

Amazon

Exam Questions DVA-C02

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 wants to add request validation to a production environment Amazon API Gateway API. The developer needs to test the changes before the API is deployed to the production environment. For the test, the developer will send test requests to the API through a testing tool. Which solution will meet these requirements with the LEAST operational overhead?

- A. Export the existing API to an OpenAPI file.
- B. Create a new API. Modify the new API to add request validation.
- C. Import the OpenAPI file.
- D. Perform the test.
- E. Modify the existing API to add request validation.
- F. Deploy the existing API to production.
- G. Modify the existing API to add request validation.
- H. Deploy the updated API to a new API Gateway stage.
- I. Perform the test.
- J. Deploy the updated API to the API Gateway production stage.
- K. Create a new API.
- L. Add the necessary resources and methods, including new request validation.
- M. Perform the test.
- N. Modify the existing API to add request validation.
- O. Deploy the existing API to production.
- P. Clone the existing API.
- Q. Modify the new API to add request validation.
- R. Perform the test.
- S. Modify the existing API to add request validation.
- T. Deploy the existing API to production.

Answer: B

Explanation:

Amazon API Gateway allows you to create, deploy, and manage a RESTful API to expose backend HTTP endpoints, AWS Lambda functions, or other AWS services¹. You can use API Gateway to perform basic validation of an API request before proceeding with the integration request¹. When the validation fails, API Gateway immediately fails the request, returns a 400 error response to the caller, and publishes the validation results in CloudWatch Logs¹.

To test changes before deploying to a production environment, you can modify the existing API to add request validation and deploy the updated API to a new API Gateway stage¹. This allows you to perform tests without affecting the production environment. Once testing is complete and successful, you can then deploy the updated API to the API Gateway production stage¹.

This approach has the least operational overhead as it avoids unnecessary creation of new APIs or exporting and importing of APIs. It leverages the existing infrastructure and only requires changes in the configuration of the existing API¹.

NEW QUESTION 3

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

- A. Manually instrument the X-Ray SDK in the application code.
- B. Use the X-Ray auto-instrumentation agent.
- C. Use Amazon Macie to detect and hide PII.
- D. Call the X-Ray API from AWS Lambda.
- E. Use AWS Distro for Open Telemetry.

Answer: A

Explanation:

This solution will meet the requirements by allowing the developer to control what data is sent to X-Ray and CloudWatch from the application code. The developer

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

References: [AWS X-Ray SDKs]

NEW QUESTION 4

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

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

- A. Create an AWS Lambda function
- B. Use API Gateway proxy integration to return constant HTTP responses.
- C. Create an Amazon EC2 instance that serves the backend REST API by using an AWS CloudFormation template.
- D. Customize the API Gateway stage to select a response type based on the request.
- E. Use a request mapping template to select the mock integration response.

Answer: D

Explanation:

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

<https://docs.aws.amazon.com/apigateway/latest/developerguide/how-to-mock-integration.html>

NEW QUESTION 5

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 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 API
- 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 API
- 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 company is using Amazon OpenSearch Service to implement an audit monitoring system. A developer needs to create an AWS CloudFormation custom

resource that is

associated with an AWS Lambda function to configure the OpenSearch Service domain. The Lambda function must access the OpenSearch Service domain by using Open Search Service internal master user credentials. What is the MOST secure way to pass these credentials to the Lambdas function?

- A. Use a CloudFormation parameter to pass the master user credentials at deployment to the OpenSearch Service domain's MasterUserOptions and the Lambda function's environment variable
- 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
- K. We Encrypt the parameter's value by using the AWS Key Management Service (AWS KMS) encrypt command.
- L. Use CloudFormation to create an AWS Secrets Manager secret
- M. Use a CloudFormation dynamic reference to retrieve the secret's value for the OpenSearch Service domain's MasterUserOptions
- N. Create an IAM role that has the secretsmanager: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 8

A developer needs to deploy an application running on AWS Fargate using Amazon ECS. The application has environment variables that must be passed to a container for the application to initialize.

