

DVA-C02 Dumps

DVA-C02

<https://www.certleader.com/DVA-C02-dumps.html>



NEW QUESTION 1

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. A version 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. Store the environment-specific credentials in the secret.
- D. Configure AWS Secrets Manager to create a new secret for each environment type.

Answer: D

Explanation:

AWS Secrets Manager is the best option for managing sensitive credentials across multiple environments, as it provides automatic secret rotation, auditing, and monitoring features. It also allows storing environment-specific credentials in separate secrets, which can be accessed by the applications using the SDK or CLI. AWS Systems Manager Parameter Store does not have built-in secret rotation capability, and it requires creating individual parameters or storing the entire credential set as a JSON object. Configuring the environment variables in the application code is not a secure or scalable solution, as it exposes the credentials to anyone who can access the code. References

? AWS Secrets Manager vs. Systems Manager Parameter Store

? AWS Systems Manager Parameter Store vs. Secrets Manager vs. Environment Variable in Lambda, when to use which

? AWS Secrets Manager vs. Parameter Store: Features, Cost & More

NEW QUESTION 2

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.
- B. Add an Amazon API Gateway API to invoke the function.
- C. Call the API from the client side when login confirmation is received.
- D. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification.
- E. Add an Amazon Cognito post authentication Lambda trigger for the function.
- F. Create an AWS Lambda function that uses Amazon Simple Email Service (Amazon SES) to send the email notification.
- G. Create an Amazon CloudWatch Logs log subscription filter to invoke the function based on the login status.
- H. Configure Amazon Cognito to stream all logs to Amazon Kinesis Data Firehose.
- I. 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

NEW QUESTION 3

A developer is building an application that uses AWS API Gateway APIs, AWS Lambda functions, and AWS DynamoDB tables. The developer uses the AWS Serverless Application Model (AWS SAM) to build and run serverless applications on AWS. Each time the developer pushes changes for only 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. `aws sam deploy --force-upload`
- B. `aws sam deploy --no-execute-changeset`
- C. `aws sam package`
- D. `aws sam sync --watch`

Answer: D

Explanation:

The command that will meet the requirements is `aws 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

NEW QUESTION 4

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

Amazon CodeGuru

D.

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.

References:

? [What Is AWS CodeCommit? - AWS CodeCommit]

? [AWS CodePipeline - AWS CodeCommit]

NEW QUESTION 5

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 Lambda 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
- B. Set the No Echo attribute to true.
- C. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and to create a parameter
- D. In AWS Systems Manager Parameter Store
- E. Set the No Echo attribute to true
- F. Create an IAM role that has the ssm:GetParameter permission
- G. Assign the role to the Lambda function
- H. Store the parameter name as the Lambda function's environment variable
- I. Resolve the parameter's value at runtime.
- J. 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. We encrypt the parameter's value by using the AWS Key Management Service (AWS KMS) Encrypt command.
- K. Use CloudFormation to create an AWS Secrets Manager secret
- L. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions
- M. Create an IAM role that has the secrets:manage
- N. GetSecretValue permission
- O. Assign the role to the Lambda function. Store the secret's name as the Lambda function's environment variable
- P. Resolve 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

NEW QUESTION 6

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
- B. Use unique paths in Parameter Store for each variable in each environment
- C. Store the credentials in AWS Secrets Manager in each environment.
- D. 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.
- E. Update the application to retrieve the variables from an encrypted file that is stored with the application
- F. Store the API URL and credentials in unique files for each environment.
- G. Update the application to retrieve the variables from each of the deployed environments
- H. 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.

References:

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

? [Parameter Store - AWS Systems Manager]

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

NEW QUESTION 7

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 AP
- B. Instruct the other developers to point the endpoints to the development stage.
- C. Define a new API Gateway API that points to the new API application code
- D. Instruct the other developers to point the endpoints to the new API.
- E. Implement a query parameter in the API application code that determines which code version to call.
- F. 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.

