



Amazon

Exam Questions AWS-Certified-Developer-Associate

Amazon AWS Certified Developer - Associate

NEW QUESTION 1

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

Which solution will meet this requirement MOST cost-effectively?

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

Answer: C

Explanation:

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

References:

? [Amazon Simple Storage Service (S3)]

? [Using AWS SDKs with Amazon S3]

NEW QUESTION 2

A company needs to deploy all its cloud resources by using AWS CloudFormation templates A developer must create an Amazon Simple Notification Service (Amazon SNS) automatic notification to help enforce this rule. The developer creates an SNS topic and subscribes the email address of the company's security team to the SNS topic.

The security team must receive a notification immediately if an IAM role is created without the use of CloudFormation.

Which solution will meet this requirement?

- A. Create an AWS Lambda function to filter events from CloudTrail if a role was created without CloudFormation Configure the Lambda function to publish to the SNS topi
- B. Create an Amazon EventBridge schedule to invoke the Lambda function every 15 minutes
- C. Create an AWS Fargate task in Amazon Elastic Container Service (Amazon ECS) to filter events from CloudTrail if a role was created without CloudFormation Configure the Fargate task to publish to the SNS topic Create an Amazon EventBridge schedule to run the Fargate task every 15 minutes
- D. Launch an Amazon EC2 instance that includes a script to filter events from CloudTrail if a role was created without CloudFormatio
- E. Configure the script to publish to the SNS topi
- F. Create a cron job to run the script on the EC2 instance every 15 minutes.
- G. Create an Amazon EventBridge rule to filter events from CloudTrail if a role was created without CloudFormation Specify the SNS topic as the target of the EventBridge rule.

Answer: D

Explanation:

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

? Using Amazon EventBridge rules to process AWS CloudTrail events

? Using AWS CloudFormation to create and manage AWS Batch resources

? How to use AWS CloudFormation to configure auto scaling for Amazon Cognito and AWS AppSync

? Using AWS CloudFormation to automate the creation of AWS WAF web ACLs, rules, and conditions

NEW QUESTION 3

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

What is the MOST secure way to resolve this issue?

- A. Update the IAM instance profile that is attached to the EC2 instance to include the S3:* permission for the S3 bucket.
- B. Update the IAM instance profile that is attached to the EC2 instance to include the S3:ListBucket permission for the S3 bucket.
- C. Update the developer's user permissions to include the S3:ListBucket permission for the S3 bucket.
- D. Update the S3 bucket policy by including the S3:ListBucket permission and by setting the Principal element to specify the account number of the EC2 instance.

Answer: B

Explanation:

IAM instance profiles are containers for IAM roles that can be associated with EC2 instances. An IAM role is a set of permissions that grant access to AWS resources. An IAM role can be used to allow an EC2 instance to access an S3 bucket by including the appropriate permissions in the role's policy. The

S3:ListBucket permission allows listing the objects in an S3 bucket. By updating the IAM instance profile with this permission, the application on the EC2 instance can retrieve the objects from the S3 bucket and display them to the user. Reference: Using an IAM role to grant permissions to applications running on Amazon EC2 instances

NEW QUESTION 4

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 5

A developer is working on a Python application that runs on Amazon EC2 instances. The developer wants to enable tracing of application requests to debug performance issues in the code. Which combination of actions should the developer take to achieve this goal? (Select TWO)

- A. Install the Amazon CloudWatch agent on the EC2 instances.
- B. Install the AWS X-Ray daemon on the EC2 instances.
- C. Configure the application to write JSON-formatted logs to /var/log/cloudwatch.
- D. Configure the application to write trace data to /var/log/xray.
- E. Install and configure the AWS X-Ray SDK for Python in the application.

Answer: BE

Explanation:

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

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

NEW QUESTION 6

A developer is using AWS Step Functions to automate a workflow. The workflow defines each step as an AWS Lambda function task. The developer notices that runs of the Step Functions state machine fail in the GetResource task with either an `UlegalArgumentException` error or a `TooManyRequestsException` error. The developer wants the state machine to stop running when the state machine encounters a `UlegalArgumentException` error. The state machine needs to retry the GetResource task one additional time after 10 seconds if the state machine encounters a `TooManyRequestsException` error. If the second attempt fails, the developer wants the state machine to stop running. How can the developer implement the Lambda retry functionality without adding unnecessary complexity to the state machine?

