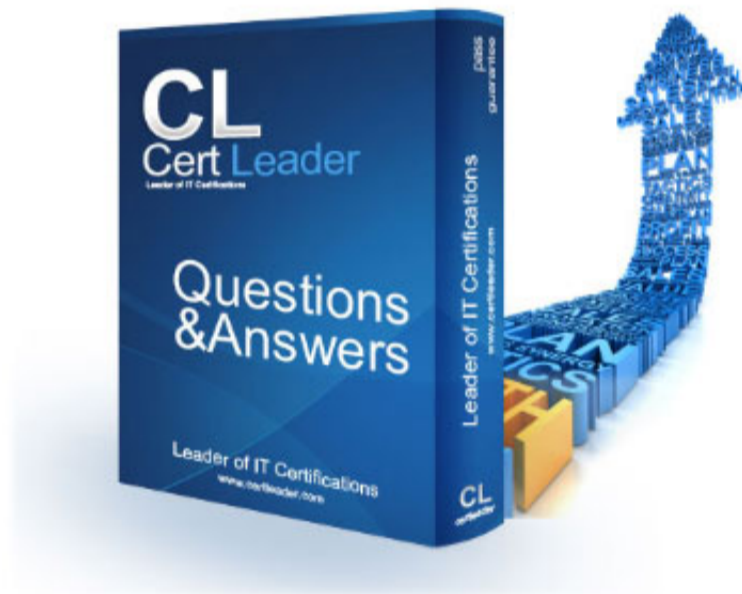


DVA-C02 Dumps

DVA-C02

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



NEW QUESTION 1

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 2

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 3

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

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

Which command will meet these requirements?

- A. `sam deploy -force-upload`
- B. `sam deploy -no-execute-changeset`
- C. `sam package`
- D. `sam sync -watch`

Answer: D

Explanation:

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

Reference: AWS SAM Accelerate

NEW QUESTION 4

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

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

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

Answer: C

Explanation:

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

References:

? [What Is AWS CodeCommit? - AWS CodeCommit]

? [AWS CodePipeline - AWS CodeCommit]

NEW QUESTION 5

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

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

Answer: D

Explanation:

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

References:

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

? [What Is Amazon CloudFront? - Amazon CloudFront]

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

NEW QUESTION 6

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

- A. Update the application to retrieve the variables from AWS Systems Manager Parameter Store
- B. Use unique paths in Parameter Store for each variable in each environment
- C. Store the credentials in AWS Secrets Manager in each environment.
- D. Update the application to retrieve the variables from AWS Key Management Service (AWS KMS). Store the API URL and credentials as unique keys for each environment.
- E. Update the application to retrieve the variables from an encrypted file that is stored with the application
- F. Store the API URL and credentials in unique files for each environment.
- G. Update the application to retrieve the variables from each of the deployed environment
- H. Define the authentication information and API URL in the ECS task definition as unique names during the deployment process.

Answer: A

Explanation:

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

References:

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

? [Parameter Store - AWS Systems Manager]

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

NEW QUESTION 7

A developer maintains an Amazon API Gateway REST API. Customers use the API through a frontend UI and Amazon Cognito authentication. The developer has a new version of the API that contains new endpoints and backward-incompatible interface changes. The developer needs to provide beta access to other developers on the team without affecting customers. Which solution will meet these requirements with the LEAST operational overhead?

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

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The developer can define a development stage on the API Gateway API and instruct the other developers to point the endpoints to the development stage. This way, the developer can

provide beta access to the new version of the API without affecting customers who use the production stage. This solution will meet the requirements with the least operational overhead.

References:

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

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

NEW QUESTION 8

A developer maintains a critical business application that uses Amazon DynamoDB as the primary data store. The DynamoDB table contains millions of documents and receives 30-60 requests each minute. The developer needs to perform processing in near-real time on the documents when they are added or updated in the DynamoDB table.

How can the developer implement this feature with the LEAST amount of change to the existing application code?

- A. Set up a cron job on an Amazon EC2 instance. Run a script every hour to query the table for changes and process the documents.
- B. Enable a DynamoDB stream on the table. Invoke an AWS Lambda function to process the documents.
- C. Update the application to send a PutEvents request to Amazon EventBridge.
- D. Create an EventBridge rule to invoke an AWS Lambda function to process the documents.
- E. Update the application to synchronously process the documents directly after the DynamoDB write.

Answer: B

Explanation:

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

NEW QUESTION 9

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

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

Answer: A

Explanation:

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

? Set up mock integrations for API Gateway REST APIs

? Mock Integration for API Gateway - AWS CloudFormation

? Mocking API Responses with API Gateway

? How to mock API Gateway responses with AWS SAM

NEW QUESTION 10

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

Which action can help the company achieve this goal?

- A. Enable API caching in API Gateway.
- B. Configure API Gateway to use an interface VPC endpoint.
- C. Enable cross-origin resource sharing (CORS) for the APIs.
- D. Configure usage plans and API keys in API Gateway.

Answer: A

Explanation:

Amazon API Gateway is a service that enables developers to create, publish, maintain, monitor, and secure APIs at any scale. The developer can enable API caching in API Gateway to cache responses from the backend integration point for a specified time-to-live (TTL) period. This can improve the responsiveness of the APIs by reducing the number

of calls made to the backend service. References:

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

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

NEW QUESTION 10

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.
- 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 'AWS::Lambda::Function' 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 12

A developer is designing a serverless application for a game in which users register and log in through a web browser. The application makes requests on behalf of users to a set of AWS Lambda functions that run behind an Amazon API Gateway HTTP API.

The developer needs to implement a solution to register and log in users on the application's sign-in page. The solution must minimize operational overhead and must minimize ongoing management of user identities.