References:

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

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

NEW QUESTION 8

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:

Setting the event source mapping maximum concurrency is the best way to control how many messages from each queue are processed by the Lambda function at a time. The maximum concurrency setting limits the number of batches that can be processed concurrently from the same event source. By setting it to 10 for the high priority queue and to 90 for the low priority queue, the developer can ensure that the Lambda function always reads up to 10 simultaneous messages from the high priority queue before processing messages from the low priority queue, and that the total number of concurrent invocations does not exceed 100. The other solutions are either not effective or not relevant. The batch size setting controls how many messages are sent to the Lambda function in a single invocation, not how many invocations are allowed at a time. The delivery delay setting controls how long a message is invisible in the queue after it is sent, not how often it is processed by the Lambda function. The batch window setting controls how long the event source mapping can buffer messages before sending a batch, not how many batches are processed concurrently. References

? Using AWS Lambda with Amazon SQS

? AWS Lambda Event Source Mapping - Examples and best practices | Shisho Dojo

? Lambda event source mappings - AWS Lambda

? aws_lambda_event_source_mapping - Terraform Registry

NEW QUESTION 9

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.
- D. Create an EventBridge rule to invoke an AWS Lambda function to process the documents.
- E. Update the application to synchronously process the documents directly after the DynamoDB write.

Answer: B

Explanation:

<https://aws.amazon.com/blogs/database/dynamodb-streams-use-cases-and-design-patterns/>

NEW QUESTION 10

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:

The HTTP 401 error indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. This is the appropriate error code to return when the user login fails due to invalid credentials. The HTTP 405 error means that the method specified in the request is not allowed for the resource identified by the request URI, which is not the case here. The other error codes are not relevant to the authentication failure scenario.

References

? HTTP Status Codes

? AWS Lambda Function Errors in API Gateway

NEW QUESTION 10

A mobile app stores blog posts in an Amazon DynamoDB 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 than 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
- B. Create a script to find old posts with a table scan and remove posts that are older than 48 hours by using the Batch Write Item API operation
- C. Schedule a cron job on an Amazon EC2 instance once an hour to start the script.
- D. For each item add a new attribute of type String that has a timestamp that is set to the blog post creation time
- E. Create a script to find old posts with a table scan and remove posts that are older than 48 hours by using the Batch Write item API operation
- F. Place the script in a container image
- G. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Fargate that invokes the container every 5 minutes.
- H. 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
- I. Create a global secondary index (GSI) that uses the new attribute as a sort key
- J. Create an AWS Lambda function that references the GSI and removes expired items by using the Batch Write item API operation. Schedule the function with an Amazon CloudWatch event every minute.
- K. For each item add a new attribute of type Number that has a timestamp that is set to 48 hours after the blog post creation time
- L. 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.

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

NEW QUESTION 14

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
- B. Update the API Gateway endpoint to reference the new Lambda function version
- C. Upload and publish the optimized Lambda function code
- D. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release
- E. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function
- F. Publish the API to the canary stage.
- G. Define a function version for the currently deployed production Lambda function
- H. Update the API Gateway endpoint to reference the new Lambda function version
- I. Upload and publish the optimized Lambda function code
- J. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function
- K. Deploy a new API Gateway stage.
- L. Define an alias on the \$LATEST version of the Lambda function
- M. Update the API Gateway endpoint to reference the new Lambda function alias
- N. Upload and publish the optimized Lambda function code
- O. On the production API Gateway stage, define a canary release and set the percentage of traffic to direct to the canary release
- P. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function
- Q. Publish to the canary stage.
- R. Define a function version for the currently deployed production Lambda function
- S. Update the API Gateway endpoint to reference the new Lambda function version
- T. Upload and publish the optimized Lambda function code
- U. Update the API Gateway endpoint to use the \$LATEST version of the Lambda function
- V. 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 alias. A Lambda alias allows you to invoke different versions of a function using the same name. You can also split traffic between two aliases by assigning weights to them.

? 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:

? 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 needed.

NEW QUESTION 19

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 IAM roles for the identity pools.
- B. Program the sign-in page to create users' IAM groups with the IAM 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:

<https://docs.aws.amazon.com/cognito/latest/developerguide/signing-up-users-in-your-app.html>

NEW QUESTION 23

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::StackId 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:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/mappings-section-structure.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo-parameter-reference.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/pseudo-parameter-reference.html>

NEW QUESTION 24

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 exceeds 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.
- B. Use Amazon CloudWatch Logs Insights to query the CloudWatch log.
- C. Schedule the Lambda function to check the CloudWatch logs and notify the existing SNS topic.
- D. Publish custom metrics to CloudWatch that record the failures of the external payment processing API call.
- E. Configure a CloudWatch alarm to notify the existing SNS topic when error rate exceeds the specified rate.
- F. Publish the results of the external payment processing API calls to a new Amazon SNS topic.
- G. Subscribe the support team members to the new SNS topic.
- H. Write the results of the external payment processing API calls to Amazon S3. Schedule an Amazon Athena query to run at regular intervals.
- I. 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.