How should the environment variables be passed to the container?

- A. Define an array that includes the environment variables under the environment parameter within the service definition.
- B. Define an array that includes the environment variables under the environment parameter within the task definition.
- C. Define an array that includes the environment variables under the entryPoint parameter within the task definition.
- D. Define an array that includes the environment variables under the entryPoint parameter within the service definition.

Answer: B

Explanation:

This solution allows the environment variables to be passed to the container when it is launched by AWS Fargate using Amazon ECS. The task definition is a text file that describes one or more containers that form an application. It contains various parameters for configuring the containers, such as CPU and memory requirements, network mode, and environment variables. The environment parameter is an array of key-value pairs that specify environment variables to pass to a container. Defining an array that includes the environment variables under the entryPoint parameter within the task definition

will not pass them to the container, but use them as command-line arguments for overriding the default entry point of a container.

Defining an array that includes the environment variables under the environment or entryPoint parameter within the service definition will not pass them to the container, but cause an error because these parameters are not valid for a service definition.

Reference: [Task Definition Parameters], [Environment Variables]

NEW QUESTION 9

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 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 Number that has a timestamp that is set to 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. Create a global secondary index (GSI) that uses the new attribute as a sort key. 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 at a company recently created a serverless application to process and show data from business reports. The application's user interface (UI) allows users to select and start processing the files. The UI displays a message when the result is available to view. The application uses AWS Step Functions with AWS Lambda functions to process the files. The developer used Amazon API Gateway and Lambda functions to create an API to support the UI.

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

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

- A. Change the API Gateway route to add an X-Amz-Invocation-Type header with a static value of 'Event' in the integration request. Deploy the API Gateway stage to apply the changes.
- B. Change the configuration of the Lambda function that implements the request to process a file to run asynchronously.
- C. Configure the maximum age of the event so that the Lambda function will run asynchronously.
- D. Change the API Gateway timeout value to match the Lambda function timeout value.
- E. Deploy the API Gateway stage to apply the changes.
- F. Change the API Gateway route to add an X-Amz-Target header with a static value of 'A sync' in the integration request. Deploy the API Gateway stage to apply the changes.

Answer: A

Explanation:

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

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

NEW QUESTION 16

A company is using AWS CloudFormation to deploy a two-tier application. The application will use Amazon RDS as its backend database. The company wants a solution that will randomly generate the database password during deployment. The solution also must automatically rotate the database password without requiring changes to the application.

What is the MOST operationally efficient solution that meets these requirements'?

- A. Use an AWS Lambda function as a CloudFormation custom resource to generate and rotate the password.
- B. Use an AWS Systems Manager Parameter Store resource with the SecureString data type to generate and rotate the password.
- C. Use a cron daemon on the application's host to generate and rotate the password.
- D. Use an AWS Secrets Manager resource to generate and rotate the password.

Answer: D

Explanation:

This solution will meet the requirements by using AWS Secrets Manager, which is a service that helps protect secrets such as database credentials by encrypting them with AWS Key Management Service (AWS KMS) and enabling automatic rotation of secrets. The developer can use an AWS Secrets Manager resource in AWS CloudFormation template, which enables creating and managing secrets as part of a CloudFormation stack. The developer can use an AWS::SecretsManager::Secret resource type to generate and rotate the password for accessing RDS database during deployment. The developer can also specify a RotationSchedule property for the secret resource, which defines how often to rotate the secret and which Lambda function to use for rotation logic. Option A is not optimal because it will use an AWS Lambda function as a CloudFormation custom resource, which may introduce additional complexity and overhead for creating and managing a custom resource and implementing rotation logic. Option B is not optimal because it will use an AWS Systems Manager Parameter Store resource with the SecureString data type, which does not support automatic rotation of secrets. Option C is not optimal because it will use a cron daemon on the application's host to generate and rotate the password, which may incur more costs and require more maintenance for running and securing a host.

