora MySQL DB cluster. The application worked properly during testing.

A developer configured an Amazon CloudFront distribution with field-level encryption that uses an AWS Key Management Service (AWS KMS) key. After the configuration of the distribution, the application behaved unexpectedly. All the data in the database changed from plaintext to ciphertext.

The developer must ensure that the data is not stored in the database as the ciphertext from the CloudFront field-level encryption.

Which solution will meet this requirement?

A. Change the CloudFront Viewer protocol policy from "HTTP and HTTPS" to "HTTPS only."

B. Add a Lambda function that uses the KMS key to decrypt the data fields before saving the data to the database.

C. Enable encryption on the DB cluster by using the same KMS key that is used in CloudFront.

D. Request and deploy a new SSL certificate to use with the CloudFront distribution.

_____ Answer: B

Explanation:

Question: 255 _____

A developer is writing an application to analyze the traffic to a fleet of Amazon EC2 instances. The EC2 instances run behind a public Application Load Balancer (ALB). An HTTP server runs on each of the EC2 instances, logging all requests to a log file.

The developer wants to capture the client public IP addresses. The developer analyzes the log files and notices only the IP address of the ALB.

What must the developer do to capture the client public IP addresses in the log file?

A. Add a Host header to the HTTP server log configuration file.

B. Install the Amazon CloudWatch Logs agent on each EC2 instance. Configure the agent to write to the log file.

C. Install the AWS X-Ray daemon on each EC2 instance. Configure the daemon to write to the log file.

D. Add an X-Forwarded-For header to the HTTP server log configuration file.

Answer: D

Explanation:

Question: 256

A developer is creating an application that uses an AWS Lambda function to transform and load data from an Amazon S3 bucket. When the developer tests the application, the developer finds that some invocations of the Lambda function are slower than others. The developer needs to update the Lambda function to have predictable invocation durations that run with low latency. Any initialization activities, such as loading libraries and instantiating clients, must run during allocation time rather than during actual function invocations.

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

A. Create a schedule group in Amazon EventBridge Scheduler to invoke the Lambda function.

B. Configure provisioned concurrency for the Lambda function to have the necessary number of execution environments.

C. Use the SLATEST version of the Lambda function.

D. Configure reserved concurrency for the Lambda function to have the necessary number of execution environments.

E. Deploy changes, and publish a new version of the Lambda function.

Answer: B, D

Explanation:

Question: 257

A developer is writing a web application that must share secure documents with end users. The documents are stored in a private Amazon S3 bucket. The application must allow only authenticated users to download specific documents when requested, and only for a duration of 15 minutes.

How can the developer meet these requirements?

A. Copy the documents to a separate S3 bucket that has a lifecycle policy for deletion after 15 minutes.

B. Create a presigned S3 URL using the AWS SDK with an expiration time of 15 minutes.

C. Use server-side encryption with AWS KMS managed keys (SSE-KMS) and download the documents using HTTPS.

D. Modify the S3 bucket policy to only allow specific users to download the documents. Revert the change after 15 minutes.

Answer: B

Explanation:

Question: 258

A developer is implementing a serverless application by using AWS CloudFormation to provision Amazon S3 web hosting. Amazon API Gateway, and AWS Lambda functions. The Lambda function source code is zipped and uploaded to an S3 bucket. The S3 object key of the zipped source code is specified in the Lambda resource in the CloudFormation template.

The developer notices that there are no changes in the Lambda function every time the CloudFormation stack is updated.

How can the developer resolve this issue?

A. Create a new Lambda function alias before updating the CloudFormation stack.

B. Change the S3 object key or the S3 version in the CloudFormation template before updating the CloudFormation stack.

C. Upload the zipped source code to another S3 bucket before updating the CloudFormation stack.

D. Associate a code signing configuration with the Lambda function before updating the CloudFormation stack.

Answer: B

Explanation:

Question: 259

An application interacts with Amazon Aurora to store and track customer information. The primary database is set up with multiple read replicas for improving the performance of the read queries. However, one of the Aurora replicas is receiving most or all of the traffic, while the other Aurora replica remains idle.

How can this issue be resolved?

A. Disable application-level DNS caching.

B. Enable application-level DNS caching.

C. Enable application pooling.

D. Disable application pooling.

Answer: A

Explanation:

Question: 260 _____

A company runs continuous integration/continuous delivery (CI/CD) pipelines for its application on AWS CodePipeline. A developer must write unit tests and run them as part of the pipelines before

staging the artifacts for testing.

How should the developer incorporate unit tests as part of CI/CD pipelines?

A. Create a separate CodePipeline pipeline to run unit tests.

B. Update the AWS CodeBuild build specification to include a phase for running unit tests.

C. Install the AWS CodeDeploy agent on an Amazon EC2 instance to run unit tests.

D. Create a testing branch in a git repository for the pipelines to run unit tests.

Answer: B

Explanation:

Question: 261

A developer has designed an application to store incoming data as JSON files in Amazon S3 objects. Custom business logic in an AWS Lambda function then transforms the objects, and the Lambda function loads the data into an Amazon DynamoDB table. Recently, the workload has experienced sudden and significant changes in traffic. The flow of data to the DynamoDB table is becoming throttled.

The developer needs to implement a solution to eliminate the throttling and load the data into the DynamoDB table more consistently.

Which solution will meet these requirements?

A. Refactor the Lambda function into two functions. Configure one function to transform the data and one function to load the data into the DynamoDB table. Create an Amazon Simple Queue Service (Amazon SQS) queue in between the functions to hold the items as messages and to invoke the second function.

B. Turn on auto scaling for the DynamoDB table. Use Amazon CloudWatch to monitor the table's read and write capacity metrics and to track consumed capacity.

C. Create an alias for the Lambda function. Configure provisioned concurrency for the application to use.

D. Refactor the Lambda function into two functions. Configure one function to store the data in the DynamoDB table. Configure the second function to process the data and update the items after the data is stored in DynamoDB. Create a DynamoDB stream to invoke the second function after the data is

stored.

Answer: A

Explanation:

Question: 262

A developer compiles an AWS Lambda function and packages the result as a .zip file. The developer uses the Functions page on the Lambda console to attempt to upload the local packaged .zip file.
When pushing the package to Lambda, the console returns the following error:

An error occurred (RequestEntityTooLargeException) when calling the UpdateF

Which solutions can the developer use to publish the code? (Select TWO.)

A. Upload the package to Amazon S3. Use the Functions page on the Lambda console to upload the package from the S3 location.

B. Create an AWS Support ticket to increase the maximum package size.

C. Use the update-function-code AWS CLI command. Pass the -publish parameter.

D. Repackage the Lambda function as a Docker container image. Upload the image to Amazon Elastic Container Registry {Amazon ECR). Create a new Lambda function by using the Lambda console. Reference the image that is deployed to Amazon ECR.

E. Sign the .zip file digitally. Create a new Lambda function by using the Lambda console. Update the configuration of the new Lambda function to include the Amazon Resource Name (ARN) of the code signing configuration.

Answer: A, D

Explanation:

Question: 263

A developer is creating an application that uses an Amazon DynamoDB table. The developer needs to develop code that reads all records that were added to the table during the previous day. creates HTML reports, and pushes the reports into third-party storage. The item size varies from 1 KB to 4 KB, and the index structure is defined with the date. The developer needs to minimize the read capacity that the application requires from the DynamoDB table.

Which DynamoDB API operation should the developer use in the code to meet these requirements?

A. Query

B. Scan

C. BatchGetItem

D. GetItem

Answer: A

Explanation:

Question: 264

A developer needs to write an AWS CloudFormation template on a local machine and deploy a CloudFormation stack to AWS.

What must the developer do to complete these tasks?

A. Install the AWS CLI. Configure the AWS CLI by using an I AM user name and password.

B. Install the AWS CLI. Configure the AWS CLI by using an SSH key.

C. Install the AWS CLI. Configure the AWS CLI by using an 1AM user access key and secret key.

D. Install an AWS software development kit (SDK). Configure the SDK by using an X.509 certificate.

Answer: C

Explanation:

Question: 265

A developer is building an application that includes an AWS Lambda function that is written in .NET Core. The Lambda function's code needs to interact with Amazon DynamoDB tables and Amazon S3 buckets. The developer must minimize the Lambda function's deployment time and invocation duration.

Which solution will meet these requirements?

A. Increase the Lambda function's memory.

B. Include the entire AWS SDK for .NET in the Lambda function's deployment package.

C. Include only the AWS SDK for .NET modules for DynamoDB and Amazon S3 in the Lambda function's deployment package.

D. Configure the Lambda function to download the AWS SDK for .NET from an S3 bucket at runtime.

Answer: C

Explanation:

Question: 266

A developer is setting up a deployment pipeline. The pipeline includes an AWS CodeBuild build stage that requires access to a

database to run integration tests. The developer is using a buildspec.yml file to configure the database connection. Company policy requires automatic rotation of all database credentials.

Which solution will handle the database credentials MOST securely?

A. Retrieve the credentials from variables that are hardcoded in the buildspec.yml file. Configure an AWS Lambda function to rotate the credentials.

B. Retrieve the credentials from an environment variable that is linked to a SecureString parameter in AWS Systems Manager Parameter Store. Configure Parameter Store for automatic rotation.

C. Retrieve the credentials from an environment variable that is linked to an AWS Secrets Manager secret. Configure Secrets Manager for automatic rotation.

D. Retrieve the credentials from an environment variable that contains the connection string in plaintext. Configure an Amazon EventBridge event to rotate the credentials.

Answer: B

Explanation:

Question: 267

A developer is troubleshooting a three-tier application, which is deployed on Amazon EC2 instances. There is a connectivity problem between the application servers and the database servers.

Which AWS services or tools should be used to identify the faulty component? (Select TWO.)

A. AWS CloudTrail

B. AWS Trusted Advisor

C. Amazon VPC Flow Logs

D. Network access control lists

E. AWS Config rules

Answer: C, D

Explanation:

Question: 268

In a move toward using microservices, a company's management team has asked all development teams to build their services so that API requests depend only on that service's data store. One team is building a Payments service which has its own database; the service needs data that originates in the Accounts database. Both are using Amazon DynamoDB.

What approach will result in the simplest, decoupled, and reliable method to get near-real time updates from the Accounts

database?

A. Use AWS Glue to perform frequent ETL updates from the Accounts database to the Payments database.

B. Use Amazon ElastiCache in Payments, with the cache updated by triggers in the Accounts database.

C. Use Amazon Data Firehose to deliver all changes from the Accounts database to the Payments database.

D. Use Amazon DynamoDB Streams to deliver all changes from the Accounts database to the Payments database.

Answer: D

Explanation:

Question: 269

A company is developing a serverless application by using AWS Lambda functions. One of the Lambda functions needs to access an Amazon RDS DB instance. The DB instance is in a private subnet inside a VPC.

The company creates a role that includes the necessary permissions to access the DB instance. The company then assigns the role to the Lambda function. A developer must take additional action to give the Lambda function access to the DB instance.

What should the developer do to meet these requirements?

A. Assign a public IP address to the DB instance. Modify the security group of the DB instance to allow inbound traffic from the IP address of the Lambda function.

B. Set up an AWS Direct Connect connection between the Lambda function and the DB instance.

C. Configure an Amazon CloudFront distribution to create a secure connection between the Lambda function and the DB instance.

D. Configure the Lambda function to connect to the private subnets in the VPC. Add security group rules to allow traffic to the DB instance from the Lambda function.

Answer: D

Explanation:

Question: 270

A company is planning to deploy an application on AWS behind an Elastic Load Balancing (ELB) load balancer. The application uses an HTTP/HTTPS listener and must access the client IP addresses.

Which load-balancing solution meets these requirements?

A. Use an Application Load Balancer and the X-Forwarded-For headers.

B. Use a Network Load Balancer (NLB). Enable proxy protocol support on the NLB and the target application.

C. Use an Application Load Balancer. Register the targets by the instance ID.

D. Use a Network Load Balancer and the X-Forwarded-For headers.

Answer: B

Explanation:

Question: 271

A developer is designing an event-driven architecture. An AWS Lambda function that processes data needs to push processed data to a subset of four consumer Lambda functions. The data must be routed based on the value of one field in the data.

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

A. Create an Amazon Simple Queue Service {Amazon SQS) queue and even! source mapping for each consumer Lambda function. Add message routing logic to the data-processing Lambda function.

B. Create an Amazon Simple Notification Service (Amazon SNS) topic. Subscribe the four consumer Lambda functions to the topic. Add message filtering logic to each consumer Lambda function. Subscribe the data-processing Lambda function to the SNS topic.

C. Create a separate Amazon Simple Notification Service (Amazon SNS) topic and subscription for each consumer Lambda function. Add message routing logic to the data-processing Lambda function to publish to the appropriate topic.

D. Create a single Amazon Simple Notification Service {Amazon SNS) topic. Subscribe the four consumer Lambda functions to the topic. Add SNS subscription filter policies to each subscription. Configure the data-processing Lambda function to publish to the topic.

Answer: B

Explanation:

Question: 272

A developer is building an application that processes a stream of user-supplied data. The data stream must be consumed by multiple Amazon EC2 based processing applications in parallel and in real time. Each processor must be able to resume without losing data if there is a service interruption.

The application architect plans to add other processors in the near future, and wants to minimize the amount of data duplication involved.

Which solution will satisfy these requirements?

A. Publish the data to Amazon Simple Queue Service (Amazon SQS).

B. Publish the data to Amazon Data Firehose.

C. Publish the data to Amazon EventBridge.

D. Publish the data to Amazon Kinesis Data Streams.

Answer: D

Explanation:

Question: 273

A large company has its application components distributed across multiple AWS accounts. The company needs to collect and visualize trace data across these accounts.

What should be used to meet these requirements?

A. AWS X-Ray

B. Amazon CloudWatch

C. Amazon VPC flow logs

D. Amazon OpenSearch Service

Answer: A

Explanation:

Question: 274

A developer must cache dependent artifacts from Maven Central, a public package repository, as part of an application's build pipeline. The build pipeline has an AWS CodeArtifact repository where artifacts of the build are published. The developer needs a solution that requires minimum changes to the build pipeline.

Which solution meets these requirements?

A. Modify the existing CodeArtifact repository to associate an upstream repository with the public package repository.

B. Create a new CodeArtifact repository that has an external connection to the public package repository.

C. Create a new CodeArtifact domain that contains a new repository that has an external connection to the public package repository.

D. Modify the CodeArtifact repository resource policy to allow artifacts to be fetched from the public package repository.

Answer: A

Explanation:

Question: 275

A company is providing read access to objects in an Amazon S3 bucket for different customers. The company uses 1AM permissions to restrict access to the S3 bucket. The customers can access only their own files.

Due to a regulation requirement, the company needs to enforce encryption in transit for interactions with Amazon S3.

Which solution will meet these requirements?