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

Answer: C

Explanation:

The best way to implement the Lambda retry functionality is to use the `Retry` field in the state definition of the GetResource task. The `Retry` field allows the developer to specify an array of retriers, each with an error type, an interval, and a maximum number of attempts. By setting the error type to `TooManyRequestsException`, the interval to 10 seconds, and the maximum attempts to 1, the developer can achieve the desired behavior of retrying the GetResource task once after 10 seconds if it encounters a `TooManyRequestsException` error. If the retry fails, the state machine will stop running. If the GetResource task encounters an `UlegalArgumentException` error, the state machine will also stop running without retrying, as this error type is not specified in the `Retry` field. References:
? Error handling in Step Functions
? Handling Errors, Retries, and adding Alerting to Step Function State Machine Executions
? The Jitter Strategy for Step Functions Error Retries on the New Workflow Studio

NEW QUESTION 7

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

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

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

- A. Access on all buckets except the “DOC-EXAMPLE-BUCKET” bucket
- B. Access on all buckets that start with “DOC-EXAMPLE-BUCKET” except the “DOC-EXAMPLE-BUCKET/secrets” bucket
- C. Access on all objects in the “DOC-EXAMPLE-BUCKET” bucket along with access to all S3 actions for objects in the “DOC-EXAMPLE-BUCKET” bucket that start with “secrets”
- D. Access on all objects in the “DOC-EXAMPLE-BUCKET” bucket except on objects that start with “secrets”

Answer: D

Explanation:

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

Reference: Using IAM policies for Amazon S3

NEW QUESTION 8

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

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

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

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

Answer: C

Explanation:

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service that enables pub/sub communication between distributed systems. The developer can create an SNS topic and configure the Lambda function to publish messages with specific attributes to the topic. The developer can subscribe each partner to the SNS topic and apply the appropriate filter policy to the topic subscriptions. This way, each partner will receive updates for only their own orders based on the message attributes. This solution will meet the requirements in the most scalable way and allow adding new partners in the future with minimal code changes.

References:

? [Amazon Simple Notification Service (SNS)]

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

NEW QUESTION 9

A developer creates a static website for their department. The developer deploys the static assets for the website to an Amazon S3 bucket and serves the assets with Amazon CloudFront. The developer uses origin access control (OAC) on the CloudFront distribution to access the S3 bucket.

The developer notices users can access the root URL and specific pages but cannot access directories without specifying a file name. For example,

/products/index.html works, but /products returns an error. The developer needs to enable accessing directories without specifying a file name without exposing the S3 bucket publicly.

Which solution will meet these requirements?

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

Answer: A

Explanation:

The simplest and most efficient way to enable accessing directories without specifying a file name is to update the CloudFront distribution's settings to index.html as the default root object. This will instruct CloudFront to return the index.html object when a user requests the root URL or a directory URL for the distribution.

This solution does not require enabling static website hosting on the S3 bucket, creating a CloudFront function, or creating a custom error response. References

? Specifying a default root object

? cloudfront-default-root-object-configured

? How to setup CloudFront default root object?

? Ensure a default root object is configured for AWS Cloudfront ...

NEW QUESTION 10

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

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

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

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

K. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table

DynamoDB

L. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

M. Create a users table in DynamoDB

N. Use the table to manage user account

O. Create a Lambda authorizer that validates user credentials against the users table

P. Integrate the Lambda authorizer with API Gateway to control access to the API

Q. Use the Lambda function to store the photos in Amazon S3. Store the object's S3 key as part of the photo details in the DynamoDB table

R. Retrieve previously uploaded photos by querying DynamoDB for the S3 key.

Answer: B

Explanation:

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

References:

? [Amazon Cognito User Pools]

? [Use Amazon Cognito User Pools - Amazon API Gateway]

? [Amazon Simple Storage Service (S3)]

? [Amazon DynamoDB]

NEW QUESTION 10

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

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

Which solution will meet these requirements?

- A. Create an Amazon RDS for MySQL DB instance
- B. Store the unique identifier for each request in a database table
- C. Modify the Lambda function to check the table for the identifier before processing the request.
- D. Create an Amazon DynamoDB table
- E. Store the unique identifier for each request in the table
- F. Modify the Lambda function to check the table for the identifier before processing the request.
- G. Create an Amazon DynamoDB table

H. Store the unique identifier for each request in the table

receives a duplicate request.