References: [AWS Secrets Manager], [AWS::SecretsManager::Secret]

NEW QUESTION 17

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 18

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 20

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 of the new 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 to read from the DynamoDB table and write to the S3 bucket.
- H. Associate the stream's Amazon Resource Name (ARN) with the Lambda function.
- I. Modify the Lambda function to publish to a new Amazon SNS topic.
- J. Create a new Lambda function that subscribes to the SNS topic and writes to the S3 bucket.
- 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 25

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 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 for the other developers on the team
- F. Check the response. Document how to run the unit testing framework.
- 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 testing
- 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 33

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 35

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 38

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 43

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

Dynamic application security testing occurs in the final stage of the pipeline after a new image is deployed to a development namespace in the EKS cluster. A developer needs to

place an analysis stage before this deployment to analyze the container image earlier in the CI/CD pipeline.

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

- A. Build the container image and run the docker scan command locally
- B. Mitigate any findings before pushing changes to the source code repository
- C. Write a pre-commit hook that enforces the use of this workflow before commit.
- D. Create a new CodePipeline stage that occurs after the container image is built
- E. Configure ECR basic image scanning to scan on image push
- F. Use an AWS Lambda function as the action provider
- G. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.
- H. Create a new CodePipeline stage that occurs after source code has been retrieved from its repository
- I. Run a security scanner on the latest revision of the source code
- J. Fail the pipeline if there are findings.
- K. Add an action to the deployment stage of the pipeline so that the action occurs before the deployment to the EKS cluster
- L. Configure ECR basic image scanning to scan on image push
- M. Use an AWS Lambda function as the action provider
- N. Configure the Lambda function to check the scan results and to fail the pipeline if there are findings.

Answer: B

Explanation:

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

Reference: Amazon ECR image scanning

NEW QUESTION 45

A company is building a compute-intensive application that will run on a fleet of Amazon EC2 instances. The application uses attached Amazon Elastic Block Store (Amazon EBS) volumes for storing data. The Amazon EBS volumes will be created at time of initial deployment. The application will process sensitive information. All of the data must be encrypted. The solution should not impact the application's performance.

Which solution will meet these requirements?

- A. Configure the fleet of EC2 instances to use encrypted EBS volumes to store data.
- B. Configure the application to write all data to an encrypted Amazon S3 bucket.
- C. Configure a custom encryption algorithm for the application that will encrypt and decrypt all data.
- D. Configure an Amazon Machine Image (AMI) that has an encrypted root volume and store the data to ephemeral disks.

Answer: A

Explanation:

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with Amazon EC2 instances¹. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources associated with your EC2 instances¹. When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted: Data at rest inside the volume, all data moving between the volume and the instance, all snapshots created from the volume, and all volumes created from those snapshots¹. Therefore, option A is correct.

NEW QUESTION 48

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

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

- A. Create a separate CloudFormation template for each EC2 instance type in the list.
- B. In the Resources section of the CloudFormation template, create resources for each EC2 instance type in the list.
- C. In the CloudFormation template, create a separate parameter for each EC2 instance type in the list.
- D. In the CloudFormation template, create a parameter with the list of EC2 instance types as AllowedValues.

Answer: D

Explanation:

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

NEW QUESTION 49

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 51

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 a 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 task.
- B. Add a catcher to the GetResource task.
- C. Configure the catcher with an error type of TooManyRequestsException.
- 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 TooManyRequestsException.
- 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.
- H. Duplicate the GetResource task. Rename the new GetResource task to TryAgain. Add a catcher to the original GetResource task.
- I. Configure the catcher with an error type of TooManyRequestsException.
- J. 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 52

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 56

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 61

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 65

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 66

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

Which change to the AWS SAM template will meet these requirements?

- A. Set the Deployment Preference Type to Canaryl OPercent10Minute
- B. Set the AutoPublishAlias property to the Lambda alias.
- C. Set the Deployment Preference Type to Linearl OPercentEverylOMinute
- D. Set AutoPublishAlias property to the Lambda alias.
- E. Set the Deployment Preference Type to Canaryl OPercentlOMinute
- F. Set the PreTraffic and PostTraffic properties to the Lambda alias.
- G. Set the Deployment Preference Type to Linearl OPercentEvery10Minute
- H. Set PreTraffic and PostTraffic properties to the Lambda alias.