A. Add a bucket policy to the S3 bucket to deny S3 actions when the aws:SecureTransport condition is equal to false.

B. Add a bucket policy to the S3 bucket to deny S3 actions when the s3:x-amz-acl condition is equal to public-read.

C. Add an 1AM policy to the 1AM users to enforce the usage of the AWS SDK.

D. Add an 1AM policy to the 1AM users that allows S3 actions when the s3:x-amz-acl condition is equal to bucket-owner-read.

Answer: A

Explanation:

Question: 276

A team is developing an application that is deployed on Amazon EC2 instances. During testing, the team receives an error. The EC2 instances are unable to access an Amazon S3 bucket.

Which steps should the team take to troubleshoot this issue? (Select TWO.)

A. Check whether the policy that is assigned to the JAM role that is attached to the EC2 instances grants access to Amazon S3.

B. Check the S3 bucket policy to validate the access permissions for the S3 bucket.

C. Check whether the policy that is assigned to the 1AM user that is attached to the EC2 instances grants access to Amazon S3.

D. Check the S3 Lifecycle policy to validate the permissions that are assigned to the S3 bucket.

E. Check the security groups that are assigned to the EC2 instances. Make sure that a rule is not blocking the access to Amazon S3.

Answer: A, B

Explanation:

Question: 277

A developer created several AWS Lambda functions that write data to a single Amazon S3 bucket. The developer configured all the Lambda functions to send logs and metrics to Amazon CloudWatch.

The developer receives reports that one of the Lambda functions writes data to the bucket very slowly. The developer needs to measure the latency between the problematic Lambda function and the S3 bucket.

Which solution will meet this requirement?

A. Enable AWS X-Ray on the Lambda function. In the generated trace map. select the line between Lambda and Amazon S3.

B. Query the Lambda function's log file in Amazon CloudWatch Logs Insights. Return the average of the auto-discovered ©duration field.

C. Enable CloudWatch Lambda Insights on the function. View the latency graph that CloudWatch Lambda Insights provides.

D. Enable AWS X-Ray on the Lambda function. Select Amazon S3 in the latency graph to view the latency histogram.

Answer: A

Explanation:

Question: 278

A developer is integrating Amazon ElastiCache in an application. The cache will store data from a database. The cached data must populate real-time dashboards. Which caching strategy will meet these requirements?

A. A read-through cache

B. A write-behind cache

C. A lazy-loading cache

D. A write-through cache

Answer: D

Explanation:

Question: 279

A company's application has an AWS Lambda function that processes messages from IoT devices. The company wants to monitor the Lambda function to ensure that the Lambda function is meeting its required service level agreement (SLA).

A developer must implement a solution to determine the application's throughput in near real time. The throughput must be based on the number of messages that the Lambda function receives and processes in a given time period. The Lambda function performs initialization and post-processing steps that must not factor into the throughput measurement.

What should the developer do to meet these requirements?

A. Use the Lambda function's ConcurrentExecutions metric in Amazon CloudWatch to measure the throughput.

B. Modify the application to log the calculated throughput to Amazon CloudWatch Logs. Use Amazon EventBridge to invoke a separate Lambda function to process the logs on a schedule.

C. Modify the application to publish custom Amazon CloudWatch metrics when the Lambda function receives and processes each message. Use the metrics to calculate the throughput.

D. Use the Lambda function's Invocations metric and Duration metric to calculate the throughput in Amazon CloudWatch.

Answer: C

Explanation:

Question: 280

A developer is using AWS CodeDeploy to launch an application onto Amazon EC2 instances. The application deployment fails during testing. The developer notices an IAM_ROLE_PERMISSIONS error code in Amazon CloudWatch logs.

What should the developer do to resolve the error?

A. Ensure that the deployment group is using the correct role name for the CodeDeploy service role.

B. Attach the AWSCodeDeployRoleECS policy to the CodeDeploy service role.

C. Attach the AWSCodeDeployRole policy to the CodeDeploy service role.

D. Ensure the CodeDeploy agent is installed and running on all instances in the deployment group.

Answer: C

Explanation:

Question: 281

A company is building a serverless application that uses AWS Lambda functions. The company needs to create a set of test events to test Lambda functions in a development environment. The test events will be created once and then will be used by all the developers in an 1AM developer group. The test events must be editable by any of the 1AM users in the 1AM developer group.

Which solution will meet these requirements?

A. Create and store the test events in Amazon S3 as JSON objects. Allow S3 bucket access to all 1AM users.

B. Create the test events. Configure the event sharing settings to make the test events shareable.

C. Create and store the test events in Amazon DynamoDB. Allow access to DynamoDB by using 1AM roles.

D. Create the test events. Configure the event sharing settings to make the test events private.

Answer: B

Explanation:

Question: 282

A developer has deployed an AWS Lambda function that is subscribed to an Amazon Simple Notification Service {Amazon SNS) topic. The developer must implement a solution to add a record of each Lambda function invocation to an Amazon Simple Queue Service {Amazon SQS) queue.

Which solution will meet this requirement?

A. Configure the SQS queue as a dead-letter queue for the Lambda function.

B. Create code that uses the AWS SDK to call the SQS SendMessage operation to add the invocation details to the SQS queue. Add the code to the end of the Lambda function.

C. Add two asynchronous invocation destinations to the Lambda function: one destination for successful invocations and one destination for failed invocations. Configure the SQS queue as the destination for each type. Create an Amazon CloudWatch alarm based on the DestinationDeliveryFailures metric to catch any message that cannot be delivered.

D. Add a single asynchronous invocation destination to the Lambda function to capture successful invocations. Configure the SQS queue as the destination. Create an Amazon CloudWatch alarm based on the DestinationDeliveryFailures metric to catch any message that cannot be delivered.

Answer: D

Explanation:

Question: 283

A development learn has an Amazon API Gateway REST API that is backed by an AWS Lambda function.

Users have reported performance issues for the Lambda function. The development team identified the source of the issues as a cold start of the Lambda function. The development team needs to reduce the time needed for the Lambda function to initialize.

Which solution will meet this requirement?

A. Change the Lambda concurrency lo reserved concurrency.

B. Increase the timeout of the Lambda function.

C. Increase the memory allocation of the Lambda function.

D. Configure provisioned concurrency for the Lambda function.

Answer: D

Explanation:

Question: 284

A developer is creating a stock trading application. The developer needs a solution to send text messages to application users to confirmation when a trade has been completed.

The solution must deliver messages in the order a user makes stock trades. The solution must not send duplicate messages.

Which solution will meet these requirements?

A. Configure the application to publish messages to an Amazon Data Firehose delivery stream. Configure the delivery stream to have a destination of each user's mobile phone number that is passed in the trade confirmation message.

B. Create an Amazon Simple Queue Service (Amazon SQS) FIFO queue. Use the SendMessageIn API call to send the trade confirmation messages to the queue. Use the SendMessageOut API to send the messages to users by using the information provided in the trade confirmation message.

C. Configure a pipe in Amazon EventBridge Pipes. Connect the application to the pipe as a source. Configure the pipe to use each user's mobile phone number as a target. Configure the pipe to send incoming events to the users.

D. Create an Amazon Simple Notification Service (SNS) FIFO topic. Configure the application to use the AWS SDK to publish notifications to the SNS topic to send SMS messages to the users.

Answer: C

Explanation:

Question: 285

A company offers a business-to-business software service that runs on dedicated infrastructure deployed in each customer's AWS account. Before a feature release, the company needs to run integration tests on real AWS test infrastructure. The test infrastructure consists of Amazon EC2 instances and an Amazon RDS database.

A developer must set up a continuous delivery process that will provision the test infrastructure across the different AWS accounts. The developer then must run the integration tests.

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

A. Use AWS CodeDeploy with AWS CloudFormation StackSets to deploy the infrastructure. Use Amazon CodeGuru to run the tests.

B. Use AWS CodePipeline with AWS CloudFormation StackSets to deploy the infrastructure. Use AWS CodeBuild to run the tests.

C. Use AWS CodePipeline with AWS CloudFormation change sets to deploy the infrastructure. Use a CloudFormation custom resource to run the tests.

D. Use AWS Serverless Application Model (AWS SAM) templates with AWS CloudFormation change sets to deploy the infrastructure. Use AWS CodeDeploy to run the tests.

Answer: B

Explanation:

Question: 286

A developer is making changes to a custom application that uses AWS Elastic Beanstalk.

Which solutions will update the Elastic Beanstalk environment with the new application version after the developer completes the changes? (Select TWO.)

A. Package the application code into a .zip file. Use the AWS Management Console to upload the .zip file and deploy the packaged application.

B. Package the application code into a .tar file. Use the AWS Management Console to create a new application version from the .tar file. Update the environment by using the AWS CLI.

C. Package the application code into a .tar file. Use the AWS Management Console to upload the .tar file and deploy the packaged application.

D. Package the application code into a .zip file. Use the AWS CLI to create a new application version from the .zip file and to update the environment.

E. Package the application code into a .zip file. Use the AWS Management Console to create a new application version from the .zip file. Rebuild the environment by using the AWS CLI.

Answer: A, D

Explanation:

Question: 287

A company has an AWS Step Functions state machine named myStateMachine. The company configured a service role for Step Functions. The developer must ensure that only the myStateMachine state machine can assume the service role.

A. "Condition": { "ArnLike": { "aws ":"arn:aws:states:ap-south-1:111111111111:stateMachine

" } }

B. "Condition": { "ArnLike": { "aws

":"arn:aws:states:ap-south-1:*:stateMachine

" } }

Answer: A

Explanation:

Comprehensive Detailed Step by Step Explanation with All AWS Developer Reference:

To ensure that only a specific AWS Step Functions state machine (myStateMachine) can assume the service role, you must configure the correct trust policy in AWS IAM.

Trust Policies: Trust policies determine which entities (services or users) are allowed to assume the role. In this case, we want to restrict the trust policy to only allow the specific state machine (myStateMachine) to assume the role.

Using ArnLike: The condition "ArnLike" is used to specify that the SourceArn (which refers to the ARN of the entity assuming the role) must match a specific ARN. Option A specifies the exact ARN of the myStateMachine state machine, ensuring that only this state machine can assume the role.

Option B: This option is incorrect because it uses a wildcard (*) for the account ID, which would allow any state machine in the ap-south-1 region to assume the role, not just the specific one.

Reference:

AWS Step Functions IAM Policies

Question: 288

A company stores customer credit reports in an Amazon S3 bucket. An analytics service uses standard Amazon S3 GET requests to access the reports. A developer must implement a solution to redact personally identifiable information (PII) from the reports before the reports reach the analytics service.

A. Load the S3 objects into Amazon Redshift by using a COPY command. Implement dynamic data masking. Refactor the analytics service to read from Amazon Redshift.

B. Set up an S3 Object Lambda function. Attach the function to an S3 Object Lambda Access Point.

Program the function to call a PII redaction API.

C. Use AWS Key Management Service (AWS KMS) to implement encryption in the S3 bucket. Reupload all the existing S3 objects. Give the kms

permission to the analytics service.

D. Create an Amazon Simple Notification Service (Amazon SNS) topic. Implement message data protection. Refactor the analytics service to publish data access requests to the SNS topic.

Answer: B

Explanation:

Comprehensive Detailed Step by Step Explanation with All AWS Developer Reference:

To redact PII from S3 objects before they are accessed by the analytics service, the most efficient solution is to use S3 Object Lambda. S3 Object Lambda allows you to add your own code (Lambda function) to process and transform data when it is retrieved from Amazon S3. You can attach a Lambda function to an S3 Object Lambda Access Point, which in this case would run a redaction API to remove PII from the reports.

Operational Efficiency: S3 Object Lambda handles data processing on the fly, without requiring the data to be permanently transformed or moved to another service (like Amazon Redshift).

Alternatives:

Option A: Loading the data into Amazon Redshift would require refactoring the analytics service and maintaining an additional data pipeline, increasing complexity.

Option C: Using AWS KMS for encryption protects data at rest and in transit, but it does not address PII redaction.

Option D: SNS is a messaging service and does not support direct data transformation.

Question: 289

A company is using the AWS Serverless Application Model (AWS SAM) to develop a social media application. A developer needs a quick way to test AWS Lambda functions locally by using test event payloads. The developer needs the structure of these test event payloads to match the actual events that AWS services create.

A. Create shareable test Lambda events. Use these test Lambda events for local testing.
B. Store manually created test event payloads locally. Use the sam local invoke command with the file path to the payloads.
C. Store manually created test event payloads in an Amazon S3 bucket. Use the sam local invoke command with the S3 path to the payloads.
D. Use the sam local generate-event command to create test payloads for local testing.

Answer: D

Explanation:

Comprehensive Detailed Step by Step Explanation with All AWS Developer Reference:

The AWS Serverless Application Model (SAM) includes features for local testing and debugging of AWS Lambda functions. One of the most efficient ways to generate test payloads that match actual AWS event structures is by using the sam local generate-event command.

sam local generate-event: This command allows developers to create pre-configured test event payloads for various AWS services

(e.g., S3, API Gateway, SNS). These generated events accurately reflect the format that the service would use in a live environment, reducing the manual work required to create these events from scratch.

Operational Overhead: This approach reduces overhead since the developer does not need to manually create or maintain test events. It ensures that the structure is correct and up-to-date with the latest AWS standards.

Alternatives:

Option A suggests using shareable test events, but manually creating or sharing these events introduces more overhead.

Option B and C both involve manually storing and maintaining test events, which adds unnecessary complexity compared to using sam local generate-event.

Reference:

AWS SAM CLI documentation

Question: 290

A developer is updating an Amazon API Gateway REST API to have a mock endpoint. The developer wants to update the integration request mapping template so the endpoint will respond to mock integration requests with specific HTTP status codes based on various conditions.

A. { if($input.params('integration') == "mock") "statusCode": 404 else "statusCode": 500 end }
B. { if($input.params('scope') == "internal") "statusCode": 200 else "statusCode": 500 end }
C. { if($input.path("integration")) "statusCode": 200 else "statusCode":404 end }
D. { if($context.integration.status) "statusCode": 200 else "statusCode": 500 end }

Answer: D

Explanation:

Comprehensive Detailed Step by Step Explanation with All AWS Developer Reference:

In this scenario, the developer is configuring a mock integration in API Gateway. The integration request mapping template allows you to map the incoming request data to a format that the API expects. For mock integration, it's common to return specific HTTP status codes based on the conditions.

Using $context.integration.status: The $context.integration.status variable refers to the status of the API Gateway integration, which is useful for generating responses based on the condition. Option D correctly uses this variable to determine the HTTP status code, returning 200 for a successful mock request or 500 for a failure.

Alternatives:

Options A, B, and C do not use the correct context variables for handling mock integrations. These options would not return the correct status codes based on the actual integration status.

Reference:

API Gateway Mapping Templates and Accessing Context Variables

Question: 291

A developer is creating an AWS Lambda function that needs network access to private resources in a VPC.

A. Attach the Lambda function to the VPC through private subnets. Create a security group that allows network access to the private resources. Associate the security group with the Lambda function.

B. Configure the Lambda function to route traffic through a VPN connection. Create a security group that allows network access to the private resources. Associate the security group with the Lambda function.

C. Configure a VPC endpoint connection for the Lambda function. Set up the VPC endpoint to route traffic through a NAT gateway.