References:

? [What Is Amazon CloudWatch? - Amazon CloudWatch]

? [Publishing Custom Metrics - Amazon CloudWatch]

? [Creating Amazon CloudWatch Alarms - Amazon CloudWatch]

NEW QUESTION 28

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 in 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.

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

NEW QUESTION 32

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 databases
- B. Store the session data and the application data in the MySQL database.
- C. Use Amazon ElastiCache for Memcached to store and manage the session data
- D. Use an Amazon RDS for MySQL DB instance to store the application data.
- E. Use Amazon ElastiCache for Memcached to store and manage the session data and the application data.
- F. Use the EC2 instance store to manage the session data
- G. 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.

References: Amazon ElastiCache, Amazon RDS

NEW QUESTION 35

An online food company provides an Amazon API Gateway HTTP API to 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 partners require additional Lambda function to receive orders. The company has created an Amazon S3 bucket. The company needs to store all orders and updates in the S3 bucket for future analysis.

How can the developer ensure that all 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
- B. Configure the new Lambda function to write to the S3 bucket
- C. Modify the original Lambda function to post updates to the new API endpoint.
- D. Use Amazon Kinesis Data Streams to create a new data stream
- E. Modify the Lambda function to publish orders to the data stream. Configure the data stream to write to the S3 bucket.
- F. Enable DynamoDB Streams on the DynamoDB table
- G. Create a new Lambda function

H. Associate the stream's Amazon Resource Name (ARN) with the Lambda Function

Configure the Lambda function to write to the S3

I. Modify the Lambda function to publish to a new Amazon

J. Simple Lambda function receives order

K. Subscribe a new Lambda function to the topic

L. 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. References: Using DynamoDB Streams, Using AWS Lambda with Amazon S3

NEW QUESTION 39

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 instance
- B. Deploy a file system on the EBS volum
- C. Use the host operating system to share a folde
- D. Update the application code to read and write configuration files from the shared folder.
- E. Deploy a micro EC2 instance with an instance store volum
- F. Use the host operating system to share a folde
- G. Update the application code to read and write configuration files from the shared folder.
- H. Create an Amazon S3 bucket to host the repositor
- I. Migrate the existing .xml files to the S3 bucke
- J. Update the application code to use the AWS SDK to read and write configuration files from Amazon S3.
- K. Create an Amazon S3 bucket to host the repositor
- L. Migrate the existing .xml files to the S3 bucke
- M. Mount the S3 bucket to the EC2 instances as a local volum
- N. 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.

References:

- ? [Amazon Simple Storage Service (S3)]
- ? [Using AWS SDKs with Amazon S3]

NEW QUESTION 43

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 IAM 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 topi
- B. Create an Amazon EventBridge schedule to invoke the Lambda function every 15 minutes
- C. 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 15 minutes
- D. Launch an Amazon EC2 instance that includes a script to filter events from CloudTrail if a role was created without CloudFormatio
- E. Configure the script to publish to the SNS topi
- F. Create a cron job to run the script on the EC2 instance every 15 minutes.
- G. 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:

Creating an Amazon EventBridge rule is the most efficient and scalable way to monitor and react to events from CloudTrail, such as the creation of an IAM role without CloudFormation. EventBridge allows you to specify a filter pattern to match the events you are interested in, and then specify an SNS topic as the target to send notifications. This solution does not require any additional resources or code, and it can trigger notifications in near real-time. The other solutions involve creating and managing additional resources, such as Lambda functions, Fargate tasks, or EC2 instances, and they rely on polling CloudTrail events every 15 minutes, which can introduce delays and increase costs. References

- ? Using Amazon EventBridge rules to process AWS CloudTrail events
- ? Using AWS CloudFormation to create and manage AWS Batch resources
- ? How to use AWS CloudFormation to configure auto scaling for Amazon Cognito and AWS AppSync
- ? Using AWS CloudFormation to automate the creation of AWS WAF web ACLs, rules, and conditions

