

Exam Questions DVA-C02

DVA-C02

<https://www.2passeasy.com/dumps/DVA-C02/>



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 needs 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 Variables in Lambda, when to use which

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

NEW QUESTION 2

A developer is storing sensitive data generated by an application in Amazon S3. The developer wants to encrypt the data at rest. A company policy requires an audit trail of when the AWS Key Management Service (AWS KMS) key was used and by whom. Which encryption option will meet these requirements?

- A. Server-side encryption with Amazon S3 managed keys (SSE-S3)
Server-side encryption with AWS KMS managed keys (SSE-KMS)
- B. Server-side encryption with customer-provided keys (SSE-C)
- D. Server-side encryption with self-managed keys

Answer: B

Explanation:

This solution meets the requirements because it encrypts data at rest using AWS KMS keys and provides an audit trail of when and by whom they were used. Server-side encryption with AWS KMS managed keys (SSE-KMS) is a feature of Amazon S3 that encrypts data using keys that are managed by AWS KMS. When SSE-KMS is enabled for an S3 bucket or object, S3 requests AWS KMS to generate data keys and encrypts data using these keys. AWS KMS logs every use of its keys in AWS CloudTrail, which records all API calls to AWS KMS as events. These events include information such as who made the request, when it was made, and which key was used. The company policy can use CloudTrail logs to audit critical events related to their data encryption and access. Server-side encryption with Amazon S3 managed keys (SSE-S3) also encrypts data at rest using keys that are managed by S3, but does not provide an audit trail of key usage. Server-side encryption with customer-provided keys (SSE-C) and server-side encryption with self-managed keys also encrypt data at rest using keys that are provided or managed by customers, but do not provide an audit trail of key usage and require additional overhead for key management.

Reference: [Protecting Data Using Server-Side Encryption with AWS KMS–Managed Encryption Keys (SSE-KMS)], [Logging AWS KMS API calls with AWS CloudTrail]

NEW QUESTION 3

A developer is deploying a company's application to Amazon EC2 instances. The application generates gigabytes of data files each day. The files are rarely accessed but the files must be available to the application's users within minutes of a request during the first year of storage. The company must retain the files for 7 years.

How can the developer implement the application to meet these requirements MOST cost-effectively?

- A. Store the files in an Amazon S3 bucket. Use the S3 Glacier Instant Retrieval storage class. Create an S3 Lifecycle policy to transition the files to the S3 Glacier Deep Archive storage class after 1 year.
- B. Store the files in an Amazon S3 bucket.
- C. Use the S3 Standard storage class.
- D. Create an S3 Lifecycle policy to transition the files to the S3 Glacier Flexible Retrieval storage class after 1 year.
- E. Store the files on an Amazon Elastic Block Store (Amazon EBS) volume. Use Amazon Data Lifecycle Manager (Amazon DLM) to create snapshots of the EBS volumes and to store those snapshots in Amazon S3.
- F. Store the files on an Amazon Elastic File System (Amazon EFS) mount.
- G. Configure EFS lifecycle management to transition the files to the EFS Standard-Infrequent Access (Standard-IA) storage class after 1 year.

Answer: A

Explanation:

Amazon S3 Glacier Instant Retrieval is an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in

milliseconds. With S3 Glacier Instant Retrieval, you can save up to 68% on storage costs compared to using the S3 Standard-Infrequent Access (S3 Standard-IA) storage class, when your data is accessed once per quarter. <https://aws.amazon.com/s3/storage-classes/glacier/instant-retrieval/>

NEW QUESTION 4

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

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

- A. Create and deploy an AWS Lambda function in each desired Region
- B. Configure the Lambda function to create a stack from an AWS CloudFormation template in that Region when the function is invoked.
Create an AWS CloudFormation template that defines the load test resource
- B**: Use the AWS CLI create-stack-set command to create a stack set in the desired Regions.
- E. Create an AWS Systems Manager document that defines the resource
- F. Use the document to create the resources in the desired Regions.
- G. Create an AWS CloudFormation template that defines the load test resource
- H. Use the AWS CLI deploy command to create a stack from the template in each Region.

Answer: B

Explanation:

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

NEW QUESTION 5

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

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

- A. AWS CodeDeploy
- B. AWS CodeArtifact
- C. AWS CodeCommit
- D. Amazon CodeGuru

Answer: C

Explanation:

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

References:

? [What Is AWS CodeCommit? - AWS CodeCommit]

? [AWS CodePipeline - AWS CodeCommit]

NEW QUESTION 6

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

- A. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the AP
- B. Create a DNS A record for the custom domain.
- C. Import the SSL/TLS certificate into CloudFront
- D. Create a DNS CNAME record for the custom domain.
- E. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the same Region as the AP
- F. Create a DNS CNAME record for the custom domain.
- G. Import the SSL/TLS certificate into AWS Certificate Manager (ACM) in the us-east-1 Region
- H. Create a DNS CNAME record for the custom domain.

Answer: D

Explanation:

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

References:

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

? [What Is Amazon CloudFront? - Amazon CloudFront]

? [Custom Domain Names for APIs - Amazon API Gateway]

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 cod
- 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 is building a new application that runs on AWS and uses Amazon API Gateway to expose APIs. Teams of developers are working on separate components of the application in parallel. The company wants to publish an API without an integrated backend, so that teams that depend on the application backend can continue the development work before the API backend development is complete. Which solution will meet these requirements?