D. Configure an AWS PrivateLink endpoint for the private resources. Configure the Lambda function to reference the PrivateLink endpoint.

Answer: A

Explanation:

Comprehensive Detailed Step by Step Explanation with All AWS Developer Reference:

When you need to provide an AWS Lambda function access to private resources in a VPC, the most common and straightforward approach is to attach the Lambda function to a VPC via private subnets. Once the Lambda function is associated with the VPC, you need to configure appropriate security groups to control the access to the private resources.

Lambda with VPC Access: Lambda functions can be attached to private subnets in a VPC, allowing them to access resources like RDS, EC2, or internal services within that VPC.

Security Groups: A security group acts as a virtual firewall for the Lambda function, ensuring that it can access only the necessary resources and ports in the VPC.

Alternatives:

Option B involves routing traffic through a VPN, which adds unnecessary complexity and operational overhead compared to simply attaching the Lambda to the VPC.

Option C requires configuring a VPC endpoint and a NAT gateway, which can be complex and costly.

Option D refers to AWS PrivateLink, which is used to access services over private connections, but it's unnecessary in this scenario unless you need a cross-VPC connection.

Reference:

Lambda functions in a VPC

Question: 292

A company hosts a batch processing application on AWS Elastic Beanstalk with instances that run the most recent version of Amazon Linux. The application sorts and processes large datasets. In recent weeks, the application's performance has decreased significantly during a peak period for traffic. A developer suspects that the application issues are related to the memory usage. The developer checks the Elastic Beanstalk console and notices that memory usage is not being tracked.

How should the developer gather more information about the application performance issues?

A. Configure the Amazon CloudWatch agent to push logs to Amazon CloudWatch Logs by using port 443.

B. Configure the Elastic Beanstalk .ebextensions directory to track the memory usage of the instances.

C. Configure the Amazon CloudWatch agent to track the memory usage of the instances.

D. Configure an Amazon CloudWatch dashboard to track the memory usage of the instances.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

To monitor memory usage in Amazon Elastic Beanstalk environments, it's important to understand that default Elastic Beanstalk monitoring capabilities in Amazon CloudWatch do not track memory usage, as memory metrics are not collected by default. Instead, the Amazon CloudWatch agent must be configured to collect memory usage metrics.

Why Option C is Correct:

The Amazon CloudWatch agent can be installed and configured to monitor system-level metrics such as memory and disk utilization.

To enable memory tracking, developers need to install the CloudWatch agent on the Amazon Elastic Compute Cloud (EC2) instances associated with the Elastic Beanstalk environment.

After installation, the agent can be configured to collect memory metrics, which can then be sent to CloudWatch for further analysis.

How to Implement This Solution:

Install the CloudWatch Agent:

Use .ebextensions or AWS Systems Manager to install and configure the CloudWatch agent on the EC2 instances running in the Elastic Beanstalk environment.

Modify CloudWatch Agent Configuration:

Create a config.json file that specifies memory usage tracking and other desired metrics.

Enable Metrics Reporting:

The CloudWatch agent can push the metrics to CloudWatch, where they can be monitored.

Why Other Options are Incorrect:

Option A: Configuring the agent to push logs is not sufficient to track memory metrics. This option addresses logging but not

system-level metrics like memory usage.

Option B: The .ebextensions directory is used to customize Elastic Beanstalk environments but does not directly track memory metrics without additional configuration of the CloudWatch agent.

Option D: Configuring a CloudWatch dashboard will only visualize the metrics that are already being collected. It will not enable memory usage tracking.

AWS Documentation Reference:

Amazon CloudWatch Agent Overview

Elastic Beanstalk Customization Using .ebextensions

Monitoring Custom Metrics

Question: 293

A company is creating a new application that gives users the ability to upload and share short video files. The average size of the video files is 10 MB. After a user uploads a file, a message needs to be placed into an Amazon Simple Queue Service (Amazon SQS) queue so the file can be processed. The files need to be accessible for processing within 5 minutes.
Which solution will meet these requirements MOST cost-effectively?

A. Write the files to Amazon S3 Glacier Deep Archive. Add the S3 location of the files to the SQS queue.
B. Write the files to Amazon S3 Standard. Add the S3 location of the files to the SQS queue.
C. Write the files to an Amazon Elastic Block Store (Amazon EBS) General Purpose SSD volume. Add the EBS location of the files to the SQS queue.
D. Write messages that contain the contents of the uploaded files to the SQS queue.

_____ Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

Amazon S3 Standard provides immediate access to files and is cost-effective for files that need to be accessed within 5 minutes.

By adding the S3 location to the SQS queue, you avoid transferring large files directly, which is both more efficient and

scalable.

Why Other Options are Incorrect:

Option A: S3 Glacier Deep Archive is designed for archival storage with retrieval times ranging from minutes to hours, which does not meet the 5-minute requirement.

Option C: Amazon EBS is designed for block storage attached to EC2 instances, which adds unnecessary complexity and cost.

Option D: SQS is not designed to handle large file content directly and has message size limits (256 KB).

AWS Documentation Reference:

Amazon S3 Overview

Amazon SQS Best Practices

Question: 294

A developer is building a three-tier web application that should be able to handle a minimum of 5000 requests per minute. Requirements state that the web tier should be completely stateless while the application maintains session state for the users.

How can session data be externalized, keeping latency at the LOWEST possible value?

A. Create an Amazon RDS instance, then implement session handling at the application level to leverage a database inside the RDS database instance for session data storage.

B. Implement a shared file system solution across the underlying Amazon EC2 instances, then implement session handling at the application level to leverage the shared file system for session data storage.

C. Create an Amazon ElastiCache (Memcached) cluster, then implement session handling at the application level to leverage the cluster for session data storage.

D. Create an Amazon DynamoDB table, then implement session handling at the application level to leverage the table for session data storage.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option C is Correct:

Amazon ElastiCache (Memcached) provides low-latency, in-memory caching suitable for session storage. It ensures stateless web tier operations and supports the high throughput of 5000 requests per minute.

Why Other Options are Incorrect:

Option A: RDS has higher latency compared to in-memory caching solutions like ElastiCache.

Option B: Shared file systems introduce additional complexity and are not ideal for low-latency session data storage.

Option D: DynamoDB has low latency but is less performant than ElastiCache for in-memory session management.

AWS Documentation Reference:

Amazon ElastiCache for Session State Management

Question: 295

A company has an AWS Step Functions state machine named myStateMachine. The company configured a service role for Step Functions. The developer must ensure that only the myStateMachine state machine can assume the service role.

Which statement should the developer add to the trust policy to meet this requirement?

A. "Condition": { "ArnLike": { "aws:SourceArn":"urn:aws:states:ap-south-1:111111111111:stateMachine:myStateMachine" } }

B. "Condition": { "ArnLike": { "aws:SourceArn":"arn:aws:states:ap-south- 1:*:stateMachine:myStateMachine" } }

C. "Condition": { "StringEquals": { "aws:SourceAccount": "111111111111" } }

D. "Condition": { "StringNotEquals": { "aws:SourceArn":"arn:aws:states:ap-south-1:111111111111:stateMachine:myStateMachine" } }

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:

The ArnLike condition with the specific ARN for myStateMachine ensures that only this state machine can assume the role. The format urn:aws:states is correct for specifying Step Functions resources.

Why Other Options are Incorrect:

Option B: Wildcards (*) in the ARN allow more resources to assume the role, which violates the requirement.

Option C: This condition restricts the account but not the specific state machine.

Option D: A StringNotEquals condition is used to deny specific values, which does not ensure exclusivity for the desired state machine.

AWS Documentation Reference:

IAM Trust Policies for Step Functions

Question: 296

A company runs a web application on Amazon EC2 instances behind an Application Load Balancer. The application uses Amazon DynamoDB as its database. The company wants to ensure high performance for reads and writes.

Which solution will meet these requirements MOST cost-effectively?

A. Configure auto-scaling for the DynamoDB table with a target utilization of 70%. Set the minimum and maximum capacity units based on the expected workload.

B. Use DynamoDB on-demand capacity mode for the table. Specify a maximum throughput higher than the expected peak read and write capacity units.

C. Use DynamoDB provisioned throughput mode for the table. Create an Amazon CloudWatch alarm on the ThrottledRequests metric. Invoke an AWS Lambda function to increase provisioned capacity.

D. Create an Amazon DynamoDB Accelerator (DAX) cluster. Configure the application to use the DAX endpoint.

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:

Auto-scaling with a target utilization ensures the DynamoDB table dynamically adjusts capacity based on workload, maintaining high performance while optimizing cost. Setting a reasonable target utilization minimizes overprovisioning and throttling risks.

Why Other Options are Incorrect:

Option B: On-demand capacity is costlier than provisioned throughput for predictable workloads.

Option C: Using manual CloudWatch alarms and Lambda for scaling is less efficient and adds operational overhead.

Option D: DAX accelerates read performance but does not improve write performance.

AWS Documentation Reference:

DynamoDB Auto Scaling

Question: 297

A company regularly receives route status updates from its delivery trucks as events in Amazon EventBridge. The company is building an API-based application in a VPC that will consume and process the events to create a delivery status dashboard. The API application must not be available by using public IP addresses because of security and compliance requirements.
How should the company send events from EventBridge to the API application?

A. Create an AWS Lambda function that runs in the same VPC as the API application. Configure the function as an EventBridge target. Use the function to send events to the API.

B. Create an internet-facing Application Load Balancer (ALB) in front of the API application. Associate a security group with rules that block access from all external sources except for EventBridge. Configure the ALB as an EventBridge target.

C. Create an internet-facing Network Load Balancer (NLB) in front of the API application. Associate a security group with rules that block access from all external sources except for EventBridge. Configure the NLB as an EventBridge target.

D. Use the application API endpoint in the VPC as a target for EventBridge. Send events directly to the application API endpoint from EventBridge.

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:
Running an AWS Lambda function within the same VPC ensures secure communication without exposing the API application to public IP addresses. The Lambda function can serve as a secure EventBridge target to send events to the API.

Why Other Options are Incorrect:

Option B & C: Internet-facing load balancers expose public IP addresses, which violates compliance requirements.

Option D: EventBridge cannot directly target an endpoint within a private VPC without intermediary services like Lambda.

AWS Documentation Reference:

EventBridge Targets

Question: 298

A company has a large amount of data in an Amazon DynamoDB table. A large batch of data is appended to the table once each day. The company wants a solution that will make all the existing and future data in DynamoDB available for analytics on a long-term basis.
Which solution meets these requirements with the LEAST operational overhead?

A. Configure DynamoDB incremental exports to Amazon S3.

B. Configure Amazon DynamoDB Streams to write records to Amazon S3.

C. Configure Amazon EMR to copy DynamoDB data to Amazon S3.

D. Configure Amazon EMR to copy DynamoDB data to Hadoop Distributed File System (HDFS).

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:

DynamoDB supports incremental exports to Amazon S3 natively, making data analytics-ready with minimal operational overhead.

Why Other Options are Incorrect:

Option B: DynamoDB Streams require additional processing logic to write to S3, increasing complexity.

Option C & D: Using EMR for data movement adds unnecessary operational overhead compared to native exports.

AWS Documentation Reference:

DynamoDB Exports to S3

Question: 299

An ecommerce company is planning to migrate an on-premises Microsoft SQL Server database to the AWS Cloud. The company needs to migrate the database to SQL Server Always On availability groups. The cloud-based solution must be highly available.

Which solution will meet these requirements?

A. Deploy three Amazon EC2 instances with SQL Server across three Availability Zones. Attach one Amazon Elastic Block Store (Amazon EBS) volume to the EC2 instances.

B. Migrate the database to Amazon RDS for SQL Server. Configure a Multi-AZ deployment and read replicas.

C. Deploy three Amazon EC2 instances with SQL Server across three Availability Zones. Use Amazon FSx for Windows File Server as the storage tier.

D. Deploy three Amazon EC2 instances with SQL Server across three Availability Zones. Use Amazon S3 as the storage tier.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option C is Correct:

SQL Server Always On availability groups require a shared storage solution. Amazon FSx for Windows File Server provides the shared storage necessary to implement Always On availability groups in a highly available configuration.

Why Other Options are Incorrect:

Option A: A single EBS volume cannot provide shared storage for Always On availability groups.

Option B: RDS does not support SQL Server Always On availability groups.

Option D: S3 is not a suitable storage tier for SQL Server database operations.

AWS Documentation Reference:

Amazon FSx for Windows File Server

Question: 300

A social media application is experiencing high volumes of new user requests after a recent marketing campaign. The application is served by an Amazon RDS for MySQL instance. A solutions architect examines the database performance and notices high CPU usage and many "too many connections" errors that lead to failed requests on the database. The solutions architect needs to address the failed requests.

Which solution will meet this requirement?

A. Deploy an Amazon DynamoDB Accelerator (DAX) cluster. Configure the application to use the DAX cluster.
B. Deploy an RDS Proxy. Configure the application to use the RDS Proxy.
C. Migrate the database to an Amazon RDS for PostgreSQL instance.
D. Deploy an Amazon ElastiCache (Redis OSS) cluster. Configure the application to use the ElastiCache cluster.

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

RDS Proxy manages database connections efficiently, reducing overhead on the RDS instance and mitigating "too many connections" errors.

Why Other Options are Incorrect:

Option A: DAX is for DynamoDB, not RDS.

Option C: Migration to PostgreSQL does not address the current issue.

Option D: ElastiCache is useful for caching but does not solve connection pool issues.

AWS Documentation Reference:

Amazon RDS Proxy

Question: 301

A company hosts a stateless web application with low data storage in a single AWS Region. The company wants to increase the resiliency of the application to include a multi-Region presence. The company wants to set the recovery time objective (RTO) and recovery point objective (RPO) to hours. The company needs a low-cost and low-complexity disaster recovery (DR) strategy. Which DR strategy should the company use?

A. Warm standby
B. Pilot light
C. Backup and restore
D. Multi-site active-active

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

The pilot light strategy keeps a minimal version of the environment in another Region and scales up during a disaster. It achieves an RTO and RPO of hours at a low cost and complexity.

Why Other Options are Incorrect:

Option A: Warm standby is more expensive as it keeps a scaled-down, fully functioning version running in another Region.

Option C: Backup and restore has a longer RTO and RPO than hours.

Option D: Multi-site active-active is costly and more complex than required.

AWS Documentation Reference:

Disaster Recovery Strategies on AWS

Question: 302

A developer is building an application that uses an Amazon RDS for PostgreSQL database. To meet security requirements, the developer needs to ensure that data is encrypted at rest. The developer must be able to rotate the encryption keys on demand.

A. Use an AWS KMS managed encryption key to encrypt the database.

B. Create a symmetric customer managed AWS KMS key. Use the key to encrypt the database.

C. Create a 256-bit AES-GCM encryption key. Store the key in AWS Secrets Manager, and enable managed rotation. Use the key to encrypt the database.

D. Create a 256-bit AES-GCM encryption key. Store the key in AWS Secrets Manager. Configure an AWS Lambda function to perform key rotation. Use the key to encrypt the database.