Which solution will meet these requirements?

- A. Create Amazon Cognito user pools for external social identity providers. Configure IAM roles for the identity pools.
- B. Program the sign-in page to create users' IAM groups with the IAM roles attached to the groups.
- C. Create an Amazon RDS for SQL Server DB instance to store the users and manage the permissions to the backend resources in AWS.
- D. Configure the sign-in page to register and store the users and their passwords in an Amazon DynamoDB table with an attached IAM policy.

Answer: A

Explanation:

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

NEW QUESTION 14

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

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

The company wants the support team to receive notifications in near real time only when

the payment processing external API error rate exceeds 5% of the total number of transactions in an hour. Developers need to use an existing Amazon Simple Notification Service (Amazon SNS) topic that is configured to notify the support team.

Which solution will meet these requirements?

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

Answer: B

Explanation:

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

References:

? [What Is Amazon CloudWatch? - Amazon CloudWatch]

? [Publishing Custom Metrics - Amazon CloudWatch]

? [Creating Amazon CloudWatch Alarms - Amazon CloudWatch]

NEW QUESTION 15

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 Simple Queue Service (Amazon SQS) queue serves as an event source for an AWS Lambda function. In the SQS queue, each item corresponds to a video file that the Lambda function must convert to a smaller resolution. The Lambda function is timing out on longer video files, but the Lambda function's timeout is already configured to its maximum value.

What should a developer do to avoid the timeouts without additional code changes?

- A. Increase the memory configuration of the Lambda function.
- B. Increase the visibility timeout on the SQS queue.
- C. Increase the instance size of the host that runs the Lambda function.
- D. Use multi-threading for the conversion.

Answer: A

Explanation:

Increasing the memory configuration of the Lambda function will also increase the CPU and network throughput available to the function. This can improve the performance of the video conversion process and reduce the execution time of the function. This solution does not require any code changes or additional resources. It is also recommended to follow the best practices for preventing Lambda function timeouts¹. References

? Troubleshoot Lambda function invocation timeout errors | AWS re:Post

NEW QUESTION 21

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

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

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

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

H. Associate the stream's Amazon Resource Name (ARN) with the Lambda function. Configure the Lambda function to write to the S3 bucket as records appear in the table's stream.

- I. Modify the Lambda function to publish to a new Amazon SNS topic.
- J. Create a new Lambda function that receives orders from the SNS topic.
- 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 24

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

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

- A. Create an AWS CodeCommit project.
- B. Add the repository package's build and test commands to the project's buildspec.
- C. Create an AWS CodeBuild project.

commands to the project's buildspec

- D. Add the repository package's build and test
- E. Create an AWS CodeDeploy protect
- F. Add the repository package's build and test commands to the project's buildspec
- G. Add an action to the source stag
- H. Specify the newly created project as the action provide
- I. Specify the build attract as the actions input artifact.
- J. Add a new stage to the pipeline alter the source stag
- K. Add an action to the new stag
- L. Speedy the newly created protect as the action provide
- M. Specify the source artifact as the action's input artifact.

Answer: BE

Explanation:

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

NEW QUESTION 27

A developer is creating a simple proof-of-concept demo by using AWS CloudFormation and AWS Lambda functions The demo will use a CloudFormation template to deploy an existing Lambda function The Lambda function uses deployment packages and dependencies stored in Amazon S3 The developer defined anAWS Lambda Function resource in a CloudFormation template. The developer needs to add the S3 bucket to the CloudFormation template. What should the developer do to meet these requirements with the LEAST development effort?

- A. Add the function code in the CloudFormation template inline as the code property
- B. Add the function code in the CloudFormation template as the ZipFile property.
- C. Find the S3 key for the Lambda function Add the S3 key as the ZipFile property in the CloudFormation template.
- D. Add the relevant key and bucket to the S3Bucket and S3Key properties in the CloudFormation template

Answer: D

Explanation:

The easiest way to add the S3 bucket to the CloudFormation template is to use the S3Bucket and S3Key properties of the AWS::Lambda::Function resource. These properties specify the name of the S3 bucket and the location of the .zip file that contains the function code and dependencies. This way, the developer does not need to modify the function code or upload it to a different location. The other options are either not feasible or not efficient. The code property can only be used for inline code, not for code stored in S3. The ZipFile property can only be used for code that is less than 4096 bytes, not for code that has dependencies. Finding the S3 key for the Lambda function and adding it as the ZipFile property would not work, as the ZipFile property expects a base64-encoded .zip file, not an S3 location. References
? AWS::Lambda::Function - AWS CloudFormation
? Deploying Lambda functions as .zip file archives
? AWS Lambda Function Code - AWS CloudFormation

NEW QUESTION 31

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 35

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

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Amazon Machine Images (AMIs) are encrypted snapshots of EC2 instances that can be used to launch new instances. The developer can create new AMIs from the existing instances and specify encryption parameters. The developer can copy the encrypted AMIs to the destination Region and use them to create a new application stack. The developer can delete the unencrypted AMIs after the encryption process is complete. This solution will meet the encryption requirement and allow the developer to expand the application to run in the destination Region. References:

? [Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud]
? [Encrypting an Amazon EBS Snapshot - Amazon Elastic Compute Cloud]
? [Copying an AMI - Amazon Elastic Compute Cloud]

NEW QUESTION 36

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 straight-forward 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 41

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 46

A company has an application that runs as a series of AWS Lambda functions. Each Lambda function receives data from an Amazon Simple Notification Service (Amazon SNS) topic and writes the data to an Amazon Aurora DB instance. To comply with an information security policy, the company must ensure that the Lambda functions all use a single securely encrypted database connection string to access Aurora. Which solution will meet these requirements'?