- A. Create API Gateway resources and set the integration type value to MOCK. Configure the method integration request and integration response to associate a response with an HTTP status code. Create an API Gateway stage and deploy the API.
- B. Create an AWS Lambda function that returns mocked responses and various HTTP status codes. Create an API Gateway stage and deploy the API.
- C. Create API Gateway resources and set the integration type value to AWS_PROXY. Deploy the API.
- D. Create an EC2 application that returns mocked HTTP responses. Create API Gateway resources and set the integration type value to AWS. Create an API Gateway stage and deploy the API.
- E. Create API Gateway resources and set the integration type value set to HTTP_PROXY. Create an API Gateway stage and deploy the API.
- F. Add mapping templates and deploy the API.
- G. Create an AWS Lambda layer that returns various HTTP status codes. Associate the Lambda layer with the API deployment.

Answer: A

Explanation:

The best solution for publishing an API without an integrated backend is to use the MOCK integration type in API Gateway. This allows the developer to return a static response to the client without sending the request to a backend service. The developer can configure the method integration request and integration response to associate a response with an HTTP status code, such as 200 OK or 404 Not Found. The developer can also create an API Gateway stage and deploy the API to make it available to the teams that depend on the application backend. The other solutions are either not feasible or not efficient. Creating an AWS Lambda function, an EC2 application, or an AWS Lambda layer would require additional resources and code to generate the mocked responses and HTTP status codes. These solutions would also incur additional costs and complexity, and would not leverage the built-in functionality of API Gateway. References:

? Set up mock integrations for API Gateway REST APIs

? Mock Integration for API Gateway - AWS CloudFormation

? Mocking API Responses with API Gateway

? How to mock API Gateway responses with AWS SAM

NEW QUESTION 9

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 Number that has a timestamp that is set to 48 hours after the blog post creation time.
- E. String that has a timestamp that is set to the blog post creation time.
- F. 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.
- G. Place the script in a container image.
- H. Schedule an Amazon Elastic Container Service (Amazon ECS) task on AWS Fargate that invokes the container every 5 minutes.
- I. 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.
- J. Create a global secondary index (GSI) that uses the new attribute as a sort key.
- K. 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.
- L. 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.
- M. Number that has a timestamp that is set to 48 hours after the blog post creation time.
- N. creation time. Configure the DynamoDB table with a TTL that references the new attribute.

Answer: D

Explanation:

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

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

NEW QUESTION 10

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 alias¹. A Lambda alias allows you to invoke different versions of a function using the same name¹. You can also split traffic between two aliases by assigning weights to them¹.

? In this scenario, the developer needs to test their changes in production on a small percentage of all traffic without changing the API Gateway URL. To achieve this, the developer can follow these steps:

? By using this solution, the developer can test their changes in production on a small percentage of all traffic without changing the API Gateway URL. The developer can also monitor and compare metrics between the canary and production releases, and promote or disable the canary as needed².

NEW QUESTION 10

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 15

A developer needs to migrate an online retail application to AWS to handle an anticipated increase in traffic. The application currently runs on two servers: one server for the web application and another server for the database. The web server renders webpages and manages session state in memory. The database server hosts a MySQL database that contains order details. When traffic to the application is heavy, the memory usage for the web server approaches 100% and the application slows down considerably.

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

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

- A. Use an EC2 instance to host the MySQL database
- 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 16

A developer is testing a new file storage application that uses an Amazon CloudFront distribution to serve content from an Amazon S3 bucket. The distribution accesses the S3 bucket by using an origin access identity (OAI). The S3 bucket's permissions explicitly deny access to all other users. The application prompts users to authenticate on a login page and then uses signed cookies to allow users to access their personal storage directories. The developer has configured the distribution to use its default cache behavior with restricted viewer access and has set the origin to point to the S3 bucket. However, when the developer tries to navigate to the login page, the developer receives a 403 Forbidden error. The developer needs to implement a solution to allow unauthenticated access to the login page. The solution also must keep all private content secure. Which solution will meet these requirements?

- A. Add a second cache behavior to the distribution with the same origin as the default cache behavior
- B. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted
- C. Keep the default cache behavior's settings unchanged.
- D. Add a second cache behavior to the distribution with the same origin as the default cache behavior
- E. Set the path pattern for the second cache behavior to *, and make viewer access restricted
- F. Change the default cache behavior's path pattern to the path of the login page, and make viewer access unrestricted.
- G. Add a second origin as a failover origin to the default cache behavior
- H. Point the failover origin to the S3 bucket
- I. Set the path pattern for the primary origin to *, and make viewer access restricted
- J. Set the path pattern for the failover origin to the path of the login page, and make viewer access unrestricted.
- K. Add a bucket policy to the S3 bucket to allow read access
- L. Set the resource on the policy to the Amazon Resource Name (ARN) of the login page object in the S3 bucket
- M. Add a CloudFront function to the default cache behavior to redirect unauthorized requests to the login page's S3 URL.