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

A customer-managed AWS Key Management Service (KMS) key allows for encryption at rest and provides the ability to rotate the key on demand. This ensures compliance with security requirements for key management and database encryption.

RDS integrates natively with AWS KMS, allowing the use of a customer-managed key for encrypting data at rest.

Key rotation can be managed directly in AWS KMS without needing custom solutions.

Why Other Options are Incorrect:

Option A: AWS KMS managed encryption keys (AWS-owned keys) do not support key rotation on demand.

Option C & D: Storing keys in AWS Secrets Manager with custom rotation is not a recommended approach for database encryption. AWS KMS is designed specifically for secure key management and encryption.

AWS Documentation Reference:

Encrypting Amazon RDS Resources

AWS Key Management Service (KMS)

Question: 303

A developer is migrating a containerized application from an on-premises environment to an Amazon ECS cluster.
In the on-premises environment, the container uses a Docker file to store the application. Service dependency configurations such as databases, caches, and storage volumes are stored in a docker- compose.yml file.
Both files are located at the top level of the code base that the developer needs to containerize.

When the developer deploys the code to Amazon ECS, the instructions from the Docker file are carried out. However, none of the configurations from docker-compose.yml are applied.
The developer needs to resolve the error and ensure the configurations are applied.

A. Store the file path for the docker-compose.yml file as a Docker label. Add the label to the ECS cluster's container details.

B. Add the details from the docker-compose.yml file to an ECS task definition. Associate the task with the ECS cluster.

C. Create a namespace in the ECS cluster. Associate the docker-compose.yml file to the namespace.

D. Update the service type of the ECS cluster to REPLICA, and redeploy the stack.

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

Amazon ECS does not natively process docker-compose.yml files. Instead, the configurations from docker-compose.yml must be converted into ECS-compatible configurations within a task definition. Task definitions are the primary way to specify container configurations in ECS, including service dependencies like databases, caches, and volumes.

Steps to Resolve the Error:

Extract the configurations from the docker-compose.yml file.

Map the dependencies and settings to the appropriate ECS task definition fields.

Deploy the task definition to the ECS cluster.

Why Other Options are Incorrect:

Option A: Docker labels do not directly impact ECS task execution or integrate with ECS service configurations.

Option C: ECS namespaces do not exist as a feature.

Option D: Changing the service type to REPLICA does not resolve the issue of missing service dependency configurations.

AWS Documentation Reference:

Amazon ECS Task Definitions

Migrating Docker Compose Workloads to ECS

Question: 304

A developer created a Node.js-based AWS Lambda function by using a container image of an AWS OS-only base image. There is a new security patch for Node.js that must be patched to the new Lambda function.

Which solution will meet this requirement?

A. Set the runtime update mode of the Lambda function to Auto.

B. Patch the runtime version by redeploying the same version of the Lambda function.

C. Rebuild the Lambda container code with the latest version of the AWS OS base image. Publish a new version of the Lambda function.

D. Rebuild the Lambda container code with the latest Node.js patch version. Publish a new version of the Lambda function.

Answer: D

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option D is Correct:

When using a container-based AWS Lambda function, you are responsible for updating the base image for runtime patches. Rebuilding the container with the latest Node.js patch version ensures the function is updated with the required security patches. Publishing a new version of the Lambda function makes the updated image available for use.

Why Other Options are Incorrect:

Option A: The runtime update mode applies to functions using AWS-managed runtimes, not container-based runtimes.

Option B: Redeploying the same version of the Lambda function does not apply the security patch.

Option C: Rebuilding with the AWS OS base image does not guarantee that the latest Node.js patch version is included.

AWS Documentation Reference:

Lambda Function Container Images

Node.js Updates in AWS Lambda

Question: 305

A developer is preparing to deploy an AWS CloudFormation stack for an application from a template that includes an IAM user.

The developer needs to configure the application's resources to retain the IAM user after successful creation. However, the developer also needs to configure the application to delete the IAM user if the stack rolls back.

A. Update CloudFormation template with the following deletion policy:

AWSTemplateFormatVersion: '2010-05-09'

Resources:

appUser:

Type: AWS::IAM::User

DeletionPolicy: Retain

B. Update CloudFormation template with the following deletion policy:

AWSTemplateFormatVersion: '2010-09-09'

Resources:

appUser:

Type: AWS::IAM::User

DeletionPolicy: RetainExceptOnCreate

C. Update the CloudFormation service role to include the following policy:

```
{
"Version": "2012-10-17",
"Statement": [{
  "Effect": "Allow",
  "Action": ["cloudformation:UpdateTerminationProtection"],
  "Resource": "*"
}]
}
```

D. Update the stack policy to include the following statements:

```
{
"Statement": [{
  "Effect": "Deny",
  "Action": "Update:*",
  "Principal": "*",
  "Resource": "*",
  "Condition": {
  "StringEquals": {
    "ResourceType": "AWS::IAM::User"
   }
  }
}]
}
```

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

The RetainExceptOnCreate deletion policy ensures that the IAM user is retained after successful stack creation but is deleted if the stack creation fails or rolls back. This meets both requirements.

Why Other Options are Incorrect:

Option A: The Retain policy retains the resource regardless of stack status and does not delete the IAM user upon rollback.

Option C: Updating the service role for termination protection does not address the specific deletion behavior for the IAM user.

Option D: Stack policy controls updates, not resource deletion behavior during rollbacks.

AWS Documentation Reference:

CloudFormation DeletionPolicy Attribute

Question: 306

A healthcare company uses AWS Amplify to host a patient management system. The system uses Amazon API Gateway to expose RESTful APIs. The backend logic of the system is handled by AWS Lambda functions.

One of the Lambda functions receives patient data that includes personally identifiable information (PII). The Lambda function sends the patient data to an Amazon DynamoDB table. The company must encrypt all patient data at rest and in transit before the data is stored in DynamoDB.

A. Configure the Lambda function to use AWS KMS keys with the AWS Database Encryption SDK to encrypt the patient data before sending the data to DynamoDB.

B. Use AWS managed AWS KMS keys to encrypt the data in the DynamoDB table.

C. Configure a DynamoDB stream on the table to invoke a Lambda function. Configure the Lambda function to use an AWS KMS key to encrypt the DynamoDB table and to update the table.

D. Use an AWS Step Functions workflow to transfer the data to an Amazon SQS queue. Configure a Lambda function to encrypt the data in the queue before sending the data to the DynamoDB table.

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:

Encrypting PII at rest and in transit before storing it in DynamoDB ensures end-to-end security. Using the AWS Database

Encryption SDK with KMS keys allows the Lambda function to encrypt data before transmission, meeting security and compliance requirements.

Why Other Options are Incorrect:

Option B: While AWS-managed KMS keys encrypt DynamoDB data at rest, they do not encrypt data in transit.

Option C: DynamoDB streams process updates after the data is written to the table, failing to encrypt PII in transit.

Option D: Step Functions and SQS add unnecessary complexity and still require encryption logic for both transit and at rest.

AWS Documentation Reference:

Encrypting Data in DynamoDB

AWS Database Encryption SDK

Question: 307

A developer is using an AWS CloudFormation template to create a pipeline in AWS CodePipeline. The template creates an Amazon S3 bucket that the pipeline references in a source stage. The template also creates an AWS CodeBuild project for a build stage. The pipeline sends notifications to an Amazon SNS topic. Logs for the CodeBuild project are stored in Amazon CloudWatch Logs.

The company needs to ensure that the pipeline's artifacts are encrypted with an existing customermanaged AWS KMS key. The developer has granted the pipeline permissions to use the KMS key. Which additional step will meet these requirements?

A. Create an Amazon S3 gateway endpoint that the pipeline can access.

B. In the CloudFormation template, use the KMS key to encrypt the logs in CloudWatch Logs.

C. Apply an S3 bucket policy that ensures the pipeline sends only encrypted objects to the S3 bucket.

D. Configure the notification topic to use the existing KMS key to enable encryption with the existing KMS key.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option C is Correct:

Ensuring that pipeline artifacts are encrypted with a customer-managed AWS KMS key involves configuring the S3 bucket policy to require encryption. This policy ensures all objects uploaded to the bucket are encrypted with the specified KMS key.

Why Other Options are Incorrect:

Option A: A gateway endpoint improves S3 access efficiency but does not enforce encryption.

Option B: Encrypting CloudWatch Logs is unrelated to securing pipeline artifacts.

Option D: Configuring SNS for encryption does not affect the artifacts stored in the S3 bucket.

AWS Documentation Reference:

Using Server-Side Encryption with S3 Bucket Policies

Question: 308

A developer previously deployed an AWS Lambda function as a .zip package. The developer needs to deploy the Lambda function as a container.

A. Create an Amazon ECR repository in the same AWS Region as the Lambda function. Package the Lambda function into a container image. Build the image and upload it to the Amazon ECR repository. Update the existing Lambda function configuration to specify the repository URI and container image tag.

B. Create an AWS SAM template that defines the Lambda function and its resources as code. Include a container image in the template, and store the container image in an Amazon S3 bucket. Deploy the AWS SAM template. Specify the S3 bucket URI.

C. Create an AWS CloudFormation template that defines the Lambda function and its resources as code. Include a container image in the template, and store the image in an Amazon S3 bucket. Deploy the CloudFormation template. Specify the S3 bucket URI.

D. Create an Amazon ECR repository in the same AWS Region as the Lambda function. Build the image and upload it to the Amazon ECR repository. Update the existing Lambda function to use the new image by specifying the repository URI.

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:
Converting a Lambda function to use a container image involves packaging the function code into a container image, storing the image in Amazon Elastic Container Registry (ECR), and updating the function to use the ECR repository URI.

Why Other Options are Incorrect:

Option B: SAM templates support container-based Lambda deployment, but storing the image in S3 is not applicable.

Option C: CloudFormation does not natively support specifying Lambda container images in S3.

Option D: While partially correct, it omits the need to specify the image tag for the deployment.

AWS Documentation Reference:

Lambda Container Images

Question: 309

A company has developed an application that uses AWS Lambda functions to process messages from an Amazon SQS queue. One of the Lambda functions makes a call to an external API that is expected to encounter temporary service unavailability.

A developer needs to configure the function to retry failed messages from an Amazon SQS deadletter queue. The developer notices that the Lambda function is re-processing some messages in the queue more than once.

Which solution will resolve this issue?

A. Set a message retention period for each message. Configure the Lambda function to add a MessageId to each message.

B. Set the visibility timeout parameter at the queue level. Configure the Lambda function to delete processed messages from the queue.

C. Set a receive message wait time for each message. Configure the Lambda function to add a MessageId to each message.

D. Set the delivery delay parameter at the queue level. Configure the Lambda function to delete processed messages from the queue.

Answer: B

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option B is Correct:

Setting the visibility timeout ensures that once a message is being processed, it is temporarily hidden from other consumers. The Lambda function must delete processed messages to avoid re-processing when the visibility timeout expires.

Why Other Options are Incorrect:

Option A: Message retention affects how long messages stay in the queue, not how they are processed.

Option C: Receive message wait time optimizes long polling but does not prevent re-processing.

Option D: Delivery delay introduces latency for new messages and does not address message reprocessing.

AWS Documentation Reference:

Using SQS Visibility Timeout

Question: 310

A company deploys a new application to AWS. The company is streaming application logs to Amazon CloudWatch Logs. The company's development team must receive notification by email when the word "ERROR" appears in any log lines. A developer sets up an Amazon SNS topic and subscribes the development team to the topic.

What should the developer do next to meet the requirements?

A. Select the appropriate log group. Create a CloudWatch metric filter with "ERROR" as the search term. Create an alarm on this metric that notifies the SNS topic when the metric is 1 or higher.

B. In CloudWatch Logs Insights, select the appropriate log group. Create a metric query to search for the term "ERROR" in the

logs. Create an alarm on this metric that notifies the SNS topic when the metric is 1 or higher.

C. Select the appropriate log group. Create an SNS subscription filter with "ERROR" as the filter pattern. Select the SNS topic as the destination.

D. Create a CloudWatch alarm that includes "ERROR" as a filter pattern, a log group dimension that defines the appropriate log group, and a destination that notifies the SNS topic.

Answer: A

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option A is Correct:

Creating a CloudWatch metric filter with "ERROR" as the search term ensures that occurrences of the word "ERROR" in logs are counted. An alarm can then be set to notify the SNS topic when the count exceeds 1. This is the standard approach to monitor specific patterns in CloudWatch Logs.

Why Other Options are Incorrect:

Option B: CloudWatch Logs Insights is used for querying logs manually and does not integrate directly with alarms for automated notifications.

Option C: SNS subscription filters are not a feature for filtering log patterns in CloudWatch Logs.

Option D: CloudWatch alarms do not directly include log filter patterns; a metric filter is needed to create the metric first.

AWS Documentation Reference:

Creating CloudWatch Metric Filters

Question: 311

A company is building a content authoring application. The application has multiple user groups, such as content creator, reviewer, approver, and administrator. The company needs to assign users fine-grained permissions for specific parts of the application.

The company needs a solution to configure, maintain, and analyze user permissions. The company wants a solution that can be easily adapted to work with newer applications in the future. The company must use a third-party OpenID Connect (OIDC) identity provider (IdP) to authenticate users.

A. Configure an Amazon Cognito identity pool for the application. Use the identity pool identities within the application to manage user permissions.

B. Configure the application to check user permissions upon request. Configure the application logic to manage user

permissions.

C. Use Amazon Verified Permissions to set up user permissions. Integrate Verified Permissions with a third-party IdP. Configure the application to request authorization decisions from Verified Permissions.

D. Set up an IAM role for each user group. Assign users appropriate IAM roles. Configure the application to determine appropriate permissions for each user based on the user's IAM role.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option C is Correct:
Amazon Verified Permissions provides fine-grained access control capabilities, making it ideal for managing complex user permissions. It integrates with OIDC IdPs for authentication and allows applications to request authorization decisions dynamically. It is also easily adaptable to newer applications.

Why Other Options are Incorrect:

Option A: Cognito identity pools do not natively support fine-grained permission analysis or management.

Option B: Managing permissions in application logic adds significant operational overhead.

Option D: IAM roles are not designed for application-specific fine-grained access control and are more suitable for resource-level permissions.

AWS Documentation Reference:

Amazon Verified Permissions

Question: 312

A developer runs an application that displays scores for sports games on Amazon EC2 instances. The application uses a Redis client to retrieve the scores from an Amazon ElastiCache (Redis OSS) cluster. The developer observes increased latency during operations on the cache because of connection failures to the cluster. The developer needs to resolve the latency issues.

A. Configure the Redis client to use an exponential backoff retry strategy to establish cache connections.

B. Store the scores in the application's memory. Perform bulk set operations on the scores that are stored in memory.

C. Configure the Redis client in the application to persist connections to the cluster by implementing a connection pool.

D. Deploy more nodes in the ElastiCache cluster. Update the Redis client to discover the new nodes.

Answer: C

Explanation:

Comprehensive Detailed Explanation with all AWS Reference

Why Option C is Correct:

Implementing a connection pool in the Redis client reduces connection overhead and avoids frequent connection establishment, which helps to reduce latency and connection failures.