I. Modify the Lambda function to return a client error response when the function

J. Create an Amazon ElastiCache for Memcached instance

K. Store the unique identifier for each request in the cache

L. Modify the Lambda function to check the cache for the identifier before processing the request.

Answer: B

Explanation:

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

NEW QUESTION 12

A developer is working on an ecommerce platform that communicates with several third-party payment processing APIs. The third-party payment services do not provide a test environment.

The developer needs to validate the ecommerce platform's integration with the third-party payment processing APIs. The developer must test the API integration code without invoking the third-party payment processing APIs.

Which solution will meet these requirements?

A. Set up an Amazon API Gateway REST API with a gateway response configured for status code 200. Add response templates that contain sample responses captured from the real third-party API.

B. Set up an AWS AppSync GraphQL API with a data source configured for each third-party API. Specify an integration type of Mock. Configure integration responses by using sample responses captured from the real third-party API.

C. Create an AWS Lambda function for each third-party API.

D. Embed responses captured from the real third-party API.

E. Configure Amazon Route 53 Resolver with an inbound endpoint for each Lambda function's Amazon Resource Name (ARN).

F. Set up an Amazon API Gateway REST API for each third-party API. Specify an integration request type of Mock. Configure integration responses by using sample responses captured from the real third-party API.

Answer: D

Explanation:

Amazon API Gateway can mock responses for testing purposes without requiring any integration backend. This allows the developer to test the API integration code without invoking the third-party payment processing APIs. The developer can configure integration responses by using sample responses captured from the real third-party API. References:

? Mocking Integration Responses in API Gateway

? Set up Mock Integrations for an API in API Gateway

NEW QUESTION 15

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

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

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

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

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

Answer: B

Explanation:

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

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

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

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

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

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

Reference: [AWS CodeDeploy lifecycle event hooks reference]

NEW QUESTION 16

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 20

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 22

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 27

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

- A. In the central AWS CDK application, write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- B. Create an AWS CloudFormation template from a JSON file
- C. Use the template to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- D. In the central AWS CDK application
- E. write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- F. Create an AWS CDK custom resource Use the custom resource to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- G. In the central AWS CDK, write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- H. Create an API in AWS Amplify Use the API to attach the function code to an AWS Lambda function and to invoke the Lambda function when the deployment stack runs.
- I. In the AWS Lambda console write a handler function in the code that uses AWS SDK calls to check for and delete unused resource
- J. Create an AWS CDK custom resource
- K. Use the custom resource to import the Lambda function into the stack and to invoke the Lambda function when the deployment stack runs.

Answer: B

Explanation:

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

configuration or integration. Creating a CloudFormation template from a JSON file will require additional configuration and integration with the central AWS CDK application. Creating an API in AWS Amplify will require additional configuration and integration with the central AWS CDK application and may not provide optimal performance or availability. Writing a handler function in the AWS Lambda console will require additional configuration and integration with the central AWS CDK application.

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

NEW QUESTION 28

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 30

A developer is migrating an application to Amazon Elastic Kubernetes Service (Amazon EKS). The developer migrates the application to Amazon Elastic Container Registry (Amazon ECR) with an EKS cluster.

As part of the application migration to a new backend, the developer creates a new AWS account. The developer makes configuration changes to the application to point the application to the new AWS account and to use new backend resources. The developer successfully tests the changes within the application by deploying the pipeline.

The Docker image build and the pipeline deployment are successful, but the application is still connecting to the old backend. The developer finds that the application's configuration is still referencing the original EKS cluster and not referencing the new backend resources.

Which reason can explain why the application is not connecting to the new resources?

- A. The developer did not successfully create the new AWS account.
- B. The developer added a new tag to the Docker image.
- C. The developer did not update the Docker image tag to a new version.
- D. The developer pushed the changes to a new Docker image tag.

Answer: C

Explanation:

The correct answer is C. The developer did not update the Docker image tag to a new version.

* C. The developer did not update the Docker image tag to a new version. This is correct. When deploying an application to Amazon EKS, the developer needs to specify the Docker image tag that contains the application code and configuration. If the developer does not update the Docker image tag to a new version after making changes to the application, the EKS cluster will continue to use the old Docker image tag that references the original backend resources. To fix this issue, the developer should update the Docker image tag to a new version and redeploy the application to the EKS cluster.

* A. The developer did not successfully create the new AWS account. This is incorrect. The creation of a new AWS account is not related to the application's connection to the

backend resources. The developer can use any AWS account to host the EKS cluster and the backend resources, as long as they have the proper permissions and configurations.