Answer: A

Explanation:

The solution that will meet the requirements is to add a second cache behavior to the distribution with the same origin as the default cache behavior. Set the path pattern for the second cache behavior to the path of the login page, and make viewer access unrestricted. Keep the default cache behavior's settings unchanged. This way, the login page can be accessed without authentication, while all other content remains secure and requires signed cookies. The other options either do not allow unauthenticated access to the login page, or expose private content to unauthorized users.
Reference: Restricting Access to Amazon S3 Content by Using an Origin Access Identity

NEW QUESTION 20

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

- A. Implement Kinesis Data Firehose data transformation as an AWS Lambda function
- B. Configure the function to remove the customer identifier
- C. Set an Amazon S3 bucket as the destination of the delivery stream.
- D. Launch an Amazon EC2 instance
- E. Set the EC2 instance as the destination of the delivery stream
- F. Run an application on the EC2 instance to remove the customer identifier
- G. Store the transformed data in an Amazon S3 bucket.
- H. Create an Amazon OpenSearch Service instance
- I. Set the OpenSearch Service instance as the destination of the delivery stream
- J. Use search and replace to remove the customer identifier
- K. Export the data to an Amazon S3 bucket.
- L. Create an AWS Step Functions workflow to remove the customer identifier
- M. As the last step in the workflow, store the transformed data in an Amazon S3 bucket
- N. Set the workflow as the destination of the delivery stream.

Answer: A

Explanation:

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

NEW QUESTION 23

An application that runs on AWS receives messages from an Amazon Simple Queue Service (Amazon SQS) queue and processes the messages in batches. The application sends the data to another SQS queue to be consumed by another legacy application. The legacy system can take up to 5 minutes to process some transaction data. A developer wants to ensure that there are no out-of-order updates in the legacy system. The developer cannot alter the behavior of the legacy system. Which solution will meet these requirements?

- A. Use an SQS FIFO queue
- B. Configure the visibility timeout value.
- C. Use an SQS standard queue with a SendMessageBatchRequestEntry data type
- D. Configure the DelaySeconds values.
- E. Use an SQS standard queue with a SendMessageBatchRequestEntry data type
- F. Configure the visibility timeout value.
- G. Use an SQS FIFO queue
- H. Configure the DelaySeconds value.

Answer: A

Explanation:

? An SQS FIFO queue is a type of queue that preserves the order of messages and ensures that each message is delivered and processed only once¹. This is suitable for the scenario where the developer wants to ensure that there are no out-of-order updates in the legacy system.

? The visibility timeout value is the amount of time that a message is invisible in the queue after a consumer receives it². This prevents other consumers from processing the same message simultaneously. If the consumer does not delete the message before the visibility timeout expires, the message becomes visible again and another consumer can receive it².

? In this scenario, the developer needs to configure the visibility timeout value to be longer than the maximum processing time of the legacy system, which is 5 minutes. This will ensure that the message remains invisible in the queue until the legacy system finishes processing it and deletes it. This will prevent duplicate or out-of-order processing of messages by the legacy system.

NEW QUESTION 26

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

Answer: C

Explanation:

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

NEW QUESTION 28

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 response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.
- D. Install a unit testing framework that reproduces the Lambda execution environment
- E. Create sample events based on the Lambda Documentation Invoke the handler function by using a unit testing framework
- F. Check the response Document how to run the unit testing framework for the other developers on the team
- G. Update the CI/CD pipeline to run the unit testing framework.
- H. Install the AWS Serverless Application Model (AWS SAM) CLI tool Use the `Sam local generate-event` command to generate sample events for automated test
- I. Create automated test scripts that use the `Sam local invoke` command to invoke the Lambda function
- J. Check the response Document the test scripts for the other developers on the team Update the CI/CD pipeline to run the test scripts.
- K. Create sample events based on the Lambda documentation
- L. Create a Docker container from the Node.js base image to invoke the Lambda function
- M. Check the response Document how to run the Docker container for the other developers on the team update the CI/CD pipeline to run the Docker container.

Answer: C

Explanation:

This solution will meet the requirements by using AWS SAM CLI tool, which is a command line tool that lets developers locally build, test, debug, and deploy serverless applications defined by AWS SAM templates. The developer can use `sam local generate-event` command to generate sample events for different event sources such as API Gateway or S3. The developer can create automated test scripts that use `sam local invoke` command to invoke Lambda functions locally in an environment that closely simulates Lambda environment. The developer can check the response from Lambda functions and document how to run the test

scripts for other developers on the team. The developer can also update CI/CD pipeline to run these test scripts before deploying with AWS CDK. Option A is not optimal because it will use cdk local invoke command, which does not exist in AWS CDK CLI tool. Option B is not optimal because it will use a unit testing framework that reproduces Lambda execution environment, which may not be accurate or consistent with Lambda environment. Option D is not optimal because it will create a Docker container from Node.js base image to invoke Lambda functions, which may introduce additional overhead and complexity for creating and running Docker containers.

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

NEW QUESTION 30

A developer is creating a new REST API by using Amazon API Gateway and AWS Lambda. The development team tests the API and validates responses for the known use cases before deploying the API to the production environment.

The developer wants to make the REST API available for testing by using API Gateway locally. Which AWS Serverless Application Model Command Line Interface (AWS SAM CLI) subcommand will meet these requirements?

- A. Sam local invoke
- B. Sam local generate-event
- C. Sam local start-lambda
- D. Sam local start-api

Answer: D

Explanation:

? The sam local start-api subcommand allows you to run your serverless application locally for quick development and testing¹. It creates a local HTTP server that acts as a proxy for API Gateway and invokes your Lambda functions based on the AWS SAM template¹. You can use the sam local start-api subcommand to test your REST API locally by sending HTTP requests to the local endpoint¹.

NEW QUESTION 33

A company is using an AWS Lambda function to process records from an Amazon Kinesis data stream. The company recently observed slow processing of the records. A developer notices that the iterator age metric for the function is increasing and that the Lambda run duration is constantly above normal. Which actions should the developer take to increase the processing speed? (Choose two.)