Why Other Options are Incorrect:

Option A: Exponential backoff retry strategies help with transient failures but do not resolve latency caused by frequent connection establishment.

Option B: Storing scores in application memory adds complexity and risks inconsistency.

Option D: Adding more nodes is unnecessary unless the cluster is under heavy load. Latency due to connection failures is better addressed at the application level.

AWS Documentation Reference:

Amazon ElastiCache Best Practices

Question: 313

A developer is using AWS CodeDeploy to automate a company's application deployments to Amazon EC2.
Which application specification file properties are required to ensure the software deployments do not fail? (Select TWO.)

A. The file must be a JSON-formatted file named appspec.json.
B. The file must be a YAML-formatted file named appspec.yml.
C. The file must be stored in AWS CodeBuild and referenced from the application's source code.
D. The file must be placed in the root of the directory structure of the application's source code.
E. The file must be stored in Amazon S3 and referenced from the application's source code.

Answer: B, D

Explanation:

Comprehensive and Detailed Step-by-Step

To ensure successful software deployments using AWS CodeDeploy, the application specification file (appspec.yml or appspec.json) must adhere to specific requirements:

File Format Requirement (Option B):

The appspec.yml file is a YAML-formatted file required for defining deployment actions and file locations. This is the modern and recommended format for application specification files.

It can also be a JSON-formatted file named appspec.json, but YAML is most commonly used and accepted.

File Placement Requirement (Option D):

The application specification file must reside in the root directory of the application's source code. This is necessary so that AWS CodeDeploy can detect and use the file during deployment.

Incorrect Options:

Option A: While a JSON-formatted file (appspec.json) is valid, this is not a mandatory requirement. YAML is also acceptable, and this option does not account for it.

Option C: The application specification file is not required to be stored in AWS CodeBuild; it must be included in the source code's directory structure.

Option E: The application specification file does not need to be stored in Amazon S3. S3 is commonly used for application artifacts, but the appspec.yml or appspec.json file must exist within the deployment package or source code root.

Reference:

AWS CodeDeploy User Guide: Application Specification File

Question: 314

A developer is creating an ecommerce workflow in an AWS Step Functions state machine that includes a HTTP Task state. The task passes shipping information and order details to an endpoint. The developer needs to test the workflow to confirm that the HTTP headers and body are correct and that the responses meet expectations.

A. Use the TestState API to invoke only the HTTP Task. Set the inspection level to TRACE.
B. Use the TestState API to invoke the state machine. Set the inspection level to DEBUG.
C. Use the data flow simulator to invoke only the HTTP Task. View the request and response data.
D. Change the log level of the state machine to ALL. Run the state machine.

_____ Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

To confirm that the HTTP headers, body, and responses meet expectations, you need to test the specific HTTP Task state in isolation and inspect the details.

Option A: TestState API with TRACE:

The TestState API allows developers to test individual states in a state machine without executing the entire workflow.

Setting the inspection level to TRACE provides detailed information about the HTTP request and response, including headers, body, and status codes.

This option provides the precise and granular testing required to verify the HTTP Task functionality.

Why Other Options Are Incorrect:

Option B: The DEBUG inspection level provides less detailed information than TRACE and focuses on general debugging, not a detailed view of HTTP interactions.

Option C: Step Functions does not have a "data flow simulator" to test individual tasks; this option is not valid.

Option D: Changing the state machine's log level to ALL increases logging granularity for the entire state machine but does not allow isolated testing of a specific HTTP Task.

Reference:

AWS Step Functions: Testing State Machines

Question: 315

A bookstore has an ecommerce website that stores order information in an Amazon DynamoDB table named BookOrders. The DynamoDB table contains approximately one million records.

The table uses OrderID as a partition key. There are no other indexes.

A developer wants to build a new reporting feature to retrieve all records from the table for a specified customer, based on a CustomerID property.

A. Create a DynamoDB global secondary index (GSI) on the table. Use CustomerID as the partition key. Use the specified CustomerID value to run a query on the table.

B. Create a DynamoDB global secondary index (GSI) on the table. Use CustomerID as the sort key. Use a filter expression to perform a scan operation on the table to match on the specified CustomerID value.

C. Create a DynamoDB local secondary index (LSI) on the table. Use CustomerID as the sort key. Run a PartiQL query on the table with a SELECT statement where CustomerID equals the specified CustomerID value.

D. Create a DynamoDB local secondary index (LSI) on the table. Use CustomerID as the partition key.

Use the specified CustomerID value to run a query on the table.

Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

The requirement is to query records by CustomerID, which is not the current partition key (OrderID). To achieve this efficiently:

Option A: Create a GSI with CustomerID as the Partition Key:

A Global Secondary Index (GSI) allows developers to create a different partition key and optional sort key for querying the data.

By creating a GSI with CustomerID as the partition key, the developer can query the table efficiently using CustomerID as the

primary lookup key.

This avoids scanning the entire table and matches the requirement.

Why Other Options Are Incorrect:

Option B: Using CustomerID as a sort key for the GSI and performing a scan operation is inefficient. Queries are optimized, but scans are not.

Option C and D: Local Secondary Indexes (LSI) are only valid when the partition key remains the same as the base table. Since OrderID is the base table's partition key, using CustomerID as the partition key or sort key in an LSI is not valid.

Reference:

Amazon DynamoDB Documentation: GSIs

Question: 316

A company has many microservices that are comprised of AWS Lambda functions. Multiple teams within the company split ownership of the microservices.

An application reads configuration values from environment variables that are contained in the Lambda functions. During a security audit, the company discovers that some of the environment variables contain sensitive information.

The company's security policy requires each team to have full control over the rotation of AWS KMS keys that the team uses for its respective microservices.

A. Create AWS managed keys for all Lambda functions. Use the new AWS managed keys to encrypt the environment variables. Add kms:Decrypt permissions to the Lambda function execution roles.

B. Create customer managed keys for all Lambda functions. Use the new customer managed keys to encrypt the environment variables. Add kms:Decrypt permission to the Lambda function execution roles.

C. Create customer managed keys for all Lambda functions. Use the new customer managed keys to encrypt the environment variables. Add kms:CreateGrant permission and kms:Encrypt permission to the Lambda function execution roles.

D. Create AWS managed keys for all Lambda functions. Use the new AWS managed keys to encrypt the environment variables. Add kms:CreateGrant permission and kms:Encrypt permission to the Lambda function execution roles.

Answer: B

Explanation:

Comprehensive and Detailed Step-by-Step

Customer Managed Keys (CMK) for Granular Control (Option B):

Customer-managed KMS keys are required to meet the security policy requirement of team-specific control over KMS key rotation. Each team can manage the lifecycle of its own key.

The kms:Decrypt permission allows the Lambda function execution roles to decrypt the environment variables during

runtime.

This solution adheres to the principle of least privilege and satisfies the need for team-specific key control.

Why Other Options Are Incorrect:

Option A: AWS-managed keys cannot provide team-specific control or support the custom rotation policy required by the teams.

Option C: Adding kms:CreateGrant and kms:Encrypt permissions to Lambda roles is unnecessary for this scenario. The key usage is limited to decryption at runtime.

Option D: AWS-managed keys still lack team-specific control, and adding kms:CreateGrant and kms:Encrypt is redundant.

Reference:

AWS Lambda Environment Variables

AWS Key Management Service Documentation

Question: 317

A company runs an AWS CodeBuild project on medium-sized Amazon EC2 instances. The company wants to cost optimize the project and reduce the provisioning time.

A. Configure the project to run on a CodeBuild reserved capacity fleet.

B. Select AWS Lambda as the compute mode for the CodeBuild project.

C. Configure the project to run on a CodeBuild on-demand fleet.

D. Set up Amazon S3 caching for the CodeBuild project.

Answer: D

Explanation:

Comprehensive and Detailed Step-by-Step

Option D: Set up Amazon S3 Caching for CodeBuild:

CodeBuild supports S3 caching to store intermediate build artifacts and dependencies. This reduces the time required to download dependencies during subsequent builds, effectively lowering costs and improving build performance.

By using S3 caching, developers can optimize costs without changing the compute type or adding complexity.

Why Other Options Are Incorrect:

Option A: CodeBuild does not have a "reserved capacity fleet" option.

Option B: AWS Lambda cannot be used as the compute mode for CodeBuild projects. CodeBuild uses its own managed build

environments.

Option C: CodeBuild already operates on an on-demand basis, so this does not address the need for optimization or reduced provisioning time.

Reference:

AWS CodeBuild Caching Documentation

Question: 318

A developer needs to set up an API to provide access to an application and its resources. The developer has a TLS certificate. The developer must have the ability to change the default base URL of the API to a custom domain name. The API users are distributed globally. The solution must minimize API latency.

A. Create an Amazon CloudFront distribution that uses an AWS Lambda@Edge function to process API requests. Import the TLS certificate into AWS Certificate Manager and CloudFront. Add the custom domain name as an alias resource record set that is for the CloudFront distribution.

B. Create an Amazon API Gateway REST API. Use the private endpoint type. Import the TLS certificate into AWS Certificate Manager. Create a custom domain name for the REST API. Route traffic to the custom domain name. Disable the default endpoint for the REST API.

C. Create an Amazon API Gateway REST API. Use the edge-optimized endpoint type. Import the TLS certificate into AWS Certificate Manager. Create a custom domain name for the REST API. Route traffic to the custom domain name. Disable the default endpoint for the REST API.

D. Create an Amazon CloudFront distribution that uses CloudFront Functions to process API requests. Import the TLS certificate into AWS Certificate Manager and CloudFront. Add the custom domain name as an alias resource record set that is for the CloudFront distribution.

Answer: C

Explanation:

Comprehensive and Detailed Step-by-Step

Option C: Edge-Optimized API Gateway with Custom Domain Name:

Edge-Optimized API Gateway: This endpoint type automatically leverages the Amazon CloudFront global distribution network, minimizing latency for API users distributed globally.

Custom Domain Name: API Gateway supports custom domain names for APIs. Importing the TLS certificate into AWS Certificate Manager (ACM) and associating it with the custom domain name ensures secure connections.

Disabling the Default Endpoint: Prevents direct access via the default API Gateway URL, enforcing the use of the custom domain name.

Why Other Options Are Incorrect:

Option A: While CloudFront can distribute API requests globally, API Gateway with edge-optimized endpoints already provides this functionality natively without requiring Lambda@Edge.

Option B: Private endpoint types are used for internal access via VPC, which does not meet the global distribution and low-latency requirement.

Option D: CloudFront Functions are not needed because API Gateway's edge-optimized endpoints handle global distribution efficiently.

Reference:

Amazon API Gateway Custom Domain Names

Amazon API Gateway Endpoint Types

Question: 319

A developer used the AWS SDK to create an application that aggregates and produces log records for 10 services. The application delivers data to an Amazon Kinesis Data Streams stream.

Each record contains a log message with a service name, creation timestamp, and other log information. The stream has 15 shards in provisioned capacity mode. The stream uses service name as the partition key.

The developer notices that when all the services are producing logs, ProvisionedThroughputExceededException errors occur during PutRecord requests. The stream metrics show that the write capacity the applications use is below the provisioned capacity.

A. Change the capacity mode from provisioned to on-demand.
B. Double the number of shards until the throttling errors stop occurring.
C. Change the partition key from service name to creation timestamp.
D. Use a separate Kinesis stream for each service to generate the logs.

Answer: C

Explanation:

Comprehensive and Detailed Step-by-Step

Issue Analysis:

The stream uses service name as the partition key. This can cause "hot partition" issues when a few service names generate significantly more logs compared to others, causing uneven distribution of data across shards.

Metrics show that the write capacity used is below provisioned capacity, which confirms that the throughput errors are due to shard-level limits and not overall capacity.

Option C: Change Partition Key to Creation Timestamp:

By changing the partition key to the creation timestamp (or a composite key including timestamp), the distribution of data across shards can be randomized, ensuring an even spread of records.

This resolves the shard overutilization issue and eliminates ProvisionedThroughputExceededException.

Why Other Options Are Incorrect:

Option A: Switching to on-demand capacity mode might temporarily alleviate the issue, but the root cause (hot partitioning) remains unresolved.

Option B: Adding shards increases capacity but does not fix the skewed data distribution caused by using the service name as the partition key.

Option D: Creating separate streams for each service adds unnecessary complexity and does not scale well as the number of services grows.

Reference:

Best Practices for Kinesis Data Streams Partition Key Design

Question: 320

A development team is creating a serverless application that uses AWS Lambda functions. The team wants to streamline a testing workflow by sharing test events across multiple developers within the same AWS account. The team wants to ensure all developers can use consistent test events without compromising security.

A. Export test events as JSON files. Store the files in an Amazon S3 bucket. Configure granular IAM permissions to allow the developers to access the S3 bucket.

B. Store test events in an Amazon DynamoDB table. Create an AWS Lambda function to retrieve shared test events for the developers.

C. Configure test events to be shareable. Configure granular IAM permissions to allow the developers to access shared test events.

D. Set up a Git repository to store test events. Provide the developers with access to the repository.

Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

Option A: Use Amazon S3 for Shared Test Events:

Storing JSON test event files in an S3 bucket provides a centralized, cost-effective, and highly available solution.

Granular IAM policies can restrict access to specific developers or roles, ensuring security while maintaining consistency for

shared test events.

This solution has minimal operational overhead and integrates easily with existing workflows.

Why Other Options Are Incorrect:

Option B: Using DynamoDB and a Lambda function introduces unnecessary complexity for a relatively simple requirement. S3 provides a simpler and more cost-efficient solution.

Option C: AWS Lambda test events are not inherently shareable across developers, making this option invalid.

Option D: Using a Git repository adds operational overhead and requires developers to clone/update repositories for access, which is more cumbersome compared to S3.

Reference:

Amazon S3 for Centralized Storage

Question: 321

A company has a serverless application that uses Amazon API Gateway and AWS Lambda functions to expose a RESTful API. The company uses a continuous integration and continuous delivery (CI/CD) workflow to deploy the application to multiple environments. The company wants to implement automated integration tests after deployment.

A developer needs to set up the necessary infrastructure and processes to automate the deployment and integration tests for the serverless application.

A. Configure API Gateway stages to represent each application environment. Use AWS SAM templates to manage the infrastructure for the Lambda functions and API resources. Use AWS CodeBuild to implement automated deployment tests to validate the deployments in each stage.

B. Configure API Gateway stages to represent each application environment. Use AWS CloudFormation to manage the infrastructure for the Lambda functions and API resources. Use AWS CodeBuild to implement automated deployment tests to validate the deployments in each stage.

C. Use AWS CodePipeline to create a CI/CD pipeline. Configure API Gateway stages to represent each application environment. Use AWS CloudFormation templates to manage the infrastructure for the Lambda functions and API resources. Use AWS CodeBuild to implement automated deployment tests to validate the deployments in each stage.