- A. Use IAM database authentication for Aurora to enable secure database connections for all the Lambda functions.
- B. Store the credentials and read the credentials from an encrypted Amazon RDS DB instance.
- C. Store the credentials in AWS Systems Manager Parameter Store as a secure string parameter.
- D. Use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption.

Answer: A

Explanation:

This solution will meet the requirements by using IAM database authentication for Aurora, which enables using IAM roles or users to authenticate with Aurora databases instead of using passwords or other secrets. The developer can use IAM database authentication for Aurora to enable secure database connections for all the Lambda functions that access Aurora DB instance. The developer can create an IAM role with permission to connect to Aurora DB instance and attach it to each Lambda function. The developer can also configure Aurora DB instance to use IAM database authentication and enable encryption in transit using SSL certificates. This way, the Lambda functions can use a single securely encrypted database connection string to access Aurora without needing any secrets or passwords. Option B is not optimal because it will store the credentials and read them from an encrypted Amazon RDS DB instance, which may introduce additional costs and complexity for managing and accessing another RDS DB instance. Option C is not optimal because it will store the credentials in AWS Systems Manager Parameter Store as a secure string parameter, which may require additional steps or permissions to retrieve and decrypt the credentials from Parameter Store. Option D is not optimal because it will use Lambda environment variables with a shared AWS Key Management Service (AWS KMS) key for encryption, which may not be secure or scalable as environment variables are stored as plain text unless encrypted with AWS KMS. References: [IAM Database Authentication for MySQL and PostgreSQL], [Using SSL/TLS to Encrypt a Connection to a DB Instance]

NEW QUESTION 51

A company has an existing application that has hardcoded database credentials A developer needs to modify the existing application The application is deployed in two AWS Regions with an active-passive failover configuration to meet company's disaster recovery strategy The developer needs a solution to store the credentials outside the code. The solution must comply With the company's disaster recovery strategy Which solution Will meet these requirements in the MOST secure way?

- A. Store the credentials in AWS Secrets Manager in the primary Regio

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

Answer: A

Explanation:

AWS Secrets Manager is a service that allows you to store and manage secrets, such as database credentials, API keys, and passwords, in a secure and centralized way. It also provides features such as automatic secret rotation, auditing, and monitoring¹. By using AWS Secrets Manager, you can avoid hardcoding credentials in your code, which is a bad security practice and makes it difficult to update them. You can also replicate your secrets to another Region, which is useful for disaster recovery purposes². To access your secrets from your application, you can use the ARN of the secret, which is a unique identifier that includes the Region name. This way, your application can use the appropriate secret based on the Region where it is deployed³.

References:

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

NEW QUESTION 54

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

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

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

- A. Set up an S3 event notification that invokes the `removePii` function when an S3 GET request is made
- B. Call Amazon S3 by using a GET request to access the object without PII.
- C. Set up an S3 event notification that invokes the `removePii` function when an S3 PUT request is made
- D. Call Amazon S3 by using a PUT request to access the object without PII.
- E. Create an S3 Object Lambda access point from the S3 console
- F. Select the `removePii` function
- G. Use S3 Access Points to access the object without PII.
- H. Create an S3 access point from the S3 console
- I. Use the access point name to call the `GetObjectLegalHold` S3 API function
- J. Pass in the `removePii` function name to access the object without PII.

Answer: C

Explanation:

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

document in S3 and apply different transformations depending on who accesses it. Reference: Using AWS Lambda with Amazon S3

NEW QUESTION 59

A developer is testing an application that invokes an AWS Lambda function asynchronously. During the testing phase the Lambda function fails to process after two retries.

How can the developer troubleshoot the failure?

- A. Configure AWS CloudTrail logging to investigate the invocation failures.
- B. Configure Dead Letter Queues by sending events to Amazon SQS for investigation.
- C. Configure Amazon Simple Workflow Service to process any direct unprocessed events.
- D. Configure AWS Config to process any direct unprocessed events.

Answer: B

Explanation:

This solution allows the developer to troubleshoot the failure by capturing unprocessed events in a queue for further analysis. Dead Letter Queues (DLQs) are queues that store messages that could not be processed by a service, such as Lambda, for various reasons, such as configuration errors, throttling limits, or permissions issues. The developer can configure DLQs for Lambda functions by sending events to either an Amazon Simple Queue Service (SQS) queue or an Amazon Simple Notification Service (SNS) topic. The developer can then inspect the messages in the queue or topic to identify and fix the root cause of the failure. Configuring AWS CloudTrail logging will not capture invocation failures for asynchronous Lambda invocations, but only record API calls made by or on behalf of Lambda. Configuring Amazon Simple Workflow Service (SWF) or AWS Config will not process any direct unprocessed events, but require additional integration and configuration.

Reference: [Using AWS Lambda with DLQs], [Asynchronous invocation]

NEW QUESTION 62

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 67

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

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

- ☐ A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS).
- ☒ B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2.
- ☐ C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.
- ☐ D. Use an AWS Batch job that is submitted to an AWS Batch job queue.

Answer: C

Explanation:

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

* C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event. This is correct. AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging¹. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources². EventBridge can create rules that run on a schedule, either at regular intervals or at specific times and dates, and invoke targets such as Lambda functions³. This solution meets the requirements of creating a small application that makes the same API call once each day at a designated time, without requiring any infrastructure in the AWS Cloud or any operational overhead.

* A. Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS). This is incorrect. Amazon EKS is a fully managed Kubernetes service that allows you to run containerized applications on AWS⁴. Kubernetes cron jobs are tasks that run periodically on a given schedule⁵. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EKS cluster, which would incur additional costs and complexity.