- A. Increase the number of shards of the Kinesis data stream.
- B. Decrease the timeout of the Lambda function.
- C. Increase the memory that is allocated to the Lambda function.
- D. Decrease the number of shards of the Kinesis data stream.
- E. Increase the timeout of the Lambda function.

Answer: AC

Explanation:

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

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

NEW QUESTION 38

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

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

- A. Retry the batch operation immediately.
- B. Retry the batch operation with exponential backoff and randomized delay.
- C. Update the application to use an AWS software development kit (AWS SDK) to make the requests.
- D. Increase the provisioned read capacity of the DynamoDB tables that the operation accesses.
- E. Increase the provisioned write capacity of the DynamoDB tables that the operation accesses.

Answer: BC

Explanation:

The UnprocessedKeys element indicates that the BatchGetItem operation did not process all of the requested items in the current response. This can happen if the

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

References:

? [BatchGetItem - Amazon DynamoDB]

? [Working with Queries and Scans - Amazon DynamoDB]

? [Best Practices for Handling DynamoDB Throttling Errors]

NEW QUESTION 41

A developer is using AWS Step Functions to automate a workflow. The workflow defines each step as an AWS Lambda function task. The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an ULegalArgumentException error or a TooManyRequestsException error. The developer wants the state machine to stop running when the state machine encounters an ULegalArgumentException error. The state machine needs to retry the GetResource task one additional time after 10 seconds if the state machine encounters a TooManyRequestsException error. If the second attempt fails, the developer wants the state machine to stop running.

How can the developer implement the Lambda retry functionality without adding unnecessary complexity to the state machine'?

- A. Add a Delay task after the GetResource tas
 - B. Add a catcher to the GetResource tas
 - C. Configure the catcher with an error type of TooManyRequestsExceptio
 - D. Configure the next step to be the Delay task Configure the Delay task to wait for an interval of 10 seconds Configure the next step to be the GetResource task.
 - E. Add a catcher to the GetResource task Configure the catcher with an error type of TooManyRequestsExceptio
 - F. an interval of 10 seconds, and a maximum attempts value of 1. Configure the next step to be the GetResource task.
 - G. Add a retrier to the GetResource task Configure the retrier with an error type of TooManyRequestsException, an interval of 10 seconds, and a maximum attempts value of 1.
- Duplicate the GetResource task Rename the new GetResource task to TryAgain Add a catcher to the original GetResource task
- H. Configure the catcher with an error type of TooManyRequestsExceptio
 - I. Configure the next step to be TryAgain.

Answer: C

Explanation:

The best way to implement the Lambda retry functionality is to use the Retry field in the state definition of the GetResource task. The Retry field allows the developer to specify an array of retriers, each with an error type, an interval, and a maximum number of attempts. By setting the error type to TooManyRequestsException, the interval to 10 seconds, and the maximum attempts to 1, the developer can achieve the desired behavior of retrying the GetResource task once after 10 seconds if it encounters a TooManyRequestsException error. If the retry fails, the state machine will stop running. If the GetResource task encounters an UlegalArgumentException error, the state machine will also stop running without retrying, as this error type is not specified in the Retry field. References

- ? Error handling in Step Functions
- ? Handling Errors, Retries, and adding Alerting to Step Function State Machine Executions
- ? The Jitter Strategy for Step Functions Error Retries on the New Workflow Studio

NEW QUESTION 44

A developer creates a static website for their department The developer deploys the static assets for the website to an Amazon S3 bucket and serves the assets with Amazon CloudFront The developer uses origin access control (OAC) on the CloudFront distribution to access the S3 bucket The developer notices users can access the root URL and specific pages but cannot access directories without specifying a file name. For example, /products/index.html works, but /products returns an error The developer needs to enable accessing directories without specifying a file name without exposing the S3 bucket publicly. Which solution will meet these requirements'?

- A. Update the CloudFront distribution's settings to index.html as the default root object is set Update the Amazon S3 bucket settings and enable static website hostin
- B. Specify index.html as the Index document Update the S3 bucket policy to enable acces
- D. Update the CloudFront distribution's origin to use the S3 website endpoint
- E. Create a CloudFront function that examines the request URL and appends index.html when directories are being accessed Add the function as a viewer request CloudFront function to the CloudFront distribution's behavior.
- F. Create a custom error response on the CloudFront distribution with the HTTP error code set to the HTTP 404 Not Found response code and the response page path to /index.html Set the HTTP response code to the HTTP 200 OK response code

Answer: A

Explanation:

The simplest and most efficient way to enable accessing directories without specifying a file name is to update the CloudFront distribution's settings to index.html as the default root object. This will instruct CloudFront to return the index.html object when a user requests the root URL or a directory URL for the distribution. This solution does not require enabling static website hosting on the S3 bucket, creating a CloudFront function, or creating a custom error response. References

- ? Specifying a default root object
- ? cloudfront-default-root-object-configured
- ? How to setup CloudFront default root object?
- ? Ensure a default root object is configured for AWS Cloudfront ...

NEW QUESTION 45

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 47

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions. When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment. If there are no issues, all traffic must switch over to the new version. Which change to the AWS SAM template will meet these requirements?

- A. Set the Deployment Preference Type to Canary10Percent10Minute

- AutoPublishAlias property to the Lambda alias.
- B. Set the Deployment Preference Type to LinearIOPercentEvery10Minute
 - D. Set AutoPublishAlias property to the Lambda alias.
 - E. Set the Deployment Preference Type to CanaryIOPercentIOMinute
 - F. Set the PreTraffic and PostTraffic properties to the Lambda alias.
 - G. Set the Deployment Preference Type to LinearIOPercentEveryIOMinute
 - H. Set PreTraffic and Post Traffic properties to the Lambda alias.