NEW QUESTION 48

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 is 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_OLD_IMAGES 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]

NEW QUESTION 50

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
- B. Select the database that the parameter will access
- C. Use the default AWS Key Management Service (AWS KMS) key to encrypt the parameter
- D. Enable automatic rotation for the parameter
- E. Use the parameter from Parameter Store on the Lambda function to connect to the database.
- F. Encrypt the credentials with the default AWS Key Management Service (AWS KMS) key
- G. Store the credentials as environment variables for the Lambda function
- H. Create a second Lambda function to generate new credentials and to rotate the credentials by updating the environment variables of the first Lambda function
- I. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule
- J. Update the database to use the new credential
- K. On the first Lambda function, retrieve the credentials from the environment variable
- L. Decrypt the credentials by using AWS KMS, connect to the database.
- M. Store the credentials in AWS Secrets Manager
- N. Set the secret type to Credentials for Amazon RDS databases
- O. Select the database that the secret will access
- P. Use the default AWS Key Management Service (AWS KMS) key to encrypt the secret
- Q. Enable automatic rotation for the secret
- R. Use the secret from Secrets Manager on the Lambda function to connect to the database.
- S. Encrypt the credentials by using AWS Key Management Service (AWS KMS). Store the credentials in an Amazon DynamoDB table
- T. Create a second Lambda function to rotate the credential
- U. Invoke the second Lambda function by using an Amazon EventBridge rule that runs on a schedule
- V. Update the DynamoDB table
- W. Update the database to use the generated credential
- X. Retrieve the credentials from DynamoDB with the first Lambda function
- Y. 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

NEW QUESTION 52

A developer is configuring an applications deployment environment in AWS CodePipeline. 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 CodeCommit project
- B. Add the repository package's build and test commands to the project's buildspec
- C. Create an AWS CodeBuild project
- D. Add the repository package's build and test commands to the project's buildspec
- E. Create an AWS CodeDeploy project
- F. Add the repository package's build and test commands to the project's buildspec
- G. Add an action to the source stage
- H. Specify the newly created project as the action provider
- I. Specify the build artifact as the action's input artifact.
- J. Add a new stage to the pipeline after the source stage
- K. Add an action to the new stage
- L. Specify the newly created project as the action provider
- M. Specify the source artifact as the action's input artifact.

Answer: BE

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. References: AWS CodeBuild, AWS CodePipeline

NEW QUESTION 57

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 IAM user with an appropriate policy
- C. Store the access key ID and secret access key on the EC2 instances
- D. Modify the application to use the S3 GeneratePresignedUrl API call
- E. Modify the application to use the S3 GetObject API call and to return the object handle to the user
- F. Modify the application to delegate requests to the S3 bucket.

Answer: AC

Explanation:

The most secure way to allow the EC2 instances to access the S3 bucket is to use an EC2 instance profile and role with an appropriate policy that grants the necessary permissions. This way, the EC2 instances can use temporary security credentials that are automatically rotated and do not need to store any access keys on the instances. To allow the users who are signed in to the application to download objects from the S3 bucket, the application can use the S3 GeneratePresignedUrl API call to create a pre-signed URL that grants temporary access to a specific object. The pre-signed URL can be returned to the user, who can then use it to download the object within a specified time period. References

- ? Use Amazon S3 with Amazon EC2
- ? How to Access AWS S3 Bucket from EC2 Instance In a Secured Way
- ? Sharing an Object with Others

NEW QUESTION 60

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
- C. Ask the customers to send a request that contains the HTTP header when they make an API call.
- D. Ask the customers to use the AWS SDK API Gateway class to invoke the InvalidateCache API operation.
- E. Attach an InvalidateCache policy to the IAM execution role that the customers use to invoke the API
- F. Ask the customers to add the INVALIDATE_CACHE query string parameter when they make an API call.

Answer: D

NEW QUESTION 61

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 an AWS 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:

The easiest way to add the S3 bucket to the CloudFormation template is to use the S3Bucket and S3Key properties of the AWS::Lambda::Function resource. These properties specify the name of the S3 bucket and the location of the .zip file that contains the function code and dependencies. This way, the developer does not need to modify the function code or upload it to a different location. The other options are either not feasible or not efficient.