Answer: A

Explanation:

? The Deployment Preference Type property specifies how traffic should be shifted between versions of a Lambda 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 Deployment Preference Type 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 68

A company has an application that stores data in Amazon RDS instances. The application periodically experiences surges of high traffic that cause performance problems.

During periods of peak traffic, a developer notices a reduction in query speed in all database queries.

The team's technical lead determines that a multi-threaded and scalable caching solution should be used to offload the heavy read traffic. The solution needs to improve performance.

Which solution will meet these requirements with the LEAST complexity?

- A. Use Amazon ElastiCache for Memcached to offload read requests from the main database.
- B. Replicate the data to Amazon DynamoD
- C. Set up a DynamoDB Accelerator (DAX) cluster.
- D. Configure the Amazon RDS instances to use Multi-AZ deployment with one standby instance
- E. Offload read requests from the main database to the standby instance.
- F. Use Amazon ElastiCache for Redis to offload read requests from the main database.

Answer: A

Explanation:

? Amazon ElastiCache for Memcached is a fully managed, multithreaded, and scalable in-memory key-value store that can be used to cache frequently accessed data and improve application performance¹. By using Amazon ElastiCache for Memcached, the developer can reduce the load on the main database and handle high traffic surges more efficiently.

? To use Amazon ElastiCache for Memcached, the developer needs to create a cache cluster with one or more nodes, and configure the application to store and retrieve data from the cache cluster². The developer can use any of the supported Memcached clients to interact with the cache cluster³. The developer can also use Auto Discovery to dynamically discover and connect to all cache nodes in a cluster⁴.

? Amazon ElastiCache for Memcached is compatible with the Memcached protocol, which means that the developer can use existing tools and libraries that work with

Memcached¹. Amazon ElastiCache for Memcached also supports data partitioning, which allows the developer to distribute data among multiple nodes and scale out the cache cluster as needed.

? Using Amazon ElastiCache for Memcached is a simple and effective solution that meets the requirements with the least complexity. The developer does not need to change the database schema, migrate data to a different service, or use a different caching model. The developer can leverage the existing Memcached ecosystem and easily integrate it with the application.

NEW QUESTION 69

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 70

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
- B. Use the plaintext key from the GenerateDataKey API and a symmetric encryption algorithm 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 key¹. 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 KMS¹.

? In this scenario, the developer needs to use the GenerateDataKey API to encrypt the PDF file so that it can be decrypted later. The developer also needs to use an AWS KMS symmetric customer managed key for encryption. To achieve this, the developer can follow these steps:

NEW QUESTION 74

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 77

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

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

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

- A. Host each website by using AWS Amplify with a serverless backen
- B. Conned the repository branches that correspond to each of the desired environment
- C. Start deployments by merging code changes to a desired branch.
- D. Host each website in AWS Elastic Beanstalk with multiple environment
- E. Use the EB CLI to link each repository branc
- F. Integrate AWS CodePipeline to automate deployments from version control code merges.
- G. Host each website in different Amazon S3 buckets for each environmen
- H. Configure AWS CodePipeline to pull source code from version contro
- I. Add an AWS CodeBuild stage to copy source code to Amazon S3.
- J. Host each website on its own Amazon EC2 instanc
- K. Write a custom deployment script to bundle each website's static asset
- L. Copy the assets to Amazon EC2. Set up a workflow to run the script when code is merged.

Answer: A

Explanation:

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

NEW QUESTION 81

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 stac
- D. Create a table in the database for parameter configuratio
- 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 86

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

Which solution should the developer implement to meet these requirements?

- A. Run the amplify add test command in the Amplify CLI.
- B. Create unit tests in the applicatio
- C. Deploy the unit tests by using the amplify push command in the Amplify CLI.
- D. Add a test phase to the amplify.yml build settings for the application.
- E. Add a test phase to the aws-exports.js file for the application.