Answer: A

Explanation:

The AWS Serverless Application Model (AWS SAM) comes built-in with CodeDeploy to provide gradual AWS Lambda deployments¹. The DeploymentPreference property in AWS SAM allows you to specify the type of deployment that you want. The Canary10Percent10Minutes option means that 10 percent of your customer traffic is immediately shifted to your new version. After 10 minutes, all traffic is shifted to the new version¹. The AutoPublishAlias property in AWS SAM allows AWS SAM to automatically create an alias that points to the updated version of the Lambda function¹. Therefore, option A is correct.

NEW QUESTION 48

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

- A. Hardcode the credentials that are required to access the S3 objects in the application code
- B. Use the credentials to access the required S3 objects.
- C. Create a secret access key and access key ID with permission to access the S3 bucket
- D. Store the key and key ID in AWS Secrets Manager
- E. Configure the application to retrieve the Secrets Manager secret and use the credentials to access the S3 objects.
- F. Create a Lambda function execution role. Attach a policy to the role that grants access to specific objects in the S3 bucket.
- G. Create a secret access key and access key ID with permission to access the S3 bucket. Store the key and key ID as environment variables in Lambda
- H. Use the environment variables to access the required S3 objects.

Answer: C

Explanation:

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

NEW QUESTION 51

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 DB 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 54

A company needs to set up secure database credentials for all its AWS Cloud resources. The company's resources include Amazon RDS DB instances, Amazon DocumentDB clusters, and Amazon Aurora DB instances. The company's security policy mandates that database credentials be encrypted at rest and rotated at a regular interval. Which solution will meet these requirements MOST securely?

- A. Set up IAM database authentication for token-based access
- B. Generate user tokens to provide centralized access to RDS DB instances
- C. Amazon DocumentDB clusters and Aurora DB instances.
- D. Create parameters for the database credentials in AWS Systems Manager Parameter Store. Set the Type parameter to SecureString
- E. Set up automatic rotation on the parameters.
- F. Store the database access credentials as an encrypted Amazon S3 object in an S3 bucket. Block all public access on the S3 bucket. Set up automatic rotation on the encryption key.
- G. Use S3 server-side encryption to set up
- H. Create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console
- I. Create secrets for the database credentials in Secrets Manager. Set up secrets rotation on a schedule.

Answer: D

Explanation:

This solution will meet the requirements by using AWS Secrets Manager, which is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can create an AWS Lambda function by using the SecretsManagerRotationTemplate template in the AWS Secrets Manager console, which provides a sample code for rotating secrets for RDS DB instances, Amazon DocumentDB clusters, and Amazon Aurora DB instances. The developer can also create secrets for the database credentials in Secrets Manager, which encrypts them at rest and provides secure access to them. The developer can set up secrets rotation on a schedule, which changes the database credentials periodically according to a specified interval or event. Option A is not optimal because it will set up IAM database authentication for token-based access, which may not be compatible with all database engines and may require additional configuration and management of IAM roles or users. Option B is not optimal because it will create parameters for the database credentials in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option C is not optimal because it will store the database access credentials as an encrypted Amazon S3 object in an S3 bucket, which may introduce additional costs and complexity for accessing and securing the data.

References: [AWS Secrets Manager], [Rotating Your AWS Secrets Manager Secrets]

NEW QUESTION 57

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

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

- A. Encode each employee's contact information and photos using Base64. Store the information in an Amazon DynamoDB table using a sort key.
- B. Store each employee's contact information in an Amazon DynamoDB table along with the object keys for the photos stored in Amazon S3.
- C. Use Amazon Cognito user pools to implement the employee directory in a fully managed, software-as-a-service (SaaS) method.
- D. Store employee contact information in an Amazon RDS DB instance with the photos stored in Amazon Elastic File System (Amazon EFS).

Answer: B

Explanation:

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

References:

? [Amazon DynamoDB]

? [Amazon Simple Storage Service (S3)]

NEW QUESTION 62

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

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

- A. CloudFormation serverless intrinsic functions
- B. AWS Elastic Beanstalk
- C. AWS Serverless Application Model (AWS SAM)
- D. AWS Cloud Development Kit (AWS CDK)

Answer: C

Explanation:

AWS Serverless Application Model (AWS SAM) is an open-source framework that enables developers to build and deploy serverless applications on AWS. AWS SAM uses a template specification that extends AWS CloudFormation to simplify the

definition of serverless resources such as API Gateway, DynamoDB, and Lambda. The developer can use AWS SAM to define serverless resources in YAML and deploy them using the AWS SAM CLI.

References:

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

? [AWS SAM Template Specification - AWS Serverless Application Model]

NEW QUESTION 65

An ecommerce company is using an AWS Lambda function behind Amazon API Gateway

as its application tier. To process orders during checkout, the application calls a POST API from the frontend. The POST API invokes the Lambda function asynchronously. In rare situations, the application has not processed orders. The Lambda application logs show no errors or failures. What should a developer do to solve this problem?

- A. Inspect the frontend logs for API failure
- B. Call the POST API manually by using the requests from the log file.
- C. Create and inspect the Lambda dead-letter queue
- D. Troubleshoot the failed function
- E. Reprocess the events.
- F. Inspect the Lambda logs in Amazon CloudWatch for possible error
- G. Fix the errors.
- H. Make sure that caching is disabled for the POST API in API Gateway.