D. Use AWS CloudFormation to create and deploy the application infrastructure in each application environment. Use the AWS CLI to invoke Lambda functions to perform deployment tests after each deployment.

Answer: C

Explanation:

Comprehensive and Detailed Step-by-Step

Option C: Use AWS CodePipeline for CI/CD Workflow:

AWS CodePipeline automates the entire CI/CD pipeline, including build, deploy, and test stages. This minimizes manual effort and integrates well with AWS services.

API Gateway Stages: Represent different environments, such as dev, test, and prod, allowing isolated deployment and testing.

AWS CloudFormation Templates: Ensure that the infrastructure for Lambda and API Gateway is consistent across environments.

AWS CodeBuild for Automated Tests: Validates the deployments in each stage, ensuring integration and functionality are tested post-deployment.

Why Other Options Are Incorrect:

Option A and B: While using AWS SAM or CloudFormation for infrastructure management is valid, these options lack the fully automated CI/CD pipeline provided by CodePipeline.

Option D: Manually invoking Lambda functions using the AWS CLI introduces operational overhead and lacks the automation provided by CodePipeline.

Reference:

AWS CodePipeline Documentation

API Gateway Stages for CI/CD

Question: 322

A company needs to package and deploy an application that uses AWS Lambda to compress and decompress video clips. The application uses a video codec library that is larger than 250 MB. The application uses the library to compress the videos before storage and to decompress the videos upon retrieval.

A. Create one Lambda function. Upload one zip file that contains code to handle video compression and decompression to the function. Include the codec library in the zip file.

B. Create two Lambda functions. Upload one zip file that contains code to handle video compression to one function. Upload a second zip file that contains code for video decompression to the second function. Include the codec library in both zip files.

C. Create two Lambda functions. Upload one zip file that contains code to handle video compression to one function. Upload a second zip file that contains code for video decompression to the second function. Create one Lambda layer for the codec library. Add the layer to both functions.

D. Create two Lambda functions. Build one container image that contains code to handle video compression and a second image that contains video decompression code. Add the codec library to both images. Upload the images to Amazon ECR. Use the containers to create the Lambda functions.

Answer: D

Explanation:

Comprehensive and Detailed Step-by-Step

Option D: Use Lambda with Container Images

AWS Lambda supports container images up to 10 GB in size, making it suitable for applications with large dependencies, such as a video codec library larger than 250 MB.

By creating separate container images for video compression and decompression, the application can efficiently isolate functionality while ensuring that each function includes the required dependencies.

The container images are stored in Amazon ECR and used to create the Lambda functions.

Why Other Options Are Incorrect:

Option A: A single Lambda function with all functionalities and dependencies in one zip file is not feasible due to the 250 MB deployment package size limit for zip files.

Option B: Including the library in two separate zip files still exceeds the size limit for Lambda zip deployment packages.

Option C: While using a Lambda layer can reduce redundancy, the combined size of the layer and the zip files would exceed the limit of 250 MB.

Reference:

Using Container Images with AWS Lambda

Question: 323

A developer needs to give a new application the ability to retrieve configuration data.

The application must be able to retrieve new configuration data values without the need to redeploy the application code. If the application becomes unhealthy because of a bad configuration change, the developer must be able to automatically revert the configuration change to the previous value.

A. Use AWS Secrets Manager to manage and store the configuration data. Integrate Secrets Manager with a custom AWS Config rule that has remediation actions to track changes in the application and to roll back any bad configuration changes.

B. Use AWS Secrets Manager to manage and store the configuration data. Integrate Secrets Manager with a custom AWS Config rule. Attach a custom AWS Systems Manager document to the rule that automatically rolls back any bad configuration changes.

C. Use AWS AppConfig to manage and store the configuration data. Integrate AWS AppConfig with Amazon CloudWatch to monitor changes to the application. Set up an alarm to automatically roll back any bad configuration changes.

D. Use AWS AppConfig to manage and store the configuration data. Integrate AWS AppConfig with Amazon CloudWatch to monitor changes to the application. Set up CloudWatch Application Signals to roll back any bad configuration changes.

Answer: D

Explanation:

Comprehensive and Detailed Step-by-Step

Option D: AWS AppConfig with CloudWatch Application Signals

AWS AppConfig is designed for managing and deploying application configurations dynamically, without redeployment.

CloudWatch Application Signals provide automatic rollback mechanisms in case of an unhealthy application state due to bad configuration changes.

This solution meets the requirements with minimal operational overhead by ensuring both dynamic updates and rollback functionality.

Why Other Options Are Incorrect:

Option A and B: AWS Secrets Manager is designed for secrets management, not dynamic application configuration. Custom Config rules add unnecessary complexity.

Option C: While CloudWatch alarms can monitor application changes, using alarms for rollback requires manual setup and lacks the automatic rollback provided by Application Signals.

Reference:

AWS AppConfig Documentation

Question: 324

A developer is building a web and mobile application for two types of users: regular users and guest users. Regular users are required to log in, but guest users do not log in. Users should see only their data, regardless of whether they authenticate. Users need AWS credentials before they can access AWS resources.

A. Use an Amazon Cognito identity pool to generate temporary AWS credentials that are linked to an unauthenticated role that has access to the required resources.

B. Set up an IAM user that has permissions to the required resources. Hardcode the IAM credentials in the web and mobile application.

C. Generate temporary keys that are stored in AWS KMS. Use the temporary keys to access the required resources.

D. Generate temporary credentials. Store the temporary credentials in AWS Secrets Manager. Use the temporary credentials to access the required resources.

Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

Option A: Amazon Cognito Identity Pool with Unauthenticated Role

Cognito identity pools can generate temporary AWS credentials for both authenticated and unauthenticated users.

For guest users, Cognito assigns an unauthenticated role with limited permissions, ensuring secure access to only their resources.

This is the most secure and efficient solution for managing AWS credentials dynamically without hardcoding or storing them.

Why Other Options Are Incorrect:

Option B: Hardcoding IAM credentials in the application is insecure and violates best practices.

Option C and D: Temporary keys stored in KMS or Secrets Manager require additional implementation overhead and do not inherently manage user-specific access.

Reference:

Amazon Cognito Identity Pools

Question: 325

A developer maintains a serverless application that uses an Amazon API Gateway REST API to invoke an AWS Lambda function by using a non-proxy integration. The Lambda function returns data, which is stored in Amazon DynamoDB.

Several application users begin to receive intermittent errors from the API. The developer examines Amazon CloudWatch Logs for the Lambda function and discovers several ProvisionedThroughputExceededException errors.

The developer needs to resolve the errors and ensure that the errors do not reoccur.

A. Use provisioned capacity mode for the DynamoDB table, and assign sufficient capacity units. Configure the Lambda function to retry requests with exponential backoff.

B. Update the REST API to send requests on an Amazon SQS queue. Configure the Lambda function to process requests from the queue.

C. Configure a usage plan for the REST API.

D. Update the REST API to invoke the Lambda function asynchronously.

Answer: A

Explanation:

Comprehensive and Detailed Step-by-Step

Option A: Provisioned Capacity with Exponential Backoff:

Using provisioned capacity ensures sufficient throughput for the DynamoDB table.

Configuring the Lambda function to implement exponential backoff retries reduces the chance of exceeding capacity during peak usage.

This combination addresses the root cause (ProvisionedThroughputExceededException) and prevents errors without overprovisioning.

Why Other Options Are Incorrect:

Option B: Using SQS adds unnecessary latency and complexity. The issue lies in DynamoDB throughput, not request management.

Option C: A usage plan for the API does not address throughput issues in DynamoDB.

Option D: Invoking the Lambda function asynchronously does not resolve the DynamoDB capacity issue and might lead to delayed processing.

Reference:

DynamoDB Provisioned Throughput Documentation

Question: 326

A company has an application that consists of different microservices that run inside an AWS account. The microservices are running in containers inside a single VPC. The number of microservices is constantly increasing. A developer must create a central logging solution for application logs.

A. Create a different Amazon CloudWatch Logs stream for each microservice.
B. Create an AWS CloudTrail trail to log all the API calls.
C. Configure VPC Flow Logs to track the communications between the microservices.
D. Use AWS Cloud Map to map the interactions of the microservices.

Answer: A

Explanation:

To create a central logging solution for microservices, using Amazon CloudWatch Logs is a recommended and effective approach. Here's why:

Amazon CloudWatch Logs Streams allow you to centralize logs from different services, which is crucial as the number of microservices increases.

Each microservice can have its own dedicated log stream within Amazon CloudWatch Logs, providing clear segregation of logs while still allowing centralized management.
This setup enables developers to monitor, search, and analyze logs efficiently using tools like CloudWatch Insights.

Other options like CloudTrail (B) are designed for API activity monitoring, not application logs. VPC Flow Logs (C) focus on network traffic rather than application behavior. AWS Cloud Map (D) is for service discovery and routing, not logging.
Reference:
AWS CloudWatch Logs documentation

Question: 327

A company has a website that is developed in PHP and is launched using AWS Elastic Beanstalk. There is a new version of the website that needs to be deployed in the Elastic Beanstalk environment. The company cannot tolerate having the website ofline if an update fails. Deployments must have minimal impact and rollback as soon as possible.

A. All at once
B. Rolling
C. Snapshots
D. Immutable

Answer: D

Explanation:

The Immutable deployment method is the best choice when a company requires minimal downtime and automatic rollback in case of a failure. Here's why:
In Immutable deployments, a new set of instances is launched with the updated version. These instances are tested and validated before they replace the old instances. This ensures zero downtime and immediate rollback if the deployment fails.
All at once (A) causes downtime because the update replaces all instances simultaneously.
Rolling deployments (B) update a few instances at a time, but if a failure occurs midway, downtime or partial unavailability can happen.
Snapshots (C) are not a deployment strategy in Elastic Beanstalk.
Reference:
Elastic Beanstalk Deployment Policies

Question: 328

A gaming application stores scores for players in an Amazon DynamoDB table that has four attributes: user_id, user_name, user_score, and user_rank. The users are allowed to update their names only. A user is authenticated by web identity federation.

Which set of conditions should be added in the policy attached to the role for the dynamodb:PutItem API call?
A.

"Condition": {
"ForAllValues:StringEquals": {

"dynamodb:LeadingKeys": ["${www.amazon.com:user_id}"],
"dynamodb:Attributes": ["user_name"]
}
}
B.

"Condition": {
"ForAllValues:StringEquals": {
"dynamodb:LeadingKeys": ["${www.amazon.com:user_name}"],
"dynamodb:Attributes": ["user_id"]
}
}
C.

"Condition": {
"ForAllValues:StringEquals": {
"dynamodb:LeadingKeys": ["${www.amazon.com:user_id}"],
"dynamodb:Attributes": ["user_name", "user_id"]
}
}
D.

"Condition": {
"ForAllValues:StringEquals": {
"dynamodb:LeadingKeys": ["${www.amazon.com:user_name}"],
"dynamodb:Attributes": ["username", "userid"]
}
}

_____ Answer: A

Explanation:

The correct policy condition ensures that:
The LeadingKeys condition restricts operations to the authenticated user's user_id.
The Attributes condition limits the updatable attributes to user_name.
Explanation of Choices:
Option A: Correctly enforces both the key restriction (dynamodb:LeadingKeys) and ensures only the user_name attribute can be updated.
Option B, C, D: Use incorrect conditions, such as referencing user_name in the LeadingKeys or including other attributes like user_id in updatable fields.
Reference:
AWS DynamoDB Condition Keys Documentation

Question: 329 _____

A developer is migrating a containerized application from an on-premises environment to the AWS Cloud. The developer is using the AWS CDK to provision a container in Amazon ECS on AWS Fargate. The container is behind an Application Load Balancer (ALB).
When the developer deploys the stack, the deployment fails because the ALB fails health checks. The developer needs to resolve the failed health checks.
Which solutions will meet this requirement? (Select TWO.)

A. Confirm that the capacity providers for the container have been provisioned and are properly sized.

B. Confirm that the target group port matches the port mappings in the ECS task definition.

C. Confirm that a hosted zone associated with the ALB matches a hosted zone that is referenced in the ECS task definition.

D. Confirm that the ALB listener on the mapped port has a default action that redirects to the application's health check path endpoint.

E. Confirm that the ALB listener on the mapped port has a default action that forwards to the correct target group.

Answer: B, E

Explanation:

Option B: The target group port in the ALB must match the port specified in the ECS task definition. If there is a mismatch, the ALB health check will fail since it cannot correctly route traffic to the container.

Option E: The ALB listener must have a default action that forwards requests to the correct target group associated with the ECS service. If this configuration is missing, the health check will fail as no traffic is routed to the service.

Option A is irrelevant to resolving health check issues since capacity providers relate to provisioning compute capacity.

Option C (hosted zone) is not directly related to ALB health checks.

Option D (redirecting traffic) is not related to ECS health check configurations.

Reference:

AWS ECS Health Check Documentation

Question: 330

A developer is managing an application that uploads user files to an Amazon S3 bucket named companybucket. The company wants to maintain copies of all the files uploaded by users for compliance purposes, while ensuring users still have access to the data through the application. Which IAM permissions should be applied to users to ensure they can create but not remove files from the bucket?

A.

json

Copy code

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "statement1",
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:PutObject", "s3:DeleteObject"],
    "Resource": ["arn:aws:s3:::companybucket"]
  }
]
}
```

B.

json

Copy code

```
{
"Version": "2012-10-17",
"Statement": [
  {
```

```
    "Sid": "statement1",
    "Effect": "Allow",
    "Action": ["s3:CreateBucket", "s3:GetBucketLocation"],
    "Resource": "arn:aws:s3:::companybucket"
  }
]
}
```

C.

json

Copy code

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "statement1",
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:PutObject", "s3:DeleteObject", "s3:PutObjectRetention"],
    "Resource": "arn:aws:s3:::companybucket"
  }
]
}
```

D.

json

Copy code

```
{
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "statement1",
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:PutObject"],
    "Resource": ["arn:aws:s3:::companybucket"]
  }
]
}
```

_____ Answer: D

Explanation:

To meet the requirement:
Users must be able to upload (PutObject) and read (GetObject) files but not delete them.
Option D ensures users cannot delete files by omitting the s3:DeleteObject action while allowing s3:GetObject and s3:PutObject.
Option A: Includes s3:DeleteObject, which allows users to delete files and does not meet the requirement.
Option B: Contains unrelated actions like CreateBucket, which is not relevant here.
Option C: Adds s3:PutObjectRetention, which is unnecessary and does not restrict DeleteObject. Reference:
AWS S3 Permissions Documentation

Question: 331

A developer is building an application that stores objects in an Amazon S3 bucket. The bucket does not have versioning enabled. The objects are accessed rarely after 1 week. However, the objects must be immediately available at all times. The developer wants to optimize storage costs for the S3 bucket.

Which solution will meet this requirement?