Answer: C

Explanation:

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

Reference: End-to-end testing

NEW QUESTION 87

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 91

A developer uses AWS CloudFormation to deploy an Amazon API Gateway API and an AWS Step Functions state machine. The state machine must reference the API Gateway API after the CloudFormation template is deployed. The developer needs a solution that uses the state machine to reference the API Gateway endpoint.

Which solution will meet these requirements MOST cost-effectively?

- A. Configure the CloudFormation template to reference the API endpoint in the DefinitionSubstitutions property for the AWS StepFunctions StateMachine resource.
- B. Configure the CloudFormation template to store the API endpoint in an environment variable for the AWS::StepFunctions::StateMachine resource. Configure the state machine to reference the environment variable.
- C. Configure the CloudFormation template to store the API endpoint in a standard AWS: SecretsManager Secret resource. Configure the state machine to reference the resource.
- D. Configure the CloudFormation template to store the API endpoint in a standard AWS::AppConfig::ConfigurationProfile resource. Configure the state machine to reference the resource.

Answer: A

Explanation:

The most cost-effective solution is to use the DefinitionSubstitutions property of the AWS::StepFunctions::StateMachine resource to inject the API endpoint as a variable in the state machine definition. This way, the developer can use the intrinsic function

`Fn::GetAtt` to get the API endpoint from the AWS::ApiGateway::RestApi resource, and pass it to the state machine without creating any additional resources or environment variables. The other solutions involve creating and managing extra resources, such as Secrets Manager secrets or AppConfig configuration profiles, which incur additional costs and complexity. References

? AWS::StepFunctions::StateMachine - AWS CloudFormation

? Call API Gateway with Step Functions - AWS Step Functions

? amazon-web-services aws-api-gateway terraform aws-step-functions

NEW QUESTION 94

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 99

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

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

- A. Install and run the X-Ray SDK on the on-premises servers to capture and relay the data to the X-Ray service.
- B. Install and run the X-Ray daemon on the on-premises servers to capture and relay the data to the X-Ray service.
- C. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the PutTraceSegments API call.
- D. Capture incoming requests on-premises and configure an AWS Lambda function to pull, process, and relay relevant data to X-Ray using the PutTelemetryRecords API call.

Answer: B

Explanation:

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

References:

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

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

NEW QUESTION 104

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 107

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 110

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.

K. Perform the tests

Modify the existing API to add request validation Deploy the existing API to production.

Answer: D

Explanation:

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

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

NEW QUESTION 114

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 119

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 122

A developer is creating a serverless application that uses an AWS Lambda function. The developer will use AWS CloudFormation to deploy the application. The application will write logs to Amazon CloudWatch Logs. The developer has created a log group in a CloudFormation template for the application to use. The developer needs to modify the CloudFormation template to make the name of the log group available to the application at runtime. Which solution will meet this requirement?

- A. Use the `AWS::Include` transform in CloudFormation to provide the log group's name to the application.
- B. Pass the log group's name to the application in the user data section of the CloudFormation template.
- C. Use the CloudFormation template's Mappings section to specify the log group's name for the application.
- D. Pass the log group's Amazon Resource Name (ARN) as an environment variable to the Lambda function.

Answer: D

Explanation:

FunctionName: MyLambdaFunction Code:

S3Bucket: your-lambda-code-bucket S3Key: lambda-code.zip

Runtime: nodejs14.x # Specify the desired runtime for your Lambda function Environment:

Variables:

LOG_GROUP_NAME: !Ref MyLogGroup <https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-logs-loggroup.html>

NEW QUESTION 127

A developer is troubleshooting an application in an integration environment. In the application, an Amazon Simple Queue Service (Amazon SQS) queue consumes messages and then an AWS Lambda function processes the messages. The Lambda function transforms the messages and makes an API call to a third-party service.

There has been an increase in application usage. The third-party API frequently returns an HTTP 429 Too Many Requests error message. The error message prevents a significant number of messages from being processed successfully.

How can the developer resolve this issue?