* B. Use an Amazon Linux crontab scheduled job that runs on Amazon EC2. This is incorrect. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud⁶. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date⁷. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EC2 instance, which would incur additional costs and complexity.

* D. Use an AWS Batch job that is submitted to an AWS Batch job queue. This is incorrect. AWS Batch enables you to run batch computing workloads on the AWS Cloud⁸. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially on compute environments⁹. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to configure and manage an AWS Batch environment, which would incur additional costs and complexity.

References:

- ? 1: What is AWS Lambda? - AWS Lambda
- ? 2: What is Amazon EventBridge? - Amazon EventBridge
- ? 3: Creating an Amazon EventBridge rule that runs on a schedule - Amazon EventBridge
- ? 4: What is Amazon EKS? - Amazon EKS
- ? 5: CronJob - Kubernetes
- ? 6: What is Amazon EC2? - Amazon EC2
- ? 7: Crontab in Linux with 20 Useful Examples to Schedule Jobs - Tecmint
- ? 8: What is AWS Batch? - AWS Batch
- ? 9: Jobs - AWS Batch

NEW QUESTION 72

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

Which solution will meet this requirement?

- A. Create an AWS Key Management Service (AWS KMS) key
- B. Assign the KMS key to the S3 bucket.
- C. Set the x-amz-server-side-encryption header when invoking the PutObject API operation.
- D. Provide the encryption key in the HTTP header of every request.
- E. Apply TLS to encrypt the traffic to the S3 bucket.

Answer: B

Explanation:

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

Protecting data using server-side encryption with Amazon S3-managed encryption keys (SSE-S3)

NEW QUESTION 73

A developer is working on a serverless application that needs to process any changes to an Amazon DynamoDB table with an AWS Lambda function. How should the developer configure the Lambda function to detect changes to the DynamoDB table?

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

Answer: C

Explanation:

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

References:

? [Amazon DynamoDB]

? [DynamoDB Streams - Amazon DynamoDB]

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

NEW QUESTION 78

A company built an online event platform. For each event, the company organizes quizzes and generates leaderboards that are based on the quiz scores. The company stores the leaderboard data in Amazon DynamoDB and retains the data for 30 days after an event is complete. The company then uses a scheduled job to delete the old leaderboard data.

The DynamoDB table is configured with a fixed write capacity. During the months when many events occur, the DynamoDB write API requests are throttled when the scheduled delete job runs.

A developer must create a long-term solution that deletes the old leaderboard data and optimizes write throughput.

Which solution meets these requirements?

- A. Configure a TTL attribute for the leaderboard data.
- B. Use DynamoDB Streams to schedule and delete the leaderboard data.
- C. Use AWS Step Functions to schedule and delete the leaderboard data.
- D. Set a higher write capacity when the scheduled delete job runs.

Answer: A

Explanation:

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

NEW QUESTION 80

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 key. The data key is a random byte string that can be used with any standard encryption algorithm, such as AES or SM42. The plaintext data key can be used to encrypt or decrypt data outside of AWS KMS, while the encrypted data key can be stored with the encrypted data and later decrypted by AWS KMS.

? In this scenario, the developer needs to use the GenerateDataKey API to encrypt

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

NEW QUESTION 85

A developer is deploying an AWS Lambda function. The developer wants the ability to return to older versions of the function quickly and seamlessly. How can the developer achieve this goal with the LEAST operational overhead?

- A. Use AWS OpsWorks to perform blue/green deployments.
- B. Use a function alias with different versions.
- C. Maintain deployment packages for older versions in Amazon S3.

D. Use AWS CodePipeline for deployments and rollbacks.

Answer: B

Explanation:

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

NEW QUESTION 86

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

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

- A. Define a maintenance window for the Lambda functions to ensure that the functions get updated copies of the dependencies.
- B. Upgrade the Lambda functions to the most recent runtime version.
- C. Define a Lambda layer that contains all of the shared dependencies.
- D. Use an AWS CodeCommit repository to host the dependencies in a centralized location.

Answer: C

Explanation:

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

Reference: [AWS Lambda layers]

NEW QUESTION 89

A company's developer has deployed an application in AWS by using AWS CloudFormation. The CloudFormation stack includes parameters in AWS Systems Manager Parameter Store that the application uses as configuration settings. The application can modify the parameter values.

When the developer updated the stack to create additional resources with tags, the developer noted that the parameter values were reset and that the values ignored the latest changes made by the application. The developer needs to change the way the company deploys the CloudFormation stack. The developer also needs to avoid resetting the parameter values outside the stack.

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

- A. Modify the CloudFormation stack to set the deletion policy to Retain for the Parameter Store parameters.
- B. Create an Amazon DynamoDB table as a resource in the CloudFormation stack to hold configuration data for the application. Migrate the parameters that the application is modifying from Parameter Store to the DynamoDB table.
- C. Create an Amazon RDS DB instance as a resource in the CloudFormation stack.
- D. Create a table in the database for parameter configuration.
- E. Migrate the parameters that the application is modifying from Parameter Store to the configuration table.
- F. Modify the CloudFormation stack policy to deny updates on Parameter Store parameters.

Answer: D

Explanation:

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

NEW QUESTION 92

A developer created an AWS Lambda function that performs a series of operations that involve multiple AWS services. The function's duration time is higher than normal. To determine the cause of the issue, the developer must investigate traffic between the services without changing the function code.

Which solution will meet these requirements?