A. Create an S3 Lifecycle rule to expire objects after 7 days.

B. Create an S3 Lifecycle rule to transition objects to S3 Standard-Infrequent Access (S3 Standard-IA) after 7 days.

C. Create an S3 Lifecycle rule to transition objects to S3 Glacier Flexible Retrieval after 7 days.

D. Create an S3 Lifecycle rule to delete objects that have delete markers.

Answer: B

Explanation:

Comprehensive Detailed and Lengthy Step-by-Step Explanation with All AWS Developer Reference:

1. Understanding the Use Case:

The goal is to store objects in an S3 bucket while optimizing storage costs. The key conditions are: Objects are accessed infrequently after 1 week.

Objects must remain immediately accessible at all times.

2. AWS S3 Storage Classes Overview:

Amazon S3 offers various storage classes, each optimized for specific use cases:

S3 Standard: Best for frequently accessed data with low latency and high throughput needs.

S3 Standard-Infrequent Access (S3 Standard-IA): Optimized for infrequently accessed data but requires the same availability and immediate access as Standard storage. It provides lower storage costs but incurs retrieval charges.

S3 Glacier Flexible Retrieval (formerly S3 Glacier): Designed for archival data with retrieval latency ranging from minutes to hours. This does not meet the requirement for "immediate access."

S3 Glacier Deep Archive: Lowest-cost storage, suitable for rarely accessed data with retrieval times of hours.

3. Explanation of the Options:

Option A:

"Create an S3 Lifecycle rule to expire objects after 7 days."

Expiring objects after 7 days deletes them permanently, which does not fulfill the requirement of retaining the objects for later infrequent access.

Option B:

"Create an S3 Lifecycle rule to transition objects to S3 Standard-Infrequent Access (S3 Standard-IA) after 7 days."

This is the correct solution. S3 Standard-IA is ideal for objects accessed infrequently but still need to be available immediately. Transitioning objects to this storage class reduces storage costs while maintaining availability and low latency.

Option C:

"Create an S3 Lifecycle rule to transition objects to S3 Glacier Flexible Retrieval after 7 days."

S3 Glacier Flexible Retrieval is a low-cost archival solution. However, it does not provide immediate access as retrieval requires minutes to hours. This option does not meet the requirement.

Option D:

"Create an S3 Lifecycle rule to delete objects that have delete markers."

This option is irrelevant to the given use case, as it addresses versioning cleanup, which is not enabled in the described S3 bucket.

4. Implementation Steps for Option B:

To transition objects to S3 Standard-IA after 7 days:

Navigate to the S3 Console:

Sign in to the AWS Management Console and open the S3 service.

Select the Target Bucket:

Choose the bucket where the objects are stored.

Set Up a Lifecycle Rule:

Go to the Management tab.

Under Lifecycle Rules, click Create lifecycle rule.

Define the Rule Name and Scope:

Provide a descriptive name for the rule.

Specify whether the rule applies to the entire bucket or a subset of objects (using a prefix or tag filter).

Configure Transitions:

Choose Add transition.

Specify that objects should transition to S3 Standard-IA after 7 days.

Review and Save the Rule:

Review the rule configuration and click Save.

5. Cost Optimization Benefits:

Transitioning to S3 Standard-IA results in cost savings as it offers:

Lower storage costs compared to S3 Standard.

Immediate access to objects when required.

However, remember that there is a retrieval cost associated with S3 Standard-IA, so it is best suited for data with low retrieval frequency.

Reference:

Amazon S3 Lifecycle Configuration Guide

Amazon S3 Storage Classes

AWS S3 Pricing

AWS Documentation on S3 Standard-IA

Question: 332

A developer needs to export the contents of several Amazon DynamoDB tables into Amazon S3 buckets to comply with company data regulations. The developer uses the AWS CLI to run commands to export from each table to the proper S3 bucket. The developer sets up AWS credentials correctly and grants resources appropriate permissions. However, the exports of some tables fail.

What should the developer do to resolve this issue?

A. Ensure that point-in-time recovery is enabled on the DynamoDB tables.

B. Ensure that the target S3 bucket is in the same AWS Region as the DynamoDB table.

C. Ensure that DynamoDB streaming is enabled for the tables.

D. Ensure that DynamoDB Accelerator (DAX) is enabled.

Answer: B

Explanation:

Comprehensive Detailed and Lengthy Step-by-Step Explanation with All AWS Developer Reference:

1. Understanding the Use Case:

The developer needs to export DynamoDB table data into Amazon S3 buckets using the AWS CLI, and some exports are failing. Proper credentials and permissions have already been configured.

2. Key Conditions to Check:

Region Consistency:

DynamoDB exports require that the target S3 bucket and the DynamoDB table reside in the same AWS Region. If they are not in the same Region, the export process will fail.

Point-in-Time Recovery (PITR):

PITR is not required for exporting data from DynamoDB to S3. Enabling PITR allows recovery of table states at specific points in time but does not directly influence export functionality.

DynamoDB Streams:

Streams allow real-time capture of data modifications but are unrelated to the bulk export feature. DAX (DynamoDB Accelerator):

DAX is a caching service that speeds up read operations for DynamoDB but does not affect the export functionality.

3. Explanation of the Options:

Option A:

"Ensure that point-in-time recovery is enabled on the DynamoDB tables."

While PITR is useful for disaster recovery and restoring table states, it is not required for exporting data to S3. This option does not address the export failure.

Option B:

"Ensure that the target S3 bucket is in the same AWS Region as the DynamoDB table."

This is the correct answer. DynamoDB export functionality requires the target S3 bucket to reside in the same AWS Region as the DynamoDB table. If the S3 bucket is in a different Region, the export will fail.

Option C:

"Ensure that DynamoDB streaming is enabled for the tables."

Streams are useful for capturing real-time changes in DynamoDB tables but are unrelated to the export functionality. This option does not resolve the issue.

Option D:

"Ensure that DynamoDB Accelerator (DAX) is enabled."

DAX accelerates read operations but does not influence the export functionality. This option is irrelevant to the issue.

4. Resolution Steps:

To ensure successful exports:

Verify the Region of the DynamoDB tables:

Check the Region where each table is located.

Verify the Region of the target S3 buckets:

Confirm that the target S3 bucket for each export is in the same Region as the corresponding DynamoDB table.

If necessary, create new S3 buckets in the appropriate Regions.

Run the export command again with the correct setup:

```
aws dynamodb export-table-to-point-in-time \
--table-name <TableName> \
--s3-bucket <BucketName> \
--s3-prefix <Prefix> \
--export-time <ExportTime> \
--region <Region>
```

Reference:

Exporting DynamoDB Data to Amazon S3

S3 Bucket Region Requirements for DynamoDB Exports

AWS CLI Reference for DynamoDB Export

Question: 333

A developer is creating an application that must be able to generate API responses without backend integrations. Multiple internal teams need to work with the API while the application is still in development.
Which solution will meet these requirements with the LEAST operational overhead?

A. Create an Amazon API Gateway REST API. Set up a proxy resource that has the HTTP proxy integration type.
B. Create an Amazon API Gateway HTTP API. Provision a VPC link, and set up a private integration on the API to connect to a VPC.
C. Create an Amazon API Gateway HTTP API. Enable mock integration on the method of the API resource.
D. Create an Amazon API Gateway REST API. Enable mock integration on the method of the API resource.

Answer: D

Explanation:

Comprehensive Detailed and Lengthy Step-by-Step Explanation with All AWS Developer Reference:

1. Understanding the Use Case:

The API needs to:

Generate responses without backend integrations: This indicates the use of mock responses for testing.
Be used by multiple internal teams during development.
Minimize operational overhead.

2. Key Features of Amazon API Gateway:

REST APIs: Fully managed API Gateway option that supports advanced capabilities like mock integrations, request/response transformation, and more.

HTTP APIs: Lightweight option for building APIs quickly. It supports fewer features but has lower operational complexity and cost.

Mock Integration: Allows API Gateway to return pre-defined responses without requiring backend integration.

3. Explanation of the Options:

Option A:

"Create an Amazon API Gateway REST API. Set up a proxy resource that has the HTTP proxy integration type."

A proxy integration requires a backend service for handling requests. This does not meet the requirement of "no backend integrations."

Option B:

"Create an Amazon API Gateway HTTP API. Provision a VPC link, and set up a private integration on the API to connect to a VPC."

This requires setting up a VPC and provisioning resources, which increases operational overhead and is unnecessary for this use case.

Option C:

"Create an Amazon API Gateway HTTP API. Enable mock integration on the method of the API resource."

While HTTP APIs can enable mock integrations, they have limited support for advanced features compared to REST APIs, such as detailed request/response customization. REST APIs are better suited for development environments requiring mock responses.

Option D:

"Create an Amazon API Gateway REST API. Enable mock integration on the method of the API resource."

This is the correct answer. REST APIs with mock integration allow defining pre-configured responses directly within API Gateway, making them ideal for scenarios where backend services are unavailable. It provides flexibility for testing while minimizing operational overhead.

4. Implementation Steps:

To enable mock integration with REST API:

Create a REST API in API Gateway:

Open the API Gateway Console.

Choose Create API > REST API.

Define the API Resource and Methods:

Add a resource and method (e.g., GET or POST).

Set Up Mock Integration:

Select the method, and in the Integration Type, choose Mock Integration.

Configure the Mock Response:

Define a 200 OK response with the desired response body and headers.

Deploy the API:

Deploy the API to a stage (e.g., dev) to make it accessible.

5. Why REST API Over HTTP API?

REST APIs support detailed request/response transformations and robust mock integration features, which are ideal for development and testing scenarios.

While HTTP APIs offer lower cost and simplicity, they lack some advanced features required for finetuned mock integrations.

Reference:

Amazon API Gateway REST API Features

Mock Integration in API Gateway

Comparison of REST and HTTP APIs in API Gateway

AWS API Gateway Pricing

Question: 334

A company wants to use AWS AppConfig to gradually deploy a new feature to 15% of users to test the feature before a full deployment.

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

A. Set up a custom script within the application to randomly select 15% of users. Assign a flag for the new feature to the selected users.

B. Create separate AWS AppConfig feature flags for both groups of users. Configure the flags to target 15% of users.

C. Create an AWS AppConfig feature flag. Define a variant for the new feature, and create a rule to target 15% of users.

D. Use AWS AppConfig to create a feature flag without variants. Implement a custom traffic splitting mechanism in the application code.

Answer: C

Explanation:

Comprehensive Detailed and Lengthy Step-by-Step Explanation with All AWS Developer Reference:

1. Understanding the Use Case:

The company wants to gradually release a new feature to 15% of users to perform testing. AWS AppConfig is designed to manage and deploy configurations, including feature flags, allowing controlled rollouts.

2. Key AWS AppConfig Features:

Feature Flags: Enable or disable features dynamically without redeploying code.

Variants: Define different configurations for subsets of users.

Targeting Rules: Specify rules for which users receive a particular variant.

3. Explanation of the Options:

Option A:

"Set up a custom script within the application to randomly select 15% of users. Assign a flag for the new feature to the selected users."

While possible, this approach requires significant operational effort to manage user selection and ensure randomness. It does not leverage AWS AppConfig's built-in capabilities, which increases overhead.

Option B:

"Create separate AWS AppConfig feature flags for both groups of users. Configure the flags to target 15% of users."

Creating multiple feature flags for different user groups complicates configuration management and does not optimize the use of AWS AppConfig.

Option C:

"Create an AWS AppConfig feature flag. Define a variant for the new feature, and create a rule to target 15% of users."

This is the correct solution. Using AWS AppConfig feature flags with variants and targeting rules is the most efficient approach. It minimizes operational overhead by leveraging AWS AppConfig's built-in targeting and rollout capabilities.

Option D:

"Use AWS AppConfig to create a feature flag without variants. Implement a custom traffic splitting mechanism in the application code."

This approach requires custom implementation within the application code, increasing complexity and operational effort.

4. Implementation Steps for Option C:

Set Up AWS AppConfig:

Open the AWS Systems Manager Console.

Navigate to AppConfig.

Create a Feature Flag:

Define a new configuration for the feature flag.

Add variants (e.g., "enabled" for the new feature and "disabled" for no change).

Define a Targeting Rule:

Use percentage-based targeting to define a rule that applies the "enabled" variant to 15% of users. Targeting rules can use attributes like user IDs or geographic locations.

Deploy the Configuration:

Deploy the configuration using a controlled rollout to ensure gradual exposure.

Reference:

AWS AppConfig Documentation

Feature Flags in AWS AppConfig

AWS AppConfig Targeting Rules

Question: 335

A developer is building an application to process a stream of customer orders. The application sends processed orders to an Amazon Aurora MySQL database. The application needs to process the orders in batches.

The developer needs to configure a workflow that ensures each record is processed before the application sends each order

to the database.

Options:

A. Use Amazon Kinesis Data Streams to stream the orders. Use an AWS Lambda function to process the orders. Configure an event source mapping for the Lambda function, and set the MaximumBatchingWindowInSeconds setting to 300.

B. Use Amazon SQS to stream the orders. Use an AWS Lambda function to process the orders. Configure an event source mapping for the Lambda function, and set the MaximumBatchingWindowInSeconds setting to 0.

C. Use Amazon Managed Streaming for Apache Kafka (Amazon MSK) to stream the orders. Use an Amazon EC2 instance to process the orders. Configure an event source mapping for the EC2 instance, and increase the payload size limit to 36 MB.

D. Use Amazon DynamoDB Streams to stream the orders. Use an Amazon ECS cluster on AWS Fargate to process the orders. Configure an event source mapping for the cluster, and set the BatchSize setting to 1.

Answer: A

Explanation:

Step 1: Understanding the Problem

Processing in Batches: The application must process records in groups.

Sequential Processing: Each record in the batch must be processed before writing to Aurora. Solution Goals: Use services that support ordered, batched processing and integrate with Aurora. Step 2: Solution Analysis

Option A:

Amazon Kinesis Data Streams supports ordered data processing.

AWS Lambda can process batches of records via event source mapping with MaximumBatchingWindowInSeconds for timing control.

Configuring the batching window ensures efficient processing and compliance with the workflow. Correct Option.

Option B:

Amazon SQS is not designed for streaming; it provides reliable, unordered message delivery. Setting MaximumBatchingWindowInSeconds to 0 disables batching, which is contrary to the requirement.

Not suitable.

Option C:

Amazon MSK provides Kafka-based streaming but requires custom EC2-based processing.

This increases system complexity and operational overhead.

Not ideal for serverless requirements.

Option D:

DynamoDB Streams is event-driven but lacks strong native integration for batch ordering.

Using ECS adds unnecessary complexity.

Not suitable.

Step 3: Implementation Steps for Option A

Set up Kinesis Data Stream:

Configure shards based on the expected throughput.

Configure Lambda with Event Source Mapping:

Enable Kinesis as the event source for Lambda.

Set MaximumBatchingWindowInSeconds to 300 to accumulate data for processing.

Example:

```
{
"EventSourceArn": "arn:aws:kinesis:region:account-id:stream/stream-name",
```

"BatchSize": 100,
"MaximumBatchingWindowInSeconds": 300
}
Write Processed Data to Aurora:
Use AWS RDS Data API for efficient database operations from Lambda.
AWS Developer Reference:
Amazon Kinesis Data Streams Developer Guide
AWS Lambda Event Source Mapping
Batch Processing with Lambda

Question: 336

A social media company is designing a platform that allows users to upload data, which is stored in Amazon S3. Users can upload data encrypted with a public key. The company wants to ensure that only the company can decrypt the uploaded content using an asymmetric encryption key. The data must always be encrypted in transit and at rest.