- A. Increase the SQS event source's batch size setting.
- B. Configure provisioned concurrency for the Lambda function based on the third-party API's documented rate limits.
- C. Increase the retry attempts and maximum event age in the Lambda function's asynchronous configuration.
- D. Configure maximum concurrency on the SQS event source based on the third-party service's documented rate limits.

Answer: D

Explanation:

? Maximum concurrency for SQS as an event source allows customers to control the maximum concurrent invokes by the SQS event source¹. When multiple SQS event sources are configured to a function, customers can control the maximum concurrent invokes of individual SQS event source¹.

? In this scenario, the developer needs to resolve the issue of the third-party API frequently returning an HTTP 429 Too Many Requests error message, which prevents a significant number of messages from being processed successfully. To achieve this, the developer can follow these steps:

? By using this solution, the developer can reduce the frequency of HTTP 429 errors and improve the message processing success rate. The developer can also avoid throttling or blocking by the third-party API.

NEW QUESTION 131

A developer is creating an AWS Lambda function. The Lambda function needs an external library to connect to a third-party solution. The external library is a collection of files with a total size of 100 MB. The developer needs to make the external library available to the Lambda execution environment and reduce the Lambda package space.

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

A.

Create a Lambda layer to store the external library. Configure the Lambda function to use the layer.

- B. Create an Amazon S3 bucket. Upload the external library into the S3 bucket.
- C. Mount the S3 bucket folder in the Lambda function. Import the library by using the proper folder in the mount point.
- D. Load the external library to the Lambda function's /tmp directory during deployment of the Lambda package.
- E. Import the library from the /tmp directory.
- F. Create an Amazon Elastic File System (Amazon EFS) volume.
- G. Upload the external library to the EFS volume. Mount the EFS volume in the Lambda function.
- H. Import the library by using the proper folder in the mount point.

Answer: A

Explanation:

Create a Lambda layer to store the external library. Configure the Lambda function to use the layer. This will allow the developer to make the external library available to the Lambda execution environment without having to include it in the Lambda package, which will reduce the Lambda package space. Using a Lambda layer is a simple and straightforward solution that requires minimal operational overhead. <https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html>

NEW QUESTION 135

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 140

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

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

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

- A. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install all the patches and agents that are needed to manage and run the applicatio
- B. Update the Auto Scaling group launch configuration to use the AMI.
- C. Use EC2 Image Builder to create an Amazon Machine Image (AMI). Install the latest version of the application and all the patches and agents that are needed to manage and run the applicatio
- D. Update the Auto Scaling group launch configuration to use the AMI.
- E. Set up AWS CodeDeploy to deploy the most recent version of the application at runtime.
- F. Set up AWS CodePipeline to deploy the most recent version of the application at runtime.
- G. Remove any commands that perform operating system patching from the UserData script.

Answer: BE

Explanation:

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

? Set up AWS CodeDeploy to deploy the most recent version of the application at

runtime. This will ensure that the application code is always up to date and does not depend on the AMI.

? Remove any commands that perform operating system patching from the

UserData script. This will reduce the time that the UserData script takes to run and speed up the instance launch process.

References:

? [What Is AWS CloudFormation? - AWS CloudFormation]

? [What Is AWS CodeDeploy? - AWS CodeDeploy]

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

NEW QUESTION 142

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

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

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

For a given title, get all details about all movies that have that title. For a given genre, get all details about all movies in that genre. Which data store configuration will meet these requirements?

- A. Create an Amazon DynamoDB table.
- B. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort key.
- C. Create a global secondary index that uses the genre as the partition key and the title as the sort key.
- D. Create an Amazon RDS database instance.
- E. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort key.
- F. Create a global secondary index that uses the title as the partition key.
- G. On an Amazon RDS database instance, create a table that contains columns for title, release year, and genre.
- H. Configure the title as the primary key.
- I. On an Amazon RDS database instance, create a table where the primary key is the title and all other data is encoded into JSON format as one additional column.