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

AWS X-Ray is a service that helps you analyze and debug your applications. You can use X-Ray to trace requests made to your Lambda function and other AWS services, and identify performance bottlenecks and errors. Enabling active tracing in your Lambda function allows X-Ray to collect data from the function invocation and the downstream services that it calls. You can then review the logs and service maps in X-Ray to diagnose the issue. References

? Monitoring and troubleshooting Lambda functions - AWS Lambda

? Using AWS Lambda with AWS X-Ray

? Troubleshoot Lambda function cold start issues | AWS re:Post

NEW QUESTION 97

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 100

A company's website runs on an Amazon EC2 instance and uses Auto Scaling to scale the environment during peak times. Website users across the world are experiencing high latency due to static content on the EC2 instance, even during non-peak hours.

When completion of steps will resolve the latency issue? (Select TWO)

- A. Double the Auto Scaling group's maximum number of servers
- B. Host the application code on AWS Lambda
- C. Scale vertically by resizing the EC2 instances
- D. Create an Amazon CloudFront distribution to cache the static content
- E. Store the application's static content in Amazon S3

Answer: DE

Explanation:

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

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

NEW QUESTION 104

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 109

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

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

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

- A. Use an AWS Step Functions State machine to monitor API failure
- B. Use the Wait state to delay calling the Lambda function.
- C. Use an Amazon Simple Queue Service (Amazon SQS) queue to hold the API call
- D. Configure the Lambda function to poll the queue within the API threshold limits.
- E. Use an Amazon CloudWatch Logs metric to count the number of API call

- F. Configure an Amazon CloudWatch alarm that stops the currently running instance of the Lambda function when the metric exceeds the API threshold limits.
- G. Use Amazon Kinesis Data Firehose to batch the API calls and deliver them to an Amazon S3 bucket with an event notification to invoke the Lambda function.

Answer: A

Explanation:

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

Reference: AWS Step Functions Wait state

NEW QUESTION 110

A developer is planning to migrate on-premises company data to Amazon S3. The data must be encrypted, and the encryption Keys must support automatic annual rotation. The company must use AWS Key Management Service (AWS KMS) to encrypt the data. When type of keys should the developer use to meet these requirements?

- A. Amazon S3 managed keys
- B. Symmetric customer managed keys with key material that is generated by AWS
- C. Asymmetric customer managed keys with key material that generated by AWS
- D. Symmetric customer managed keys with imported key material

Answer: B

Explanation:

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

Reference: Using AWS KMS keys to encrypt S3 objects

NEW QUESTION 113

A company has built an AWS Lambda function to convert large image files into output files that can be used in a third-party viewer application. The company recently added a new module to the function to improve the output of the generated files. However, the new module has increased the bundle size and has increased the time that is needed to deploy changes to the function code. How can a developer increase the speed of the Lambda function deployment?

- A. Use AWS CodeDeploy to deploy the function code
- B. Use Lambda layers to package and load dependencies.
- C. Increase the memory size of the function.
- D. Use Amazon S3 to host the function dependencies

Answer: B

Explanation:

Using Lambda layers is a way to reduce the size of the deployment package and speed up the deployment process. Lambda layers are reusable components that can contain libraries, custom runtimes, or other dependencies. By using layers, the developer can separate the core function logic from the dependencies, and avoid uploading them every time the function code changes. Layers can also be shared across multiple functions or accounts, which can improve consistency and maintainability. References

? Working with AWS Lambda layers

? AWS Lambda Layers Best Practices

? Best practices for working with AWS Lambda functions

NEW QUESTION 115

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

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

- A. Package each Python library in its own .zip file archiv
- B. Deploy each Lambda function with its own copy of the library.
- C. Create a Lambda layer with the required Python librar
- D. Use the Lambda layer in both Lambda functions.
- E. Combine the two Lambda functions into one Lambda functio
- F. Deploy the Lambda function as a single .zip file archive.
- G. Download the Python library to an S3 bucke
- H. Program the Lambda functions to reference the object URLs.

Answer: B

Explanation:

AWS Lambda is a service that lets developers run code without provisioning or managing servers. Lambda layers are a distribution mechanism for libraries, custom runtimes, and other dependencies. The developer can create a Lambda layer with the

required Python library and use the layer in both Lambda functions. This will reduce the size of the Lambda deployment packages and avoid reaching the quota for the maximum size of zipped deployment packages. The developer can also benefit from using layers to manage dependencies separately from function code.

References:

? [What Is AWS Lambda? - AWS Lambda]

? [AWS Lambda Layers - AWS Lambda]

NEW QUESTION 123

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

A.

- Configure the CloudFront cach
- B. Update the application to return cached content based upon the default request headers.
- C. Override the cache method in the selected stage of API Gatewa
- D. Select the POST method.
- E. Save the latest request response in Lambda /tmp director
- F. Update the Lambda function to check the /tmp directory.
- G. Save the latest request in AWS Systems Manager Parameter Stor
- H. Modify the Lambda function to take the latest request response from Parameter Store.

Answer: B

Explanation:

Amazon API Gateway provides tools for creating and documenting web APIs that route HTTP requests to Lambda functions². You can secure access to your API with authentication and authorization controls. Your APIs can serve traffic over the internet or can be accessible only within your VPC². You can override the cache method in the selected stage of API Gateway². Therefore, option B is correct.

NEW QUESTION 124

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 128

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 133

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

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

Which solution will meet these requirements?

- A. Add an AWS Secrets Manager GenerateSecretString resource to the CloudFormation template.
- B. Set the value to reference new credentials to the CloudFormation resource.
- C. Use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resource to store the credentials as a parameter.
- D. Set the parameter value to reference the new credential.
- E. Set the parameter type to SecureString.
- F. Add an AWS Systems Manager Parameter Store resource to the CloudFormation template.
- G. Set the CloudFormation resource value to reference the new credentials. Set the resource NoEcho attribute to true.
- H. Use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resources to store the credentials as a parameter.
- I. Set the parameter value to reference the new credential.
- J. Set the parameter NoEcho attribute to true.