Answer: B

Explanation:

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

Reference: Asynchronous invocation

NEW QUESTION 69

A developer is building a microservices-based application by using Python on AWS and several AWS services. The developer must use AWS X-Ray. The developer views the service map by using the console to view the service dependencies. During testing, the developer notices that some services are missing from the service map.

What can the developer do to ensure that all services appear in the X-Ray service map?

- A. Modify the X-Ray Python agent configuration in each service to increase the sampling rate
- B. Instrument the application by using the X-Ray SDK for Python
- C. Install the X-Ray SDK for all the services that the application uses
- D. Enable X-Ray data aggregation in Amazon CloudWatch Logs for all the services that the application uses
- E. Increase the X-Ray service map timeout value in the X-Ray console

Answer: B

Explanation:

The X-Ray SDK for Python provides libraries and tools for instrumenting Python applications that use AWS services and other AWS X-Ray integrations. By installing the X-Ray SDK for all the services that the application uses, the developer can ensure that all the service dependencies are captured and displayed in the X-Ray service map. The other options are not relevant or effective for this scenario. References

? AWS X-Ray SDK for Python

? Instrumenting a Python Application

NEW QUESTION 70

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

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

- A. Deploy an Amazon EC2 instance based on Linux, and edit its `/etc/crontab` file by adding a command to periodically invoke the lambda function
- B. Configure an environment variable named `PERIOD` for the Lambda function
- C. Set the value to 600.
- D. Create an Amazon EventBridge rule that runs on a regular schedule to invoke the Lambda function.
- E. Create an Amazon Simple Notification Service (Amazon SNS) topic that has a subscription to the Lambda function with a 600-second timer.

Answer: C

Explanation:

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

Reference: Schedule AWS Lambda functions using EventBridge

NEW QUESTION 74

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

What is the PRIMARY benefit of this action?

- A. Improves legibility and systolic convention
- B. Takes advantage of runtime environment reuse
- C. Provides better error handling
- D. Creates a new SDK instance for each invocation

Answer: B

Explanation:

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

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

NEW QUESTION 79

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

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

- A. Deploy AWS X-Ray as a sidecar container to the micro service
- B. Update the task role policy to allow access to the X-Ray API.
- C. Deploy AWS X-Ray as a daemon set to the Fargate cluster
- D. Update the service role

- policy to allow access to the X-Ray API.
- E. Instrument the application by using the AWS X-Ray SD
 - F. Update the application to use the PutXrayTrace API call to communicate with the X-Ray API.
 - G. Instrument the application by using the AWS X-Ray SD
 - H. Update the application to communicate with the X-Ray daemon.
 - I. Instrument the ECS task to send the stout and spider- output to Amazon CloudWatch Log
 - J. Update the task role policy to allow the cloudwatch Putlogs action.

Answer: AE

Explanation:

The combination of steps that the developer should take to fix the errors is to deploy AWS X-Ray as a sidecar container to the microservices and instrument the ECS task to send the stdout and stderr output to Amazon CloudWatch Logs. This way, the developer can use AWS X-Ray to analyze and debug the performance of the microservices and identify any issues or bottlenecks. The developer can also use CloudWatch Logs to monitor and troubleshoot the logs from the ECS task and detect any errors or exceptions. The other options either involve using AWS X-Ray as a daemon set, which is not supported by Fargate, or using the PutTraceSegments API call, which is not necessary when using a sidecar container.

Reference: Using AWS X-Ray with Amazon ECS

NEW QUESTION 84

A company needs to distribute firmware updates to its customers around the world. Which service will allow easy and secure control of the access to the downloads at the lowest cost?

- A. Use Amazon CloudFront with signed URLs for Amazon S3.
- B. Create a dedicated Amazon CloudFront Distribution for each customer.
- C. Use Amazon CloudFront with AWS Lambda@Edge.
- D. Use Amazon API Gateway and AWS Lambda to control access to an S3 bucket.

Answer: A

Explanation:

This solution allows easy and secure control of access to the downloads at the lowest cost because it uses a content delivery network (CDN) that can cache and distribute firmware updates to customers around the world, and uses a mechanism that can restrict access to specific files or versions. Amazon CloudFront is a CDN that can improve performance, availability, and security of web applications by delivering content from edge locations closer to customers. Amazon S3 is a storage service that can store firmware updates in buckets and objects. Signed URLs are URLs that include additional information, such as an expiration date and time, that give users temporary access to specific objects in S3 buckets. The developer can use CloudFront to serve firmware updates from S3 buckets and use signed URLs to control who can download them and for how long. Creating a dedicated CloudFront distribution for each customer will incur unnecessary costs and complexity. Using Amazon CloudFront with AWS Lambda@Edge will require additional programming overhead to implement custom logic at the edge locations. Using Amazon API Gateway and AWS Lambda to control access to an S3 bucket will also require additional programming overhead and may not provide optimal performance or availability.

Reference: [Serving Private Content through CloudFront], [Using CloudFront with Amazon S3]

NEW QUESTION 88

A company wants to automate part of its deployment process. A developer needs to automate the process of checking for and deleting unused resources that supported previously deployed stacks but that are no longer used. The company has a central application that uses the AWS Cloud Development Kit (AWS CDK) to manage all deployment stacks. The stacks are spread out across multiple accounts. The developer's solution must integrate as seamlessly as possible within the current deployment process. Which solution will meet these requirements with the LEAST amount of configuration?