The code property can only be used for inline code, not for code stored in S3. The ZipFile property can only be used for code that is less than 4096 bytes, not for code that has dependencies. Finding the S3 key for the Lambda function and adding it as the ZipFile property would not work, as the ZipFile property expects a base64-encoded .zip file, not an S3 location. References

- ? AWS::Lambda::Function - AWS CloudFormation
- ? Deploying Lambda functions as .zip file archives
- ? AWS Lambda Function Code - AWS CloudFormation

NEW QUESTION 64

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.
- C. 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.

E. 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.

- F. 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:

The best solution is to create a CloudWatch metric filter to match the application error pattern in the log data. This will allow you to create a custom metric that tracks the number of errors in your application. You can then set up a CloudWatch alarm based on this metric and configure it to send an SNS notification when

the number of errors exceeds a defined threshold within a 5-minute period. This solution does not require any changes to your application code or installing any additional agents on your EC2 instances. It also leverages the existing integration between CloudWatch and SNS for sending notifications. References

- ? Create Metric Filters - Amazon CloudWatch Logs
- ? Creating Amazon CloudWatch Alarms - Amazon CloudWatch
- ? How to send alert based on log message on CloudWatch - Stack Overflow

NEW QUESTION 68

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 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 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 input, as the partition key and email_address as the sort key
- B. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property.
Add a global secondary index (GSI) to the DynamoDB table with email_address as the partition key and customer_type as the sort key
- C. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property.
- D. Perform a query operation on the GSI by using the begins_with key condition expression with the email_address property.
- E. Address property.
- F. Add a local secondary index (LSI) to the DynamoDB table with customer_type as the partition Key and email_address as the sort Key
- G. Perform a query operation on the LSI by using the begins_with Key condition expression with the email-address property.
- H. Add a local secondary index (LSI) to the DynamoDB table with job-title as the partition key and email_address as the sort key
- I. 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

NEW QUESTION 73

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
- B. Enable encryption of data in transit and at rest
- C. Store frequently accessed data in the cache.
- D. Create an Amazon ElastiCache for Memcached instance
- E. Enable encryption of data in transit and at rest
- F. Store frequently accessed data in the cache.
- G. Create an Amazon RDS for MySQL read replica
- H. Connect to the read replica by using SSL
- I. Configure the read replica to store frequently accessed data.
- J. Create an Amazon DynamoDB table and a DynamoDB Accelerator (DAX) cluster for the table
- K. 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.

References:

- ? [What Is Amazon ElastiCache? - Amazon ElastiCache]
- ? [Encryption in Transit - Amazon ElastiCache for Redis]
- ? [Encryption at Rest - Amazon ElastiCache for Redis]

NEW QUESTION 78

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 IAM 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)

NEW QUESTION 79

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 de-couple and scale for applications¹. Amazon SQS offers two types of queues: Standard and FIFO (First In First Out) queues¹. 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.

NEW QUESTION 84

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/*"
    },
    {
      "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

NEW QUESTION 88

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 partne

- B. Configure the Lambda function to publish messages for each partner to the partner's SNS topic.
- C. Create a different Lambda function for each partner
- D. Configure the Lambda function to notify each partner's service endpoint directly.
- E. Create an Amazon Simple Notification Service (Amazon SNS) topic
- F. Configure the Lambda function to publish messages with specific attributes to the SNS topic
- G. Subscribe each partner to the SNS topic
- H. Apply the appropriate filter policy to the topic subscriptions.
Create one Amazon Simple Notification Service (Amazon SNS) topic
- I. 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.

References:

- ? [Amazon Simple Notification Service (SNS)]
- ? [Filtering Messages with Attributes - Amazon Simple Notification Service]

NEW QUESTION 91

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 account
- B. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API
- C. Use the Lambda function to store the photos and details in the DynamoDB table
- D. Retrieve previously uploaded photos directly from the DynamoDB table.
- E. Use Amazon Cognito user pools to manage user account
- F. Create an Amazon Cognito user pool authorizer in API Gateway to control access to the API
- G. 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
- H. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.
- I. Create an IAM user for each user of the application during the sign-up process
- J. Use IAM authentication to access the API Gateway API

DynamoDB

- K. 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
- L. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.
- M. Create a users table in DynamoDB
- N. Use the table to manage user account
- O. Create a Lambda authorizer that validates user credentials against the users table
- P. Integrate the Lambda authorizer with API Gateway to control access to the API
- Q. 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
- R. 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.