Answer: B

Explanation:

The solution that will meet the requirements is to use the AWS SDK ssm PutParameter operation in the Lambda function from the existing custom resource to store the credentials as a parameter. Set the parameter value to reference the new credentials. Set the parameter type to SecureString. This way, the developer can store the API credentials with minimal operational overhead, as AWS Systems Manager Parameter Store provides secure and scalable storage for configuration data. The SecureString parameter type encrypts the parameter value with AWS Key Management Service (AWS KMS). The other options either involve adding additional resources to the CloudFormation template, which increases complexity and cost, or do not encrypt the parameter value, which reduces security.

Reference: [Creating Systems Manager parameters](#)

NEW QUESTION 136

A company is planning to use AWS CodeDeploy to deploy an application to Amazon Elastic Container Service (Amazon ECS). During the deployment of a new version of the application, the company initially must expose only 10% of live traffic to the new version of the deployed application. Then, after 15 minutes elapse, the company must route all the remaining live traffic to the new version of the deployed application.

Which CodeDeploy predefined configuration will meet these requirements?

- A. CodeDeployDefault::ECSCanary10Percent15Minutes
- B. CodeDeployDefault::LambdaCanary10Percent5Minutes
- C. CodeDeployDefault::LambdaCanary10Percent15Minutes

D. CodeDeployDefault ECSECSLinear10PercentEvery1 Minutes

Answer: A

Explanation:

The predefined configuration "CodeDeployDefault.ECSCanary10Percent15Minutes" is designed for Amazon Elastic Container Service (Amazon ECS) deployments and meets the specified requirements. It will perform a canary deployment, which means it will initially route 10% of live traffic to the new version of the application, and then after 15 minutes elapse, it will automatically route all the remaining live traffic to the new version. This gradual deployment approach allows

the company to verify the health and performance of the new version with a small portion of traffic before fully deploying it to all users.

NEW QUESTION 139

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 142

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 145

A developer is investigating an issue in part of a company's application. In the application messages are sent to an Amazon Simple Queue Service (Amazon SQS) queue. The AWS Lambda function polls messages from the SQS queue and sends email messages by using Amazon Simple Email Service (Amazon SES). Users have been receiving duplicate email messages during periods of high traffic.

Which reasons could explain the duplicate email messages? (Select TWO.)

- A. Standard SQS queues support at-least-once message delivery
- B. Standard SQS queues support exactly-once processing, so the duplicate email messages are because of user error.
- C. Amazon SES has the DomainKeys Identified Mail (DKIM) authentication incorrectly configured
- D. The SQS queue's visibility timeout is lower than or the same as the Lambda function's timeout.
- E. The Amazon SES bounce rate metric is too high.

Answer: AD

Explanation:

Standard SQS queues support at-least-once message delivery, which means that a message can be delivered more than once to the same or different consumers. This can happen if the message is not deleted from the queue before the visibility timeout expires, or if there is a network issue or a system failure. The SQS queue's visibility timeout is the period of time that a message is invisible to other consumers after it is received by one consumer. If the visibility timeout is lower than or the same as the Lambda function's timeout, the Lambda function might not be able to process and delete the message before it becomes visible again, leading to duplicate processing and email messages. To avoid this, the visibility timeout should be set to at least 6 times the length of the Lambda function's timeout. The other options are not related to the issue of duplicate email messages. References

? Using the Amazon SQS message deduplication ID

? Exactly-once processing - Amazon Simple Queue Service

? Amazon SQS duplicated messages in queue - Stack Overflow

? amazon web services - How long can duplicate SQS messages persist ...

? Standard SQS - Duplicate message | AWS re:Post - Amazon Web Services, Inc.

NEW QUESTION 149

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 DynamoDB table.
- 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 DB 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 DB instance, create a table where the primary key is the title and all other data is encoded into JSON format as one additional column.

Answer: A

Explanation:

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

References:

? [Amazon DynamoDB]

? [Working with Queries - Amazon DynamoDB]

? [Working with Global Secondary Indexes - Amazon DynamoDB]

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.
- 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 158

A company has an Amazon S3 bucket that contains sensitive data. The data must be encrypted in transit and at rest. The company

encrypts the data in the S3 bucket by using an AWS Key Management Service (AWS KMS) key. A developer needs to grant several other AWS accounts the permission to use the S3 GetObject operation to retrieve the data from the S3 bucket. How can the developer enforce that all requests to retrieve the data provide encryption in transit?

- A. Define a resource-based policy on the S3 bucket to deny access when a request meets the condition “aws:SecureTransport”: “false”.
- B. Define a resource-based policy on the S3 bucket to allow access when a request meets the condition “aws:SecureTransport”: “false”.
- C. Define a role-based policy on the other accounts' roles to deny access when a request meets the condition of “aws:SecureTransport”: “false”.
- D. Define a resource-based policy on the KMS key to deny access when a request meets the condition of “aws:SecureTransport”: “false”.

Answer: A

Explanation:

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

NEW QUESTION 163

A developer wants to insert a record into an Amazon DynamoDB table as soon as a new file is added to an Amazon S3 bucket. Which set of steps would be necessary to achieve this?

- A. Create an event with Amazon EventBridge that will monitor the S3 bucket and then insert the records into DynamoDB.
- B. Configure an S3 event to invoke an AWS Lambda function that inserts records into DynamoDB.
- C. Create an AWS Lambda function that will poll the S3 bucket and then insert the records into DynamoDB.
- D. Create a cron job that will run at a scheduled time and insert the records into DynamoDB.