Answer: A

Explanation:

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. The developer can create a DynamoDB table and configure the table with a primary key that consists of the title as the partition key and the release year as the sort key. This will enable querying for a given title and release year efficiently. The developer can also create a global secondary index that uses the genre as the partition key and the title as the sort key. This will enable querying for a given genre efficiently. The developer can store additional properties about the cast and production crew as attributes in the DynamoDB table. These attributes can have different data types and structures, and they do not need to be consistent across items.

References:

? [Amazon DynamoDB]

? [Working with Queries - Amazon DynamoDB]

? [Working with Global Secondary Indexes - Amazon DynamoDB]

NEW QUESTION 145

A company hosts its application on AWS. The application runs on an Amazon Elastic Container Service (Amazon ECS) cluster that uses AWS Fargate. The cluster runs behind an Application Load Balancer. The application stores data in an Amazon Aurora database. A developer encrypts and manages database credentials inside the application.

The company wants to use a more secure credential storage method and implement periodic credential rotation. Which solution will meet these requirements with the LEAST operational overhead?

- A. Migrate the secret credentials to Amazon RDS parameter group
- B. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- C. Use IAM policies and roles to grant AWS KMS permissions to access Amazon RDS.
- D. Migrate the credentials to AWS Systems Manager Parameter Store.
- E. Encrypt the parameter by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- F. Turn on secret rotation.
- G. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- H. Migrate the credentials to ECS Fargate environment variable.
- I. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation.
- J. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager.
- K. Migrate the credentials to AWS Secrets Manager.
- L. Encrypt the credentials by using an AWS Key Management Service (AWS KMS) key. Turn on secret rotation. Use IAM policies and roles to grant Amazon ECS Fargate permissions to access to AWS Secrets Manager by using keys.

Answer: D

Explanation:

AWS Secrets Manager is a service that helps you store, distribute, and rotate secrets securely. You can use Secrets Manager to migrate your credentials from your application code to a secure and encrypted storage. You can also enable automatic rotation of your secrets by using AWS Lambda functions or custom logic. You can use IAM policies and roles to grant your Amazon ECS Fargate tasks permissions to access your secrets from Secrets Manager. This solution minimizes the operational overhead of managing your credentials and enhances the security of your application. References

? AWS Secrets Manager: Store, Distribute, and Rotate Credentials Securely | AWS

News Blog

? Why You Should Audit and Rotate Your AWS Credentials Periodically - Cloud Academy

? Top 5 AWS root account best practices - TheServerSide

NEW QUESTION 149

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

- A. Change the Auto Scaling group to six desired instances.
- B. Change the deployment policy to traffic splittin
- C. Specify an evaluation time of 1 hour.
- D. Change the deployment policy to rolling with additional batc
- E. Specify a batch size of 1.
- F. Change the deployment policy to rollin
- G. Specify a batch size of 2.

Answer: C

Explanation:

This solution will solve the deployment issue by deploying the new version of the application to one new EC2 instance at a time, while keeping the old version running on

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

References: AWS Elastic Beanstalk Deployment Policies

NEW QUESTION 154

A company is preparing to migrate an application to the company's first AWS environment Before this migration, a developer is creating a proof-of-concept application to validate a model for building and deploying container-based applications on AWS.

Which combination of steps should the developer take to deploy the containerized proof-of- concept application with the LEAST operational effort? (Select TWO.)

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

To deploy a containerized application on AWS with the least operational effort, the developer should package the application into a container image by using the Docker CLI and upload the image to Amazon ECR, which is a fully managed container registry service. Then, the developer should deploy the application to Amazon ECS on AWS Fargate, which is a serverless compute engine for containers that eliminates the need to provision and manage servers or clusters. Amazon ECS will automatically scale, load balance, and monitor the application. References

? How to Deploy Docker Containers | AWS

? Deploy a Web App Using AWS App Runner

? How to Deploy Containerized Apps on AWS Using ECR and Docker

NEW QUESTION 158

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

- A. The developer must manually keep Hack of the data encryption keys used for each data object.
- B. The SDK encrypts the data encryption key and stores it (encrypted) as part of the resumed ophertext.
- C. The SDK stores the data encryption keys automaticity in Amazon S3.
- D. The data encryption key is stored m the user data for the EC2 instance.