References:

- ? [Amazon Cognito User Pools]
- ? [Use Amazon Cognito User Pools - Amazon API Gateway]
- ? [Amazon Simple Storage Service (S3)]
- ? [Amazon DynamoDB]

NEW QUESTION 94

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
- B. Store the unique identifier for each request in a database table
- C. Modify the Lambda function to check the table for the identifier before processing the request.
- D. Create an Amazon DynamoDB table
- E. Store the unique identifier for each request in the table
- F. Modify the Lambda function to check the table for the identifier before processing the request.
- G. Create an Amazon DynamoDB table
- H. Store the unique identifier for each request in the table
- I. Modify the Lambda function to return a client error response when the function receives a duplicate request.

- J. Create an Amazon ElastiCache for Memcached instance
- K. Store the unique identifier for each request in the cache
- L. 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

NEW QUESTION 96

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

NEW QUESTION 100

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 active-passive 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.
- B. Enable secret replication to the secondary Region. Update the application to use the Amazon Resource Name (ARN) based on the Region.
- C. Store credentials in AWS Systems Manager Parameter Store in the primary Region.
- D. Enable parameter replication to the secondary Region.
- E. Update the application to use the Amazon Resource Name (ARN) based on the Region.
- F. Store credentials in a config file.
- G. Upload the config file to an S3 bucket in the primary Region.
- H. Enable Cross-Region Replication (CRR) to an S3 bucket in the secondary region.
- I. Update the application to access the config file from the S3 bucket based on the Region.
- J. Store credentials in a config file.
- K. Upload the config file to an Amazon Elastic File System (Amazon EFS) file system.
- L. 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 monitoring¹. 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.

References:

- ? AWS Secrets Manager
- ? Replicating and sharing secrets
- ? Using your own encryption keys

NEW QUESTION 101

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.
- B. Generate a URL that retrieves the reports from DynamoDB.

- C. Provide the URL to customers through the web application.
- D. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption
- E. Attach the reports to an Amazon Simple Notification Service (Amazon SNS) message
- F. Subscribe the customer to email notifications from Amazon SNS.
- G. Generate the reports and then store the reports in an Amazon S3 bucket that uses server-side encryption
- H. Generate a presigned URL that contains an expiration date Provide the URL to customers through the web application
- I. Add S3 Lifecycle configuration rules to the S3 bucket to delete old reports.
- J. Generate the reports and then store the reports in an Amazon RDS database with a date stamp
- K. Generate an URL that retrieves the reports from the RDS database
- L. Provide the URL to customers through the web application
- M. 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.

References: Amazon S3 Presigned URLs, Amazon S3 Lifecycle

NEW QUESTION 105

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]

NEW QUESTION 110

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
- B. Create automated test scripts that use the `cdk local invoke` command to invoke the Lambda function
- C. Check the responses
- D. Document the test scripts for the other developers on the team
- E. Update the CI/CD pipeline to run the test scripts.

Create sample events based on the Lambda

F. Install a unit testing framework that reproduces the Lambda execution environment.

G. Invoke the handler function by using a unit testing framework

H. Check the responses

I. Document how to run the unit testing framework for the other developers on the team

J. Update the CI/CD pipeline to run the unit testing framework.

K. Install the AWS Serverless Application Model (AWS SAM) CLI tool

L. Use the `sam local generate-event` command to generate sample events for the automated test

M. Create automated test scripts that use the `sam local invoke` command to invoke the Lambda function

N. Check the responses

O. Document the test scripts for the other developers on the team

P. Update the CI/CD pipeline to run the test scripts.

Q. Create sample events based on the Lambda documentation

R. Create a Docker container from the Node.js base image to invoke the Lambda function

S. Check the responses

T. Document how to run the Docker container for the other developers on the team

U. Update the CI/CD 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 generate-event` 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.

NEW QUESTION 111

A company is migrating an on-premises database to Amazon RDS for MySQL. The company has read-heavy 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?

Use a multi-AZ Amazon RDS deployment

A: 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.