Answer: B

Explanation:

Amazon S3 is a service that provides highly scalable, durable, and secure object storage. Amazon DynamoDB is a fully managed NoSQL database service that

provides fast and consistent performance with seamless scalability. AWS Lambda is a service that lets developers run code without provisioning or managing servers. The developer can configure an S3 event to invoke a Lambda function that inserts records into DynamoDB whenever a new file is added to the S3 bucket. This solution will meet the requirement of inserting a record into DynamoDB as soon as a new file is added to S3. References:

- ? [Amazon Simple Storage Service (S3)]
- ? [Amazon DynamoDB]
- ? [What Is AWS Lambda? - AWS Lambda]
- ? [Using AWS Lambda with Amazon S3 - AWS Lambda]

NEW QUESTION 165

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 169

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 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 171

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

Which solution will meet these requirements MOST efficiently?

- A. Create a Lambda environment variable to store the table name
- B. Use the standard method for the programming language to retrieve the variable.
- C. Store the table name in a file
- D. Store the file in the `/tmp` folder
- E. Use the SDK for the programming language to retrieve the table name.
- F. Create a file to store the table name
- G. Zip the file and upload the file to the Lambda layer
- H. Use the SDK for the programming language to retrieve the table name.
- I. Create a global variable that is outside the handler in the Lambda function to store the table name.

Answer: A

Explanation:

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

Reference: Using AWS Lambda environment variables

NEW QUESTION 174

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

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

- A. Set the image resize Lambda function as a destination of the avatar generator Lambda function for the events that fail processing.
- B. Create an Amazon Simple Queue Service (Amazon SQS) queue
- C. Set the SQS queue as a destination with an on failure condition for the avatar generator Lambda function

- D. Configure the image resize Lambda function to poll from the SQS queue.
- E. Create an AWS Step Functions state machine that invokes the avatar generator Lambda function and uses the image resize Lambda function as a fallback.
- F. Create an Amazon EventBridge rule that matches events from the S3 bucket to invoke the state machine.
- G. Create an Amazon Simple Notification Service (Amazon SNS) topic.
- H. Set the SNS topic as a destination with an on failure condition for the avatar generator Lambda function.
- I. Subscribe the image resize Lambda function to the SNS topic.

Answer: A

Explanation:

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

Reference: Using AWS Lambda destinations

NEW QUESTION 177

An AWS Lambda function requires read access to an Amazon S3 bucket and requires read/write access to an Amazon DynamoDB table. The correct IAM policy already exists.

What is the MOST secure way to grant the Lambda function access to the S3 bucket and the DynamoDB table?

- A. Attach the existing IAM policy to the Lambda function.
- B. Create an IAM role for the Lambda function. Attach the existing IAM policy to the role. Attach the role to the Lambda function.
- C. Create an IAM user with programmatic access. Attach the existing IAM policy to the user.
- D. Add the user access key ID and secret access key as environment variables in the Lambda function.
- E. Add the AWS account root user access key ID and secret access key as encrypted environment variables in the Lambda function.

Answer: B

Explanation:

The most secure way to grant the Lambda function access to the S3 bucket and the DynamoDB table is to create an IAM role for the Lambda function and attach the existing IAM policy to the role. This way, you can use the principle of least privilege and avoid exposing any credentials in your function code or environment variables. You can also leverage the temporary security credentials that AWS provides to the Lambda function when it assumes the role. This solution follows the best practices for working with AWS Lambda functions¹ and designing and architecting with DynamoDB². References

? Best practices for working with AWS Lambda functions

? Best practices for designing and architecting with DynamoDB

NEW QUESTION 179

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

Which solution will meet these requirements MOST cost-effectively?

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

Answer: C

Explanation:

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

NEW QUESTION 181

An application runs on multiple EC2 instances behind an ELB.

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

- A. Write data to Amazon ElastiCache
- B. Write data to Amazon Elastic Block Store
- C. Write data to Amazon EC2 instance Store
- D. Write data to the root filesystem

Answer: A

Explanation:

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

Reference: Using ElastiCache for session management

NEW QUESTION 186

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 189

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

```
{
  "FailedRecordCount": 1,
  "Records": [
    {
      "SequenceNumber": "21269319989900637946712965403778482371",
      "ShardId": "shardId-000000000001"
    },
    {
      "ErrorCode": "ProvisionedThroughputExceededException",
      "ErrorMessage": "Rate exceeded for shard shardId-000000000001 in
                        stream exampleStreamName under account 123456789."
    },
    {
      "SequenceNumber": "21269319989999637946712965403778482985",
      "ShardId": "shardId-000000000002"
    }
  ]
}
```

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

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

Answer: AC

Explanation:

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

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

References:

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

NEW QUESTION 190

An developer is building a serverless application by using the AWS Serverless Application Model (AWS SAM). The developer is currently testing the application in a development environment. When the application is nearly finished, the developer will need to set up additional testing and staging environments for a quality assurance team.

The developer wants to use a feature of the AWS SAM to set up deployments to multiple environments.

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

- A. Add a configuration file in TOML format to group configuration entries to every environment
- B. Add a table for each testing and staging environment
- C. Deploy updates to the environments by using the sam deploy command and the --config-env flag that corresponds to the each environment.
- D. Create additional AWS SAM templates for each testing and staging environment
- E. Write a custom shell script that uses the sam deploy command and the --template-file flag to deploy updates to the environments.
- F. Create one AWS SAM configuration file that has default parameter

- G. Perform updates to the testing and staging environments by using the `--parameter-overrides` flag in the AWS SAM CLI and the parameters that the updates will override.
- H. Use the existing AWS SAM template.
- I. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment.
- J. Deploy updates to the testing and staging environments by using the `sam deploy` command.