- A. In the central AWS CDK application, write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- B. Create an AWS CloudFormation template from a JSON fil
- C. Use the template to attach the function code to an AWS Lambda function and lo invoke the Lambda function when the deployment slack runs.
- D. In the central AWS CDK applicatio
- E. write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- F. Create an AWS CDK custom resource Use the custom resource to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.

- G. In the central AWS CDK, write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- H. Create an API in AWS Amplify Use the API to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- I. In the AWS Lambda console write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- J. Create an AWS CDK custom resource
- K. Use the custom resource to import the Lambda function into the stack and to invoke the Lambda function when the deployment stack runs.

Answer: B

Explanation:

This solution meets the requirements with the least amount of configuration because it uses a feature of AWS CDK that allows custom logic to be executed during stack deployment or deletion. The AWS Cloud Development Kit (AWS CDK) is a software development framework that allows you to define cloud infrastructure as code and provision it through CloudFormation. An AWS CDK custom resource is a construct that enables you to create resources that are not natively supported by CloudFormation or perform tasks that are not supported by CloudFormation during stack deployment or deletion. The developer can write a handler function in the code that uses AWS SDK calls to check for and delete unused resources, and create an AWS CDK custom resource that attaches the function code to a Lambda function and invokes it when the deployment stack runs. This way, the developer can automate the cleanup process without requiring additional configuration or integration. Creating a CloudFormation template from a JSON file will require additional configuration and integration with the central AWS CDK application. Creating an API in AWS Amplify will require additional configuration and integration with the central AWS CDK application and may not provide optimal performance or availability. Writing a handler function in the AWS Lambda console will require additional configuration and integration with the central AWS CDK application.

Reference: [AWS Cloud Development Kit (CDK)], [Custom Resources]

NEW QUESTION 92

A developer is migrating some features from a legacy monolithic application to use AWS Lambda functions instead. The application currently stores data in an Amazon Aurora DB cluster that runs in private subnets in a VPC. The AWS account has one VPC deployed. The Lambda functions and the DB cluster are deployed in the same AWS Region in the same AWS account.

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

Which solution will meet these requirements?

- A. Configure the DB cluster's public access setting to Yes.
- B. Configure an Amazon RDS database proxy for the Lambda functions.
- C. Configure a NAT gateway and a security group for the Lambda functions.
- D. Configure the VPC, subnets, and a security group for the Lambda functions.

Answer: D

Explanation:

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

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

NEW QUESTION 94

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

Which solution will meet these requirements MOST cost-effectively?

- A. Amazon S3 with encrypted files prefixed with "config"
- B. AWS Secrets Manager secrets with a tag that is named SecretString
- C. AWS Systems Manager Parameter Store SecureString parameters
- D. CloudFormation NoEcho parameters

Answer: C

Explanation:

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

NEW QUESTION 95

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

```
{
  "FailedRecordCount": 1,
  "Records": [
    {
      "SequenceNumber": "21269319989900637946712965403778482371",
      "ShardId": "shardId-0000000000001"
    },
    {
      "ErrorCode": "ProvisionedThroughputExceededException",
      "ErrorMessage": "Rate exceeded for shard shardId-0000000000001 in
        stream exampleStreamName under account 123456789."
    },
    {
      "SequenceNumber": "21269319989999637946712965403778482985",
      "ShardId": "shardId-0000000000002"
    }
  ]
}
```

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

- A. Implement retries with exponential backoff.
- B. Use a PutRecord API instead of PutRecords.
- C. Reduce the frequency and/or size of the requests.
- D. Use Amazon SNS instead of Kinesis.
- E. Reduce the number of KCL consumers.

Answer: AC

Explanation:

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

- ? Implement retries with exponential backoff. This will introduce randomness in the retry intervals and avoid overwhelming the shards with retries.
- ? Reduce the frequency and/or size of the requests. This will reduce the load on the shards and avoid throttling errors.
- ? Increase the number of shards in the stream. This will increase the throughput capacity of the stream and accommodate higher request rates.
- ? Use a PutRecord API instead of PutRecords. This will reduce the number of records per request and avoid exceeding the payload limit.

References:

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

NEW QUESTION 97

A company is migrating its PostgreSQL database into the AWS Cloud. The company wants to use a database that will secure and regularly rotate database credentials. The company wants a solution that does not require additional programming overhead.

Which solution will meet these requirements?

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

This solution meets the requirements because it uses a PostgreSQL-compatible database that can secure and regularly rotate database credentials without requiring additional programming overhead. Amazon Aurora PostgreSQL is a relational database service that is compatible with PostgreSQL and offers high performance, availability, and scalability. AWS Secrets Manager is a service that helps you protect secrets needed to access your applications, services, and IT resources. You can store database credentials in AWS Secrets Manager and use them to access your Aurora PostgreSQL database. You can also enable automatic rotation of your secrets according to a schedule or an event. AWS Secrets Manager handles the complexity of rotating secrets for you, such as generating new passwords and updating your database with the new credentials. Using Amazon DynamoDB for the database will not meet the requirements because it is a NoSQL database that is not compatible with PostgreSQL. Using AWS Systems Manager Parameter Store for storing and rotating database credentials will require additional programming overhead to integrate with your database.

Reference: [What Is Amazon Aurora?], [What Is AWS Secrets Manager?]