C. Use a multi-AZ Amazon RDS deployment

D. Modify the code so that queries access the secondary RDS instance.

E. Deploy Amazon RDS with one or more read replicas

F. Modify the application code so that queries use the URL for the read replicas.

G. Use open source replication software to create a copy of the MySQL database on an Amazon EC2 instance

H. 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 offloading 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

NEW QUESTION 113

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.

NEW QUESTION 116

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 `DeploymentPreferenceType` property to `Canary` and `Percent10Minute`

B. Set the `AutoPublishAlias` property to the Lambda alias.

C. Set the `DeploymentPreferenceType` property to `Linear` and `PercentEvery10Minute`

D. Set `AutoPublishAlias` property to the Lambda alias.

E. Set the `DeploymentPreferenceType` property to `Canary` and `Percent10Minute`

F. Set the `PreTraffic` and `PostTraffic` properties to the Lambda alias.

G. Set the `DeploymentPreferenceType` property to `Linear` and `PercentEvery10Minute`

H. Set `PreTraffic` and `PostTraffic` properties to the Lambda alias.

Answer: A

Explanation:

? The `DeploymentPreferenceType` property specifies how traffic should be shifted between versions of a Lambda function¹. 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 shifted¹. 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 function¹. This is required to use the `DeploymentPreferenceType` property¹. The alias name can be specified by the developer, and it can be used to invoke the function with the latest code.

NEW QUESTION 121

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?

Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS).

A: 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 or sequentially on compute environments⁸. 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.

References:

- ? 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

NEW QUESTION 126

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 tabl
- B. Create a trigger to connect the data stream to the Lambda function.
- C. Create an Amazon EventBridge rule to invoke the Lambda function on a regular schedule
- D. Connect to the DynamoDB table from the Lambda function to detect changes.
- E. Enable DynamoDB Streams on the tabl
- F. Create a trigger to connect the DynamoDB stream to the Lambda function.
- G. Create an Amazon Kinesis Data Firehose delivery stream, and attach it to the DynamoDB tabl
- H. 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.

References:

- ? [Amazon DynamoDB]
- ? [DynamoDB Streams - Amazon DynamoDB]
- ? [Using AWS Lambda with Amazon DynamoDB - AWS Lambda]

NEW QUESTION 127

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.

```
aws dynamodb get-item --table-name demoman-table --key '{"id": {"N": "1993"}}'
```

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 DynamoDB cannot be accessed from the AWS CLI and needs to be 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.

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

NEW QUESTION 128

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:

"Deletes the item from your table without consuming any write throughput" <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html>

NEW QUESTION 131

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 1 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. Use the plaintext key from the GenerateDataKey API to encrypt the file.
- C. Write the plain text key from the GenerateDataKey API to disk for later use.
- D. Use the encrypted key from the GenerateDataKey API and a symmetric encryption algorithm to encrypt the file.
- E. Write the encrypted key from the GenerateDataKey API to disk for later use.
- F. Use the plaintext key from the GenerateDataKey API to encrypt the file by using the KMS Encrypt API.
- G. Write the plain text key from the GenerateDataKey API to disk for later use.
- H. 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 key. The data key is a random byte string that can be used with any standard encryption algorithm, such as AES or SM4. 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 KMS.

? 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:

NEW QUESTION 132

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 to 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: BD

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.

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

NEW QUESTION 136

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

NEW QUESTION 137

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 to an Amazon DynamoDB database.
- C. Configure the Amazon Aurora MySQL DB instance for 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.

References: [Amazon RDS Proxy], [Working with Amazon RDS Proxy]

NEW QUESTION 138

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
- D. Create a table in the database for parameter configuration
- E. Migrate the parameters that the application is modifying from Parameter Store to the configuration table
- F. Modify the CloudFormation stack policy to deny updates on Parameter Store parameters

Answer: D

Explanation:

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/protect-stack-resources.html#stack-policy-samples>

NEW QUESTION 142

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-digit 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]

NEW QUESTION 146

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?

Query the instance metadata from `http://169.254.169.254/latest/meta-data/`.

A. 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:

The instance metadata service provides information about the EC2 instance, including the public IPv4 address, which can be obtained by querying the endpoint `http://169.254.169.254/latest/meta-data/public-ipv4`. References

? Instance metadata and user data

? Get Public IP Address on current EC2 Instance

? Get the public ip address of your EC2 instance quickly

NEW QUESTION 148

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 are experiencing high latency due to static content on the EC2 instance, even during non-peak hours.