* B. The developer added a new tag to the Docker image. This is incorrect. Adding a new tag to the Docker image is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image.

* D. The developer pushed the changes to a new Docker image tag. This is incorrect. Pushing the changes to a new Docker image tag is not enough to deploy the changes to the application. The developer also needs to update the Docker image tag in the EKS cluster configuration, so that the EKS cluster can pull and run the new Docker image. References:

? 1: Amazon EKS User Guide, "Deploying applications to your Amazon EKS cluster", <https://docs.aws.amazon.com/eks/latest/userguide/deploying-applications.html>

? 2: Amazon ECR User Guide, "Pushing an image", <https://docs.aws.amazon.com/AmazonECR/latest/userguide/docker-push-ecr-image.html>

? 3: Amazon EKS User Guide, "Updating an Amazon EKS cluster", <https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html>

NEW QUESTION 31

A developer is creating an AWS Lambda function that searches for items from an Amazon DynamoDB table that contains customer contact information- The DynamoDB table items have the customer's email_address as the partition key and additional properties such as customer_type, name, and job_title.

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

What should the developer do to meet these requirements?

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

operation on the GSI by using the begins_wth key condition expression With the email_address property.

C. Add a local secondary index (LSI) to the DynamoDB table With customer_type as the partition key and email_address as the sort key Perform a query operation on the LSI by using the begins_wth key condition expression With the email_address property

D. Add a local secondary Index (LSI) to the DynamoDB table With job_title as the partition key and emad_address as the sort key Perform a query operation on the LSI by using the begins_wrth key condition expression With the email_address property

Answer: A

Explanation:

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

NEW QUESTION 36

A developer accesses AWS CodeCommit over SSH. The SSH keys configured to access AWS CodeCommit are tied to a user with the following permissions:

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

The developer needs to create/delete branches

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

A. "codecommit:CreateBranch"
"codecommit>DeleteBranch"

B. "codecommit:Put*"

C. "codecommit:Update*"

D. "codecommit:*"

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: A

Explanation:

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

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

NEW QUESTION 41

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

- A. Create a standard parameter in AWS Systems Manager Parameter Store Set Expiration and Expiration Notification policy types.
- B. Create a standard parameter in AWS Systems Manager Parameter Store Create an AWS Lambda function to expire the configuration and to send Amazon Simple Notification Service (Amazon SNS) notifications.
- C. Create an advanced parameter in AWS Systems Manager Parameter Store Set Expiration and Expiration Notification policy types.
- D. Create an advanced parameter in AWS Systems Manager Parameter Store Create an Amazon EC2 instance with a cron job to expire the configuration and to send notifications.

Answer: C

Explanation:

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

NEW QUESTION 45

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

NEW QUESTION 48

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAPIActions",
      "Effect": "Allow",
      "Action": [
        "dynamodb:GetItem",
        "dynamodb:BatchGetItem",
        "dynamodb:Scan",
        "dynamodb:Query",
        "dynamodb:ConditionCheckItem"
      ],
      "Resource": "arn:aws:dynamodb:us-west-2:account-id:table/Cars"
    }
  ]
}
```

When the application tries to read from the Cars table, an Access Denied error occurs. How can the developer resolve this error?

- A. Modify the IAM policy resource to be "arn:aws:dynamodb:us-west-2:account-id:table/*"
- B. Modify the IAM policy to include the dynamodb:* action
- C. Create a trust policy that specifies the EC2 service principal
- D. Associate the role with the policy.
- E. Create a trust relationship between the role and dynamodb.amazonaws.com.

Answer: C

Explanation:

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

NEW QUESTION 51

A developer is creating a service that uses an Amazon S3 bucket for image uploads. The service will use an AWS Lambda function to create a thumbnail of each image. Each time an image is uploaded, the service needs to send an email notification and create the thumbnail. The developer needs to configure the image processing and email notifications setup.

Which solution will meet these requirements?

- A. Create an Amazon Simple Notification Service (Amazon SNS) topic Configure S3 event notifications with a destination of the SNS topic Subscribe the Lambda function to the SNS topic Create an email notification subscription to the SNS topic
- B. Create an Amazon Simple Notification Service (Amazon SNS) topic
- C. Configure S3 event notifications with a destination of the SNS topic
- D. Subscribe the Lambda function to the SNS topic
- E. Create an Amazon Simple Queue Service (Amazon SQS) queue Subscribe the SQS queue to the SNS topic Create an email notification subscription to the SQS queue.
- F. Create an Amazon Simple Queue Service (Amazon SQS) queue Configure S3 event notifications with a destination of the SQS queue Subscribe the Lambda function to the SQS queue Create an email notification subscription to the SQS queue.
- G. Create an Amazon Simple Queue Service (Amazon SQS) queue
- H. Send S3 event notifications to Amazon EventBridge
- I. Create an EventBridge rule that runs the Lambda function when images are uploaded to the S3 bucket Create an EventBridge rule that sends notifications to the SQS queue Create an email notification subscription to the SQS queue

Answer: A

Explanation:

This solution will allow the developer to receive notifications for each image uploaded to the S3 bucket, and also create a thumbnail using the Lambda function. The SNS topic will serve as a trigger for both the Lambda function and the email notification subscription. When an image is uploaded, S3 will send a notification to the SNS topic, which will trigger the Lambda function to create the thumbnail and also send an email notification to the specified email address.

NEW QUESTION 55

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

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

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

Which solution will meet these requirements with the LEAST complexity?

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

Answer: A

Explanation:

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

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

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

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

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

NEW QUESTION 59

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 62

A developer is building a serverless application by using AWS Serverless Application Model (AWS SAM) on multiple AWS Lambda functions.

When the application is deployed, the developer wants to shift 10% of the traffic to the new deployment of the application for the first 10 minutes after deployment.

If there are no issues, all traffic must switch over to the new version.

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

- A. Set the Deployment Preference Type to Canary10Percent10Minute

AutoPublishAlias property to the Lambda alias.

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

Answer: A

Explanation:

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

NEW QUESTION 63

An organization is using Amazon CloudFront to ensure that its users experience low- latency access to its web application. The organization has identified a need to encrypt all traffic between users and CloudFront, and all traffic between CloudFront and the web application. How can these requirements be met? (Select TWO)

- A. Use AWS KMS to encrypt traffic between cloudFront and the web application.
- B. Set the Origin Protocol Policy to "HTTPS Only".
- C. Set the Origin's HTTP Port to 443.
- D. Set the Viewer Protocol Policy to "HTTPS Only" or Redirect HTTP to HTTPS"
- E. Enable the CloudFront option Restrict Viewer Access.

Answer: BD

Explanation:

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

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

NEW QUESTION 68

A developer has created an AWS Lambda function that makes queries to an Amazon Aurora MySQL DB instance. When the developer performs a test the DB instance shows an error for too many connections. Which solution will meet these requirements with the LEAST operational effort?

- A. Create a read replica for the DB instance Query the replica DB instance instead of the primary DB instance.
- B. Migrate the data to an Amazon DynamoDB database.
- C. Configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment.
- D. Create a proxy in Amazon RDS Proxy Query the proxy instead of the DB instance.

Answer: D

Explanation:

This solution will meet the requirements by using Amazon RDS Proxy, which is a fully managed, highly available database proxy for Amazon RDS that makes applications more scalable, more resilient to database failures, and more secure. The developer can create a proxy in Amazon RDS Proxy, which sits between the application

and the DB instance and handles connection management, pooling, and routing. The developer can query the proxy instead of the DB instance, which reduces the number of open connections to the DB instance and avoids errors for too many connections. Option A is not optimal because it will create a read replica for the DB instance, which may not solve the problem of too many connections as read replicas also have connection limits and may incur additional costs. Option B is not optimal because it will migrate the data to an Amazon DynamoDB database, which may introduce additional complexity and overhead for migrating and accessing data from a different database service. Option C is not optimal because it will configure the Amazon Aurora MySQL DB instance for Multi-AZ deployment, which may improve availability and durability of the DB instance but not reduce the number of connections.