Answer: A

Explanation:

The correct answer is A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the `sam deploy` command and the `--config-env` flag that corresponds to the each environment.

* A. Add a configuration file in TOML format to group configuration entries to every environment. Add a table for each testing and staging environment. Deploy updates to the environments by using the `sam deploy` command and the `--config-env` flag that corresponds to the each environment. This is correct. This solution will meet the requirements with the least development effort, because it uses a feature of the AWS SAM CLI that supports a project-level configuration file that can be used to configure AWS SAM CLI command parameter values¹. The configuration file can have multiple environments, each with its own set of parameter values, such as stack name, region, capabilities, and more². The developer can use the `--config-env` option to specify which environment to use when deploying the application³. This way, the developer can avoid creating multiple templates or scripts, or manually overriding parameters for each environment.

* B. Create additional AWS SAM templates for each testing and staging environment. Write a custom shell script that uses the `sam deploy` command and the `--template-file` flag to

deploy updates to the environments. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires creating and maintaining multiple templates and scripts for each environment. This can introduce duplication, inconsistency, and complexity in the deployment process.

* C. Create one AWS SAM configuration file that has default parameters. Perform updates to the testing and staging environments by using the `--parameter-overrides` flag in the AWS SAM CLI and the parameters that the updates will override. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires manually specifying and overriding parameters for each environment every time the developer deploys the application. This can be error-prone, tedious, and inefficient.

* D. Use the existing AWS SAM template. Add additional parameters to configure specific attributes for the serverless function and database table resources that are in each environment. Deploy updates to the testing and staging environments by using the `sam deploy` command. This is incorrect. This solution will not meet the requirements with the least development effort, because it requires modifying the existing template and adding complexity to the resource definitions for each environment. This can also make it difficult to manage and track changes across different environments.

References:

? 1: AWS SAM CLI configuration file - AWS Serverless Application Model

? 2: Configuration file basics - AWS Serverless Application Model

? 3: Specify a configuration file - AWS Serverless Application Model

NEW QUESTION 191

A company runs an application on AWS. The application stores data in an Amazon DynamoDB table. Some queries are taking a long time to run. These slow queries involve an attribute that is not the table's partition key or sort key.

The amount of data that the application stores in the DynamoDB table is expected to increase significantly. A developer must increase the performance of the queries.

Which solution will meet these requirements?

- A. Increase the page size for each request by setting the `Limit` parameter to be higher than the default value. Configure the application to retry any request that exceeds the provisioned throughput.
- B. Create a global secondary index (GSI). Set query attribute to be the partition key of the index.
- C. Perform a parallel scan operation by issuing individual scan requests in the parameters specify the segment for the scan requests and the total number of segments for the parallel scan.
- D. Turn on read capacity auto scaling for the DynamoDB table.
- E. Increase the maximum read capacity units (RCUs).

Answer: B

Explanation:

Creating a global secondary index (GSI) is the best solution to improve the performance of the queries that involve an attribute that is not the table's partition key or sort key. A GSI allows you to define an alternate key for your table and query the data using that key. This way, you can avoid scanning the entire table and reduce the latency and cost of your queries. You should also follow the best practices for designing and using GSIs in DynamoDB¹². References

? Working with Global Secondary Indexes - Amazon DynamoDB

? DynamoDB Performance & Latency - Everything You Need To Know

NEW QUESTION 195

A company built a new application in the AWS Cloud. The company automated the bootstrapping of new resources with an Auto Scaling group by using AWS CloudFormation templates. The bootstrap scripts contain sensitive data.

The company needs a solution that is integrated with CloudFormation to manage the sensitive data in the bootstrap scripts.

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

- A. Put the sensitive data into a CloudFormation parameter.
- B. Encrypt the CloudFormation templates by using an AWS Key Management Service (AWS KMS) key.
- C. Put the sensitive data into an Amazon S3 bucket. Update the CloudFormation templates to download the object from Amazon S3 during bootstrap.
- D. Put the sensitive data into AWS Systems Manager Parameter Store as a secure string parameter.
- E. Update the CloudFormation templates to use dynamic references to specify template values.
- F. Put the sensitive data into Amazon Elastic File System (Amazon EFS). Enforce EFS encryption after file system creation.
- G. Update the CloudFormation templates to retrieve data from Amazon EFS.

Answer: C

Explanation:

This solution meets the requirements in the most secure way because it uses a service that is integrated with CloudFormation to manage sensitive data in encrypted form. AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management. You can store sensitive data as secure string parameters, which are encrypted using an AWS Key Management Service (AWS KMS) key of your choice. You can also use dynamic references in your CloudFormation templates to specify template values that are stored in Parameter Store or Secrets Manager without having to include them in your templates. Dynamic references are resolved only during stack creation or update operations, which reduces exposure risks for sensitive data.

Putting sensitive data into a CloudFormation parameter will not encrypt them or protect them from unauthorized access. Putting sensitive data into an Amazon S3 bucket or Amazon Elastic File System (Amazon EFS) will require additional configuration and integration with CloudFormation and may not provide fine-grained access control or encryption for sensitive data.

Reference: [What Is AWS Systems Manager Parameter Store?], [Using Dynamic References to Specify Template Values]

NEW QUESTION 197

.....

Thank You for Trying Our Product

* 100% Pass or Money Back

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

* One year free update

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

* Trusted by Millions

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

* Shop Securely

All transactions are protected by VeriSign!

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

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