When combination of steps will resolve 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 static 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

NEW QUESTION 151

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 are name-value pairs that can be defined as configuration attributes associated with a deployment stage of a REST API. They act like environment variables and can be used in the API setup and mapping templates. For example, the development team can define a stage variable named `endpoint` and assign it different values for each stage, such as `dev.example.com` for development, `uat.example.com` for user acceptance testing, and `prod.example.com` for production. Then, the team can use the stage variable value in the integration request URL, such as `http://$ { stageVariables.endpoint}/api`. This way, the team can use the same API setup with different endpoints at each stage by resetting the stage variable value. The other solutions are either not feasible or not cost-effective. Lambda layers are used to package and load dependencies for Lambda functions, not for storing endpoints. Encoding the endpoints as query parameters would expose them to the public and make the request URL unnecessarily long. Storing the endpoints in AWS AppConfig would incur additional costs and complexity, and would require additional logic to retrieve the values from the configuration store. References

? Using Amazon API Gateway stage variables

? Setting up stage variables for a REST API deployment

? Setting stage variables using the Amazon API Gateway console

NEW QUESTION 156

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 least 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
- B. Create a new API Import the OpenAPI file Modify the new API to add request validation
- C. Perform the tests Modify the existing API to add request validation
- D. Deploy the existing API to production.
- E. Modify the existing API to add request validation
- F. Deploy the updated API to a new API Gateway stage Perform the tests Deploy the updated API to the API Gateway production stage.
- G. Create a new API Add the necessary resources and methods including new request validation
- H. Perform the tests Modify the existing API to add request validation
- I. Deploy the existing API to production.
- J. Clone the existing API Modify the new API to add request validation
Modify the existing API to add request validation Deploy the existing API to production.
- K. Perform the tests

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]

NEW QUESTION 157

A company is expanding the compatibility of its photo-sharing 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-effectively?

- A. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolution
- B. Create a dynamic CloudFront origin that automatically maps the request of each device to the corresponding photo variant.
- C. Use S3 Batch Operations to invoke an AWS Lambda function to create new variants of the photos with the required dimensions and resolution
- D. Create a Lambda@Edge function to route requests to the corresponding photo variant by using request headers.
- E. Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response
- F. Change the CloudFront TTL cache policy to the maximum value possible.

Create a Lambda@Edge function that optimizes the photos upon request and returns the photos as a response

G: 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]

NEW QUESTION 158

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:

The unified Amazon CloudWatch agent can collect both system metrics and log files from Amazon EC2 instances and on-premises servers. By installing and configuring the agent on the EC2 instances, the developer can easily access and analyze the application logs in CloudWatch without logging in to each server individually. This option requires minimum changes to the existing application and does not affect its availability or scalability. References

? Using the CloudWatch Agent

? Collecting Metrics and Logs from Amazon EC2 Instances and On-Premises Servers with the CloudWatch Agent

NEW QUESTION 159

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: BD

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.

References:

- ? [What Is Amazon API Gateway? - Amazon API Gateway]
- ? [Amazon API Gateway Metrics and Dimensions - Amazon CloudWatch]
- ? [Troubleshooting API Errors - Amazon API Gateway]

NEW QUESTION 160

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
- B. Deploy each Lambda function with its own copy of the library.
- C. Create a Lambda layer with the required Python library
- D. Use the Lambda layer in both Lambda functions.
- E. Combine the two Lambda functions into one Lambda function
- F. Deploy the Lambda function as a single .zip file archive.
- G. Download the Python library to an S3 bucket
- H. 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.

References:

- ? [What Is AWS Lambda? - AWS Lambda]
- ? [AWS Lambda Layers - AWS Lambda]

NEW QUESTION 164

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.

NEW QUESTION 165

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

All our products come with a 90-day Money Back Guarantee.

* One year free update

You can enjoy free update one year. 24x7 online support.

* Trusted by Millions

We currently serve more than 30,000,000 customers.

* Shop Securely

All transactions are protected by VeriSign!

100% Pass Your DVA-C02 Exam with Our Prep Materials Via below:

<https://www.certleader.com/DVA-C02-dumps.html>