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

NEW QUESTION 73

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

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

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

- A. Host each website by using AWS Amplify with a serverless backend
- B. Connect the repository branches that correspond to each of the desired environment
- C. Start deployments by merging code changes to a desired branch.
- D. Host each website in AWS Elastic Beanstalk with multiple environment
- E. Use the EB CLI to link each repository branch
- F. Integrate AWS CodePipeline to automate deployments from version control code merges.
- G. Host each website in different Amazon S3 buckets for each environment
- H. Configure AWS CodePipeline to pull source code from version control

- I. Add an AWS CodeBuild stage to copy source code to Amazon S3.
- J. Host each website on its own Amazon EC2 instance
- K. Write a custom deployment script to bundle each website's static asset
- L. Copy the assets to Amazon EC2. Set up a workflow to run the script when code is merged.

Answer: A

Explanation:

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

NEW QUESTION 74

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

Which solution will meet these requirements MOST securely?

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

Answer: D

Explanation:

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

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

NEW QUESTION 78

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

Which solution should the developer implement to meet these requirements?

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

Answer: C

Explanation:

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

Reference: End-to-end testing

NEW QUESTION 80

A developer is working on a web application that uses Amazon DynamoDB as its data store The application has two DynamoDB tables one table that is named artists and one table that is named songs The artists table has artistName as the partition key. The songs table has songName as the partition key and artistName as the sort key

The table usage patterns include the retrieval of multiple songs and artists in a single database operation from the webpage. The developer needs a way to retrieve this information with minimal network traffic and optimal application performance.

Which solution will meet these requirements?

- A. Perform a BatchGetItem operation that returns items from the two tables
- B. Use the list of songName artistName keys for the songs table and the list of artistName key for the artists table.
- C. Create a local secondary index (LSI) on the songs table that uses artistName as the partition key Perform a query operation for each artistName on the songs table that filters by the list of songName Perform a query operation for each artistName on the artists table
- D. Perform a BatchGetItem operation on the songs table that uses the songName/artistName key
- E. Perform a BatchGetItem operation on the artists table that uses artistName as the key.
- F. Perform a Scan operation on each table that filters by the list of songName/artistName for the songs table and the list of artistName in the artists table.

Answer: A

Explanation:

BatchGetItem can return one or multiple items from one or more tables. For reference check the link below
https://docs.aws.amazon.com/amazondynamodb/latest/APIReference/API_BatchGetItem.html

NEW QUESTION 81

A company is migrating legacy internal applications to AWS. Leadership wants to rewrite the internal employee directory to use native AWS services. A developer needs to create a solution for storing employee contact details and high-resolution photos for use with the new application. Which solution will enable the search and retrieval of each employee's individual details and high-resolution photos using AWS APIs?

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

Answer: B

Explanation:

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

References:

? [Amazon DynamoDB]

? [Amazon Simple Storage Service (S3)]

NEW QUESTION 86

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

Which solution will meet these requirements MOST cost-effectively?

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

Answer: A

Explanation:

The most cost-effective solution is to use the DefinitionSubstitutions property of the AWS::StepFunctions::StateMachine resource to inject the API endpoint as a variable in the state machine definition. This way, the developer can use the intrinsic function Fn::GetAtt to get the API endpoint from the AWS::ApiGateway::RestApi resource, and pass it to the state machine without creating any additional resources or environment variables. The other solutions involve creating and managing extra resources, such as Secrets Manager secrets or AppConfig configuration profiles, which incur additional costs and complexity. References

? AWS::StepFunctions::StateMachine - AWS CloudFormation

? Call API Gateway with Step Functions - AWS Step Functions

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

NEW QUESTION 91

A developer has a legacy application that is hosted on-premises. Other applications hosted on AWS depend on the on-premises application for proper functioning. In case of any application errors, the developer wants to be able to use Amazon CloudWatch to monitor and troubleshoot all applications from one place. How can the developer accomplish this?

- A. Install an AWS SDK on the on-premises server to automatically send logs to CloudWatch.
- B. Download the CloudWatch agent to the on-premises server.
- C. Configure the agent to use IAM user credentials with permissions for CloudWatch.
- D. Upload log files from the on-premises server to Amazon S3 and have CloudWatch read the files.
- E. Upload log files from the on-premises server to an Amazon EC2 instance and have the instance forward the logs to CloudWatch.

Answer: B

Explanation:

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

References:

? [What Is Amazon CloudWatch? - Amazon CloudWatch]

? [Installing and Configuring the CloudWatch Agent - Amazon CloudWatch]

NEW QUESTION 93

A developer deployed an application to an Amazon EC2 instance. The application needs to know the public IPv4 address of the instance.

