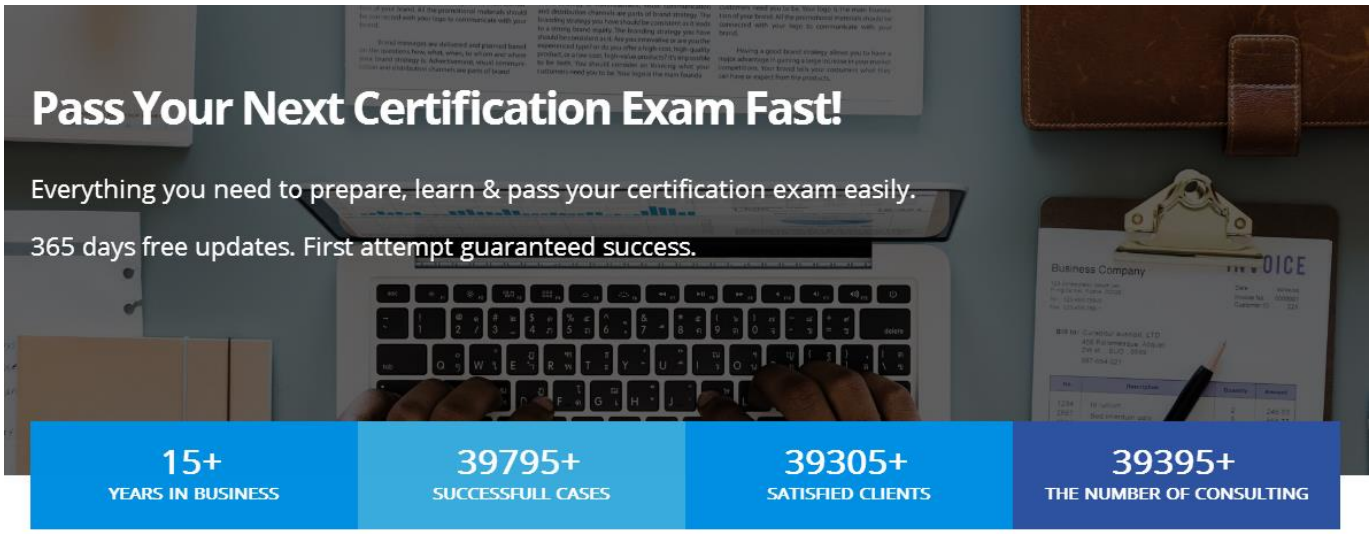
PracticeVCE



TRY BEFORE YOU BUY

Download a free sample of any of our exam questions and answers

- 24/7 customer support, Secure shopping site
- Free One year updates to match real exam scenarios
- If you failed your exam after buying our products we will refund the full amount back to you.

365 Days Free Updates

Free update is available within 365 days after your purchase. After 365 days, you will get 50% discounts for updating.

Security & Privacy

We respect customer privacy. We use McAfee's security service to provide you with utmost security for your personal information & peace of mind.

Instant Download

After Payment, our system will send you the products you purchase in mailbox in a minute after payment. If not received within 2 hours, please contact us.

Money Back Guarantee

Full refund if you fail the corresponding exam in 60 days after purchasing. And Free get any another product.

http://www.practicevce.com

Professional Study Tool and Reliable Exam Practice Material

Exam : **AWS-Certified-Developer-Associate-JP**

Title : AWS Certified Developer - Associate (AWS-Developer日本語版)

Vendor : Amazon

Version : DEMO

QUESTION NO: 1
開発者は、Amazon S3 バケットに新しいファイルが追加されるとすぐに、Amazon DynamoDB テーブルにレコードを挿入したいと考えています。
これを達成するにはどのような一連の手順が必要でしょうか?
A. Amazon EventBridge を使用して、S3 バケットを監視し、レコードを DynamoDB に挿入するイベントを作成します。
B. DynamoDB にレコードを挿入する AWS Lambda 関数を呼び出すように S3 イベントを設定します。
C. S3 バケットをポーリングし、レコードを DynamoDB に挿入する AWS Lambda 関数を作成します。
D. スケジュールされた時間に実行され、レコードを DynamoDB に挿入する cron ジョブを作成します。
Answer: B
Amazon S3 is a service that provides highly scalable, durable, and secure object storage. Amazon DynamoDB is a fully managed NoSQL database service that provides fast and consistent performance with seamless scalability. AWS Lambda is a service that lets developers run code without provisioning or managing servers.
The developer can configure an S3 event to invoke a Lambda function that inserts records into DynamoDB whenever a new file is added to the S3 bucket. This solution will meet the requirement of inserting a record into DynamoDB as soon as a new file is added to S3.
References:
[Amazon Simple Storage Service (S3)]
[Amazon DynamoDB]
[What Is AWS Lambda? - AWS Lambda]
[Using AWS Lambda with Amazon S3 - AWS Lambda]

QUESTION NO: 2
ある企業には、製品カタログを含むオンラインWebアプリケーションがあります。このカタログは、DOC-EXAMPLE-BUCKETという名前のAmazon S3バケットに保存されています。アプリケーションは、S3バケット内のオブジェクトを一覧表示し、1AMポリシーを使用してオブジェクトをダウンロードできる必要があります。
これらの要件を満たすために最小限のアクセスを許可するポリシーはどれですか?