NEW QUESTION 102

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

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

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

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

- F. Encrypt the access token by using an AWS Key Management Service (AWS KMS) customer managed ke
- G. Store the access token in an Amazon DynamoDB tabl
- H. Update the IAM role of the EC2 instances with permissions to access DynamoDB and AWS KM
- I. Retrieve the token from DynamoD
- J. Decrypt the token by using AWS KMS on the EC2 instance
- K. Use the decrypted access token to send the message to the chat.
- L. Use AWS Secrets Manager with an AWS Key Management Service (AWS KMS) customer managed key to store the access toke
- M. Add a resource-based policy to the secret to allow access from other account
- N. Update the IAM role of the EC2 instances with permissions to access Secrets Manage
- O. Retrieve the token from Secrets Manage
- P. Use the decrypted access token to send the message to the chat.
- Q. Encrypt the access token by using an AWS Key Management Service (AWS KMS) AWS managed ke
- R. Store the access token in an Amazon S3 bucke
- S. Add a bucket policy to the S3 bucket to allow access from other account
- T. Update the IAM role of the EC2 instances with permissions to access Amazon S3 and AWS KM
- . Retrieve the token from the S3 bucke
- . Decrypt the token by using AWS KMS on the EC2 instance
- . Use the decrypted access token to send the message to the chat.

Answer: C

Explanation:

<https://aws.amazon.com/premiumsupport/knowledge-center/secrets-manager-share-between-accounts/>
https://docs.aws.amazon.com/secretsmanager/latest/userguide/auth-and-access_examples_cross.html

NEW QUESTION 107

A developer is creating an AWS Lambda function that searches for items from an Amazon DynamoDB table that contains customer contact information- The DynamoDB table items have the customer's email_address as the partition key and additional properties such as customer_type, name, and job_title. The Lambda function runs whenever a user types a new character into the customer_type text input The developer wants the search to return partial matches of all the email_address property of a particular customer_type The developer does not want to recreate the DynamoDB table. What should the developer do to meet these requirements?

- A. Add a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key Perform a query operation on the GSI by using the begins_with key condition expression With the email_address property
- B. Add a global secondary index (GSI) to the DynamoDB table With email_address as the partition key and customer_type as the sort key Perform a query operation on the GSI by using the begins_with key condition expression With the email_address property.
- C. Add a local secondary index (LSI) to the DynamoDB table With customer_type as the partition key and email_address as the sort key Perform a query operation on the LSI by using the begins_with key condition expression With the email_address property
- D. Add a local secondary Index (LSI) to the DynamoDB table With job_title as the partition key and email_address as the sort key Perform a query operation on the LSI by using the begins_with key condition expression With the email_address property

Answer: A

Explanation:

By adding a global secondary index (GSI) to the DynamoDB table with customer_type as the partition key and email_address as the sort key, the developer can perform a query operation on the GSI using the Begins_with key condition expression with the email_address property. This will return partial matches of all email_address properties of a specific customer_type.

NEW QUESTION 109

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

- A. All at once
- B. Rolling with additional batch
- C. Bluegreen
- D. Immutable

Answer: B

Explanation:

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

References: AWS Elastic Beanstalk Deployment Policies

NEW QUESTION 111

A developer has code that is stored in an Amazon S3 bucket. The code must be deployed as an AWS Lambda function across multiple accounts in the same AWS Region as the S3 bucket an AWS CloudFormation template that runs for each account will deploy the Lambda function. What is the MOST secure way to allow CloudFormation to access the Lambda Code in the S3 bucket?

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

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

NEW QUESTION 112

A company is building a micro services application that consists of many AWS Lambda functions. The development team wants to use AWS Serverless Application Model (AWS SAM) templates to automatically test the Lambda functions. The development team plans to test a small percentage of traffic that is directed to new updates before the team commits to a full deployment of the application.

Which combination of steps will meet these requirements in the MOST operationally efficient way? (Select TWO.)

- A. Use AWS SAM CLI commands in AWS CodeDeploy to invoke the Lambda functions to test the deployment
- B. Declare the EventInvokeConfig on the Lambda functions in the AWS SAM templates with OnSuccess and OnFailure configurations.
Enable gradual deployments through AWS SAM templates.
- ☒ C. Set the deployment preference type to Canary10Percent130Minutes Use hooks to test the deployment.
- E. Set the deployment preference type to Linear10PercentEvery10Minutes Use hooks to test the deployment.

Answer: CD

Explanation:

This solution will meet the requirements by using AWS Serverless Application Model (AWS SAM) templates and gradual deployments to automatically test the Lambda functions. AWS SAM templates are configuration files that define serverless applications and resources such as Lambda functions. Gradual deployments are a feature of AWS SAM that enable deploying new versions of Lambda functions incrementally, shifting traffic gradually, and performing validation tests during deployment. The developer can enable gradual deployments through AWS SAM templates by adding a DeploymentPreference property to each Lambda function resource in the template. The developer can set the deployment preference type to Canary10Percent30Minutes, which means that 10 percent of traffic will be shifted to the new version of the Lambda function for 30 minutes before shifting 100 percent of traffic. The developer can also use hooks to test the deployment, which are custom Lambda functions that run before or after traffic shifting and perform validation tests or rollback actions.

References: [AWS Serverless Application Model (AWS SAM)], [Gradual Code Deployment]

NEW QUESTION 117

.....

THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual DVA-C02 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the DVA-C02 Product From:

<https://www.2passeasy.com/dumps/DVA-C02/>

Money Back Guarantee

DVA-C02 Practice Exam Features:

- * DVA-C02 Questions and Answers Updated Frequently
- * DVA-C02 Practice Questions Verified by Expert Senior Certified Staff
- * DVA-C02 Most Realistic Questions that Guarantee you a Pass on Your FirstTry
- * DVA-C02 Practice Test Questions in Multiple Choice Formats and Updatesfor 1 Year