How can the application find this information?

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

A. Query the instance user data from `http://169.254.169.254/latest/user-data/`

C. Query the Amazon Machine Image (AMI) information from `http://169.254.169.254/latest/meta-data/ami/`.

D. Check the hosts file of the operating system

Answer: A

Explanation:

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

? Instance metadata and user data

? Get Public IP Address on current EC2 Instance

? Get the public ip address of your EC2 instance quickly

NEW QUESTION 96

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

A. Amazon S3 managed keys

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

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

D. Symmetric customer managed keys with imported key material

Answer: B

Explanation:

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

Reference: Using AWS KMS keys to encrypt S3 objects

NEW QUESTION 101

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

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

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

B. Save the details of the uploaded files in a separate Amazon DynamoDB table

C. Filter the list of files in the user interface (UI) by comparing the current user ID with the user ID associated with the file in the table.

D. Use Amazon API Gateway and an AWS Lambda function to upload and download file

E. Validate each request in the Lambda function before performing the requested operation.

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

Answer: D

Explanation:

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

NEW QUESTION 105

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 Gateway
- 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 107

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 ECSLinear10PercentEvery1 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 112

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

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

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

Answer: B

Explanation:

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

Reference: Asynchronous invocation

NEW QUESTION 115

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

Which encryption option will meet these requirements?

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

Answer: B

Explanation:

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

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

NEW QUESTION 118

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 121

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

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

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

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

Answer: A

Explanation:

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

References:

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

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

NEW QUESTION 124

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 interval
- 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 128

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 130

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

Which solution will meet these requirements?

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

Answer: A

Explanation:

This solution will meet the requirements by using Amazon CloudFront, which is a content delivery network (CDN) service that speeds up the delivery of web content and APIs to end users. The developer can configure the CloudFront cache, which is a set of edge locations that store copies of popular or recently accessed content close to the viewers. The developer can also update the application to return cached content based upon the default request headers, which are a set of HTTP headers that CloudFront automatically forwards to the origin server and uses to determine whether an object in an edge location is still valid. By caching the POST requests, the developer can optimize the API resources and reduce the latency for repeated queries. Option B is not optimal because it will override the cache method in the selected stage of API Gateway, which is not possible or effective as API Gateway does not support caching for POST methods by default. Option C is not optimal because it will save the latest request response in Lambda /tmp directory, which is a local storage space that is available for each Lambda function invocation, not a cache that can be shared across multiple invocations or requests. Option D is not optimal because it will save the latest request in AWS Systems Manager Parameter Store, which is a service that provides secure and scalable storage for configuration data and secrets, not a cache for API responses.

References: [Amazon CloudFront], [Caching Content Based on Request Headers]

NEW QUESTION 133

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

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

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

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

- E. Set up AWS CodeDeploy to deploy the most recent version of the application at runtime.
- F. Set up AWS CodePipeline to deploy the most recent version of the application at runtime.
- G. Remove any commands that perform operating system patching from the UserData script.

Answer: BE

Explanation:

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

- ? Set up AWS CodeDeploy to deploy the most recent version of the application at runtime. This will ensure that the application code is always up to date and does not depend on the AMI.
- ? Remove any commands that perform operating system patching from the UserData script. This will reduce the time that the UserData script takes to run and speed up the instance launch process.

References:

- ? [What Is AWS CloudFormation? - AWS CloudFormation]
- ? [What Is AWS CodeDeploy? - AWS CodeDeploy]
- ? [Running Commands on Your Linux Instance at Launch - Amazon Elastic Compute Cloud]

NEW QUESTION 135

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 tabl
- B. Configure the table with a primary key that consists of the title as the partition key and the release year as the sort ke
- 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 tabl
- E. Configure the table with a primary key that consists of the genre as the partition key and the release year as the sort ke

- 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]

NEW QUESTION 138

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

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