Options:

A. Use server-side encryption with Amazon S3 managed keys (SSE-S3) to encrypt the data.
B. Use server-side encryption with customer-provided encryption keys (SSE-C) to encrypt the data.
C. Use client-side encryption with a data key to encrypt the data.
D. Use client-side encryption with a customer-managed encryption key to encrypt the data.

Answer: D

Explanation:

Step 1: Problem Understanding
Asymmetric Encryption Requirement: Users encrypt data with a public key, and only the company can decrypt it using a private key.
Data Encryption at Rest and In Transit: The data must be encrypted during upload (in transit) and when stored in Amazon S3 (at rest).
Step 2: Solution Analysis
Option A: Server-side encryption with Amazon S3 managed keys (SSE-S3).
Amazon S3 manages the encryption and decryption keys.
This does not meet the requirement for asymmetric encryption, where the company uses a private key.
Not suitable.
Option B: Server-side encryption with customer-provided keys (SSE-C).
Requires the user to supply encryption keys during the upload process.
Does not align with the asymmetric encryption requirement.
Not suitable.
Option C: Client-side encryption with a data key.
Data key encryption is symmetric, not asymmetric.
Does not satisfy the requirement for a public-private key pair.
Not suitable.
Option D: Client-side encryption with a customer-managed encryption key.
Data is encrypted on the client side using the public key.
Only the company can decrypt the data using the corresponding private key.

Data remains encrypted during upload (in transit) and in S3 (at rest).

Correct option.

Step 3: Implementation Steps for Option D

Generate Key Pair:

The company generates an RSA key pair (public/private) for encryption and decryption.

Encrypt Data on Client Side:

Use the public key to encrypt the data before uploading to S3.

S3 Upload:

Upload the encrypted data to S3 over an HTTPS connection.

Decrypt Data on the Server:

Use the private key to decrypt data when needed.

AWS Developer Reference:

Amazon S3 Encryption Options

Asymmetric Key Cryptography in AWS

Question: 337

A company has a serverless application that uses an Amazon API Gateway API to invoke an AWS Lambda function. A developer creates a fix for a defect in the Lambda function code. The developer wants to deploy this fix to the production environment. To test the changes, the developer needs to send 10% of the live production traffic to the updated Lambda function version.

Options:

A. Publish a new version of the Lambda function that contains the updated code.

B. Set up a new stage in API Gateway with a new Lambda function version. Enable weighted routing in API Gateway stages.

C. Create an alias for the Lambda function. Configure weighted routing on the alias. Specify a 10% weight for the new Lambda function version.

D. Set up a routing policy on a Network Load Balancer. Configure 10% of the traffic to go to the new Lambda function version.

E. Set up a weighted routing policy by using Amazon Route 53. Configure 10% of the traffic to go to the new Lambda function version.

Answer: A, C

Explanation:

Step 1: Understanding the Requirements

Gradual Traffic Shift: Test the new version by routing only 10% of production traffic to it.

Lambda Deployment: Use versioning and aliases to manage Lambda function updates.

Step 2: Solution Analysis

Option A:

Publishing a new version creates an immutable version of the Lambda function with the updated code.

This is a prerequisite for deploying changes using weighted aliases.

Correct option.

Option B:

API Gateway stages are not used for weighted routing; they represent environments like "dev" or "prod."

Weighted routing is implemented using Lambda aliases, not API Gateway stages.

Not suitable.

Option C:

Lambda aliases allow traffic to be split between versions using weighted routing.

Assign a 90% weight to the old version and 10% to the new version to implement the gradual rollout.

Correct option.

Option D:

Network Load Balancers are not suitable for managing Lambda function traffic directly.

Not applicable.

Option E:

Route 53 routing policies apply at the DNS level and are not designed for Lambda version management.

Not suitable.

Step 3: Implementation Steps

Publish a New Version:

Publish the updated Lambda function code as a new version.

Create an Alias and Configure Weighted Routing:

Create an alias (e.g., prod) and associate it with both the old and new versions.

Set weights for traffic distribution:

```
aws lambda update-alias --function-name my-function \
--name prod --routing-config '{"AdditionalVersionWeights": {"2": 0.1}}'
```

AWS Developer Reference:

Lambda Function Aliases

Weighted Traffic Routine with Lambda

Question: 338

A company created an application to consume and process dat

a. The application uses Amazon SQS and AWS Lambda functions. The application is currently working as expected, but it occasionally receives several messages that it cannot process properly. The company needs to clear these messages to prevent the queue from becoming blocked. A developer must implement a solution that makes queue processing always operational. The solution must give the company the ability to defer the messages with errors and save these messages for further analysis. What is the MOST operationally efficient solution that meets these requirements?

A. Configure Amazon CloudWatch Logs to save the error messages to a separate log stream.

B. Create a new SQS queue. Set the new queue as a dead-letter queue for the application queue. Configure the Maximum Receives setting.

C. Change the SQS queue to a FIFO queue. Configure the message retention period to 0 seconds.

D. Configure an Amazon CloudWatch alarm for Lambda function errors. Publish messages to an Amazon SNS topic to notify administrator users.

Answer: B

Explanation:

Using a dead-letter queue (DLQ) with Amazon SQS is the most operationally efficient solution for handling unprocessable messages.

Amazon SQS Dead-Letter Queue:

A DLQ is used to capture messages that fail processing after a specified number of attempts. Allows the application to continue processing other messages without being blocked. Messages in the DLQ can be analyzed later for debugging and resolution.

Why DLQ is the Best Option:

Operational Efficiency: Automatically defers messages with errors, ensuring the queue is not blocked.

Analysis Ready: Messages in the DLQ can be inspected to identify recurring issues.

Scalable: Works seamlessly with Lambda and SQS at scale.

Why Not Other Options:

Option A: Logs the messages but does not resolve the queue blockage issue.

Option C: FIFO queues and 0-second retention do not provide error handling or analysis capabilities. Option D: Alerts administrators but does not handle or store the unprocessable messages.

Steps to Implement:

Create a new SQS queue to serve as the DLQ.

Attach the DLQ to the primary queue and configure the Maximum Receives setting.

Reference:

Using Amazon SQS Dead-Letter Queues

Best Practices for Using Amazon SQS with AWS Lambda

Question: 339

A developer created an AWS Lambda function to process data in an application. The function pulls large objects from an Amazon S3 bucket, processes the data, and loads the processed data into a second S3 bucket. Application users have reported slow response times. The developer checks the logs and finds that Lambda function invocations run much slower than expected. The function itself is simple and has a small deployment package. The function initializes quickly. The developer needs to improve the performance of the application. Which solution will meet this requirement with the LEAST operational overhead?

A. Store the data in an Amazon EFS file system. Mount the file system to a local directory in the function.

B. Create an Amazon EventBridge rule to schedule invocations of the function every minute.

C. Configure the function to use ephemeral storage. Upload the objects and process data in the /tmp directory.

D. Create a Lambda layer to package the function dependencies. Add the layer to the function.

Answer: C

Explanation:

Configuring the Lambda function to use ephemeral storage and processing data in the /tmp directory improves performance by leveraging local storage during execution.

Why Option C is Correct:

Ephemeral Storage: Lambda provides temporary storage (up to 10 GB) in the /tmp directory for each invocation, which is faster than pulling data directly from S3 multiple times.

Performance Boost: Data can be downloaded to /tmp, processed locally, and uploaded to the destination S3 bucket, minimizing S3 network calls.

Low Overhead: This approach requires only minimal changes to the function's configuration.

Why Not Other Options:

Option A: Using Amazon EFS adds complexity and is unnecessary for this use case.

Option B: Scheduling the function does not address the root cause of slow performance.

Option D: Lambda layers improve deployment efficiency, not runtime performance for this scenario.

Reference:

Using Ephemeral Storage in AWS Lambda

Best Practices for S3 and Lambda Performance

Question: 340

A company uses more than 100 AWS Lambda functions to handle application services. One Lambda function is critical and must always run successfully. The company notices that occasionally, the critical Lambda function does not initiate. The company investigates the issue and discovers instances of the Lambda TooManyRequestsException: Rate Exceeded error in Amazon CloudWatch logs. Upon further review of the logs, the company notices that some of the non-critical functions run properly while the critical function fails. A developer must resolve the errors and ensure that the critical Lambda function runs successfully. Which solution will meet these requirements with the LEAST operational overhead?

A. Configure reserved concurrency for the critical Lambda function. Set reserved concurrent executions to the appropriate level.

B. Configure provisioned concurrency for the critical Lambda function. Set provisioned concurrent executions to the appropriate level.

C. Configure CloudWatch alarms for TooManyRequestsException errors. Add the critical Lambda function as an alarm state change action to invoke the critical function again after a failure.

D. Configure CloudWatch alarms for TooManyRequestsException errors. Add Amazon EventBridge as an action for the alarm state change. Use EventBridge to invoke the critical function again after a failure.

Answer: A

Explanation:

Reserved concurrency guarantees a specific number of concurrent executions for a critical Lambda function. This ensures that the critical function always has sufficient resources to execute, even if other functions are consuming concurrency.

Why Option A:

Ensures Function Availability: Reserved concurrency isolates the critical Lambda function from other functions.

Low Overhead: Configuring reserved concurrency is straightforward and requires minimal setup. Why Not Other Options:

Option B: Provisioned concurrency is ideal for reducing cold starts, not for managing execution limits. Option C & D: Alarms and re-invocation mechanisms add complexity without resolving the root cause.

Reference:

Managing Concurrency for AWS Lambda

Question: 341

A developer is creating a microservices application that runs across multiple compute environments. The application must securely access secrets that are stored in AWS Secrets Manager with minimal network latency. The developer wants a solution that reduces the number of direct calls to Secrets Manager and simplifies secrets management across environments. Which solution will meet these requirements with the LEAST operational overhead?

A. Create a custom script that retrieves secrets directly from Secrets Manager and caches the secrets in a local database for each compute environment.

B. Install the Secrets Manager Agent in each compute environment. Configure the agent to cache secrets locally. Securely retrieve the secrets from Secrets Manager as needed.

C. Implement lazy loading logic in the application to fetch secrets directly from Secrets Manager and to cache the secrets in Redis.

D. Store the secrets in an Amazon S3 bucket. Retrieve and load the secrets as environment variables during application startup for each compute environment.

Answer: B

Explanation:

The Secrets Manager Agent provides an out-of-the-box solution for securely caching secrets locally, reducing latency and operational overhead.

Why Option B is Correct:

Caching: The agent securely caches secrets locally, minimizing Secrets Manager API calls.

Security: Secrets remain secure during retrieval and storage.

Low Operational Overhead: Managed solution eliminates the need for custom logic.

Why Not Other Options:

Option A: Custom scripts introduce complexity and require ongoing maintenance.

Option C: Using Redis requires managing an additional service, increasing overhead.

Option D: Storing secrets in S3 lacks the fine-grained security controls of Secrets Manager. Reference:

Caching Secrets in AWS Secrets Manager

Question: 342

A company has an application that is based on Amazon EC2. The company provides API access to the application through Amazon API Gateway and uses Amazon DynamoDB to store the application's dat a. A developer is investigating performance issues that are affecting the application. During peak usage, the application is overwhelmed by a large number of identical data read requests that come through APIs. What is the MOST operationally efficient way for the developer to improve the application's performance?

A. Use DynamoDB Accelerator (DAX) to cache database responses.
B. Configure Amazon EC2 Auto Scaling policies to meet fluctuating demand.
C. Enable API Gateway caching to cache API responses.
D. Use Amazon ElastiCache to cache application responses.

Answer: A

Explanation:

DynamoDB Accelerator (DAX) provides a managed caching layer specifically optimized for DynamoDB, reducing latency for repeated read requests.

Why Option A is Correct:

Purpose-Built: DAX is designed for DynamoDB, enabling sub-millisecond response times for frequently accessed items.

Operational Efficiency: No need for additional application-level caching logic.

Why Not Other Options:

Option B: Auto Scaling increases capacity but does not address repetitive reads.

Option C: API Gateway caching helps reduce request processing time but does not optimize DynamoDB reads.

Option D: ElastiCache is a general-purpose cache, adding unnecessary complexity for DynamoDB use cases.

Reference:

DynamoDB Accelerator (DAX)

Question: 343

A banking company is building an application for users to create accounts, view balances, and review recent transactions. The company integrated an Amazon API Gateway REST API with AWS Lambda functions. The company wants to deploy a new version of a Lambda function that gives customers the ability to view their balances. The new version of the function displays customer transaction insights. The company wants to test the new version with a small group of users before deciding whether to make the feature available for all users. Which solution will meet these requirements with the LEAST disruption to users?

A. Create a canary deployment for the REST API. Gradually increase traffic to the new version of the function. Revert traffic to the old version if issues are detected.

B. Redeploy the REST API stage to use the new version of the function. If issues are detected, update the REST API to point to the previous version of the function.

C. Deploy the new version of the function to a new stage in the REST API. Route traffic to the new stage. If the new version fails, route traffic to the original stage.

D. Create a new REST API stage for the new version of the function. Create a weighted alias record set in Amazon Route 53 to distribute traffic between the original stage and the new stage.

Answer: A

Explanation:

API Gateway's canary deployments allow gradual traffic shifting to a new version of a function, minimizing disruption while testing.

Why Option A is Correct:

Gradual Rollout: Reduces risk by incrementally increasing traffic.

Rollback Support: Canary deployments make it easy to revert to the previous version.

Why Not Other Options:

Option B: Redeploying the stage disrupts all users.

Option C & D: Managing new stages and weighted routing introduces unnecessary complexity. Reference:

Canary Deployments in API Gateway

Question: 344

A developer is receiving an intermittent ProvisionedThroughputExceededException error from an application that is based on Amazon DynamoDB. According to the Amazon CloudWatch metrics for the table, the application is not exceeding the provisioned throughput. What could be the cause of the issue?

A. The DynamoDB table storage size is larger than the provisioned size.

B. The application is exceeding capacity on a particular hash key.

C. The DynamoDB table is exceeding the provisioned scaling operations.

D. The application is exceeding capacity on a particular sort key.

Answer: B

Explanation:

DynamoDB distributes throughput across partitions based on the hash key. A hot partition (caused by high usage of a specific hash key) can result in a ProvisionedThroughputExceededException, even if overall usage is below the provisioned capacity.

Why Option B is Correct:

Partition-Level Limits: Each partition has a limit of 3,000 read capacity units or 1,000 write capacity units per second.

Hot Partition: Excessive use of a single hash key can overwhelm its partition.

Why Not Other Options:

Option A: DynamoDB storage size does not affect throughput.

Option C: Provisioned scaling operations are unrelated to throughput errors.

Option D: Sort keys do not impact partition-level throughput.

Reference:

DynamoDB Partition Key Design Best Practices