Answer: B

Explanation:

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

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

NEW QUESTION 160

A company has developed a new serverless application using AWS Lambda functions that will be deployed using the AWS Serverless Application Model (AWS SAM) CLI.

Which step should the developer complete prior to deploying the application?

- A. Compress the application to a zip file and upload it into AWS Lambda.
- B. Test the new AWS Lambda function by first tracing it in AWS X-Ray.
- C. Bundle the serverless application using a SAM package.
- D. Create the application environment using the `eb create my-env` command.

Answer: C

Explanation:

This step should be completed prior to deploying the application because it prepares the application artifacts for deployment. The AWS Serverless Application Model (AWS SAM) is a framework that simplifies building and deploying serverless applications on AWS. The AWS SAM CLI is a command-line tool that helps you create, test, and deploy serverless applications using AWS SAM templates. The `sam package` command bundles the application artifacts, such as Lambda function code and API definitions, and uploads them to an Amazon S3 bucket. The command also returns a CloudFormation template that is ready to be deployed with the `sam deploy` command. Compressing the application to a zip file and uploading it to AWS Lambda will not work because it does not use AWS SAM templates or CloudFormation. Testing the new Lambda function by first tracing it in AWS X-Ray will not prepare the application for deployment, but only monitor its performance and errors. Creating the application environment using the `eb create my-env` command will not work because it is a command for AWS Elastic Beanstalk, not AWS SAM.

NEW QUESTION 164

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

Where should the temporary files be stored?

- A. the `/tmp` directory
- B. Amazon Elastic File System (Amazon EFS)
- C. Amazon Elastic Block Store (Amazon EBS)
- D. Amazon S3

Answer: A

Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers. Lambda provides a local file system that can be used to store temporary files during invocation. The local file system is mounted under the `/tmp` directory and has a limit of 512 MB. The temporary files are accessible only by the Lambda function that created them and are deleted after the function execution ends. The developer can store temporary files that are less than 10 MB in the `/tmp` directory and access and modify them multiple times during invocation.

References:

? [What Is AWS Lambda? - AWS Lambda]

? [AWS Lambda Execution Environment - AWS Lambda]

NEW QUESTION 165

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 170

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

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

- A. Amazon S3
- B. AWS CloudTrail
- C. Amazon CloudWatch
- D. Amazon DynamoDB

Answer: C

Explanation:

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

References: AWS Lambda Monitoring, [CloudWatch Logs Insights]

NEW QUESTION 175

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

How can the developer improve the function's performance?

- A. Increase the function's CPU core count.
- B. Increase the function's memory.
- C. Increase the function's reserved concurrency.
- D. Increase the function's timeout.

Answer: B

Explanation:

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

NEW QUESTION 177

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

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

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

- A. Create four access points that allow access to the central S3 bucket
- B. Assign an access point to each web application bucket.
- C. Create a bucket policy that allows access to the central S3 bucket
- D. Attach the bucket policy to the central S3 bucket.
- E. Create a cross-origin resource sharing (CORS) configuration that allows access to the central S3 bucket
- F. Add the CORS configuration to the central S3 bucket.
- G. Create a Content-MD5 header that provides a message integrity check for the central S3 bucket
- H. Insert the Content-MD5 header for each web application request.

Answer: C

Explanation:

This is a frequent trouble. Web applications cannot access the resources in other domains by default, except some exceptions. You must configure CORS on the

resources to be accessed. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/cors.html>

NEW QUESTION 179

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

- A. Add logging statements to the Lambda function
- B. then use Amazon CloudWatch to view the logs.
- C. Use AWS CloudTrail and then examine the logs.
- D. Use AWS X-Ray
- E. then examine the segments and errors.
- F. Run Amazon Inspector agents and then analyze performance.

Answer: C

Explanation:

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

NEW QUESTION 182

.....

Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

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

100% Actual & Verified — Instant Download, Please Click
[Order The DVA-C02 Practice Test Here](#)