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

Answer: B

Explanation:

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

? AWS X-Ray SDK for Python

? Instrumenting a Python Application

NEW QUESTION 141

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

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

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

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

? How to Deploy Docker Containers | AWS

- ? Deploy a Web App Using AWS App Runner
- ? How to Deploy Containerized Apps on AWS Using ECR and Docker

NEW QUESTION 143

An application that is deployed to Amazon EC2 is using Amazon DynamoDB. The application calls the DynamoDB REST API. Periodically, the application receives a `ProvisionedThroughputExceededException` error when the application writes to a DynamoDB table.

Which solutions will mitigate this error MOST cost-effectively? (Select TWO)

- A. Modify the application code to perform exponential back off when the error is received.
- B. Modify the application to use the AWS SDKs for DynamoDB.
- C. Increase the read and write throughput of the DynamoDB table.
- D. Create a DynamoDB Accelerator (DAX) cluster for the DynamoDB table.
- E. Create a second DynamoDB table. Distribute the reads and writes between the two tables.

Answer: AB

Explanation:

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

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

NEW QUESTION 147

A company has a social media application that receives large amounts of traffic. User posts and interactions are continuously updated in an Amazon RDS database. The data changes frequently, and the data types can be complex. The application must serve read requests with minimal latency. The application's current architecture struggles to deliver these rapid data updates efficiently. The company needs a solution to improve the application's performance.

Which solution will meet these requirements?

- A. Mastered
- B. Not Mastered

Answer: A

Explanation:

Creating an Amazon ElastiCache for Redis cluster is the best solution for improving the application's performance. Redis is an in-memory data store that can serve read requests with minimal latency and handle complex data types, such as lists, sets, hashes, and streams. By using a write-through caching strategy, the application can ensure that the data in Redis is always consistent with the data in RDS. The application can read the data from Redis instead of RDS, reducing the load on the database and improving the response time. The other solutions are either not feasible or not effective. Amazon DynamoDB Accelerator (DAX) is a caching service that works only with DynamoDB, not RDS. Amazon S3 Transfer Acceleration is a feature that speeds up data transfers between S3 and clients across the internet, not between RDS and the application. Amazon CloudFront is a content delivery network that can cache static content, such as images, videos, or HTML files, but not dynamic content, such as user posts and interactions. References

- ? Amazon ElastiCache for Redis
- ? Caching Strategies and Best Practices - Amazon ElastiCache for Redis
- ? Using Amazon ElastiCache for Redis with Amazon RDS
- ? Amazon DynamoDB Accelerator (DAX)
- ? Amazon S3 Transfer Acceleration
- ? Amazon CloudFront

NEW QUESTION 150

A developer is modifying an existing AWS Lambda function. While checking the code, the developer notices hardcoded parameter values for an Amazon RDS for SQL Server user name, password, database, host, and port. There also are hardcoded parameter values for an Amazon DynamoDB table, an Amazon S3 bucket, and an Amazon Simple Notification Service (Amazon SNS) topic. The developer wants to securely store the parameter values outside the code in an encrypted format and wants to turn on rotation for the credentials. The developer also wants to be able to reuse the parameter values from other applications and to update the parameter values without modifying code. Which solution will meet these requirements with the LEAST operational overhead?

- A. Create an RDS database secret in AWS Secrets Manager.
- B. Set the user name, password, database, host, and port in environment variables for the Lambda function.
- C. Turn on secret rotation for the RDS database secret.
- D. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic.
- E. Create an RDS database secret in AWS Secrets Manager.
- F. Set the user name, password, database, host, and port in environment variables for the Lambda function.
- G. Turn on secret rotation for the RDS database secret.
- H. Create Secure String parameters in AWS Systems Manager Parameter Store for the DynamoDB table, S3 bucket, and SNS topic.
- I. Create RDS database parameters in AWS Systems Manager Parameter Store.
- J. Store the user name, password, database, host, and port in environment variables for the Lambda function.
- K. Create encrypted Lambda environment variables for the DynamoDB table, S3 bucket, and SNS topic.
- L. Create a Lambda function and set the logic for the credentials rotation task. Schedule the credentials rotation task in Amazon EventBridge.
- M. Create RDS database parameters in AWS Systems Manager Parameter Store.
- N. Store the user name, password, database, host, and port in environment variables for the Lambda function.
- O. Store the DynamoDB table name, S3 bucket, and SNS topic in Amazon S3. Create a Lambda function and set the logic for the credentials rotation. Invoke the Lambda function on a schedule.
- P. Store the user name, password, database, host, and port in environment variables for the Lambda function.

Answer: B

Explanation:

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

environment variables for the

DynamoDB table, S3 bucket, and SNS topic, which may not be reusable or updatable without modifying code. Option C is not optimal because it will create RDS database parameters in AWS Systems Manager Parameter Store, which does not support automatic rotation of secrets. Option D is not optimal because it will store the DynamoDB table, S3 bucket, and SNS topic in Amazon S3, which may introduce additional costs and complexity for accessing configuration data. References: AWS Secrets Manager, [AWS Systems Manager Parameter Store]

NEW QUESTION 153

.....

Thank You for Trying Our Product

We offer two products:

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

2nd - Questions and Answers in PDF Format

AWS-Certified-Developer-Associate Practice Exam Features:

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

100% Actual & Verified — Instant Download, Please Click
[Order The AWS-Certified-Developer-Associate Practice Test Here](#)