B.

```
{
  "Effect": "Allow",
  "Action": [
    "s3:*"
  ],
  "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
}
]
}

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:ListBucket",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    },
    {
```

C.

```
      "Effect": "Allow",
      "Action":[
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    }
  ]
}

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":"s3:ListBucket",
      "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
    },
```

A. オプションA
B. オプションB
C. オプションC
D. オプションD
Answer: A

QUESTION NO: 3
開発者は、Amazon
EC2インスタンス群へのトラフィックを分析するアプリケーションを作成しています。EC2インスタンスはパブリックApplication Load
Balancer（ALB）の背後で稼働しています。各EC2インスタンス上でHTTPサーバーが稼働し、すべてのリクエストをログファイルに記録します。
開発者はクライアントのパブリックIPアドレスを取得したいと考えています。ログファイルを分析した結果、ALBのIPアドレスのみに気づきました。
ログ ファイルにクライアントのパブリック IP
アドレスをキャプチャするには、開発者は何をする必要がありますか?
A. HTTP サーバー ログ構成ファイルに Host ヘッダーを追加します。
B. 各EC2インスタンスにAmazon CloudWatch
Logsエージェントをインストールします。エージェントがログファイルに書き込むように設定します。
C. 各EC2インスタンスにAWS X-
Rayデーモンをインストールします。デーモンがログファイルに書き込むように設定します。
D. HTTP サーバー ログ構成ファイルに X-Forwarded-For ヘッダーを追加します。
Answer: D

QUESTION NO: 4
開発者は、本番環境のAmazon API Gateway
APIにリクエスト検証機能を追加したいと考えています。開発者は、APIを本番環境にデプロイする前に、変更内容をテストする必要があります。テストでは、テストツールを介してAPIにテストリクエストを送信します。
最も少ない運用オーバーヘッドでこれらの要件を満たすソリューションはどれでしょうか?
A.
既存のAPIをOpenAPIファイルにエクスポートします。新しいAPIを作成します。OpenAPIファイルをインポートします。新しいAPIを変更してリクエストの検証を追加します。テストを実行します。既存のAPIを変更してリクエストの検証を追加します。
既存の API を本番環境にデプロイします。
B. 既存のAPIを変更してリクエスト検証を追加します。更新したAPIを新しいAPI
Gatewayステージにデプロイします。テストを実行します。更新したAPIをAPI
Gatewayの本番ステージにデプロイします。
C.
新しいAPIを作成します。新しいリクエスト検証を含む必要なリソースとメソッドを追加します。テストを実行します。既存のAPIを修正してリクエスト検証を追加します。既存のAPIを本番環境にデプロイします。
D.
既存のAPIを複製します。新しいAPIを修正してリクエスト検証を追加します。テストを実行します。既存のAPIを修正してリクエスト検証を追加します。既存のAPIを本番環境にデプロイします。
Answer: B

Amazon API Gateway allows you to create, deploy, and manage a RESTful API to expose backend HTTP endpoints, AWS Lambda functions, or other AWS services1. You can use API Gateway to perform basic validation of an API request before proceeding with the integration request1. When the validation fails, API Gateway immediately fails the request, returns a 400 error response to the caller, and publishes the validation results in CloudWatch Logs1.
To test changes before deploying to a production environment, you can modify the existing API to add request validation and deploy the updated API to a new API Gateway stage1. This allows you to perform tests without affecting the production environment. Once testing is complete and successful, you can then deploy the updated API to the API Gateway production stage1.
This approach has the least operational overhead as it avoids unnecessary creation of new APIs or exporting and importing of APIs. It leverages the existing infrastructure and only requires changes in the configuration of the existing API1.

QUESTION NO: 5
ある企業がAWS上でアプリケーションを運用しています。アプリケーションはAmazon DynamoDBテーブルにデータを保存しています。一部のクエリの実行に時間がかかっています。これらの遅いクエリには、テーブルのパーティションキーやソートキー以外の属性が関係しています。アプリケーションがDynamoDBテーブルに保存するデータ量は今後大幅に増加すると予想されます。開発者はクエリのパフォーマンスを向上させる必要があります。
これらの要件を満たすソリューションはどれでしょうか?
A. Limit パラメータをデフォルト値よりも高く設定して、各要求のページサイズを増やします。プロビジョニングされたスループットを超える要求を再試行するようにアプリケーションを構成します。
B.
グローバルセカンダリインデックス（GSI）を作成します。クエリ属性をインデックスのパーティションキーに設定します。
C.
パラメータで個別のスキャン要求を発行し、スキャン要求のセグメントと並列スキャンの合計セグメント数を指定して、並列スキャン操作を実行します。
D. DynamoDB
テーブルの読み取りキャパシティーの自動スケーリングをオンにします。最大読み取りキャパシティーユニット (RCU) を増やします。
Answer: B
* Global Secondary Index (GSI): GSIs enable alternative query patterns on a DynamoDB table by using different partition and sort keys.
* Addressing Query Bottleneck: By making the slow-query attribute the GSI's partition key, you optimize queries on that attribute.
* Scalability: GSIs automatically scale to handle increasing data volumes.
References:
Amazon DynamoDB Global Secondary Indexes:
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GSI.html

QUESTION NO: 6
あるチームがAmazon EC2インスタンスにデプロイするアプリケーションを開発しています。テスト中にエラーが発生し、EC2インスタンスがAmazon S3バケットにアクセスできなくなりました。
この問題を解決するためにチームはどのような手順を踏む必要がありますか? (2つ選択してください)。
A. EC2 インスタンスにアタッチされている JAM ロールに割り当てられたポリシーが Amazon S3 へのアクセスを許可するかどうかを確認します。
B. S3 バケットポリシーをチェックして、S3 バケットのアクセス権限を検証します。
C. EC2 インスタンスにアタッチされている 1AM ユーザーに割り当てられているポリシーが Amazon S3 へのアクセスを許可するかどうかを確認します。
D. S3 ライフサイクル ポリシーをチェックして、S3 バケットに割り当てられているアクセス許可を検証します。
E. EC2インスタンスに割り当てられているセキュリティグループを確認してください。Amazon S3へのアクセスをブロックするルールがないことを確認してください。
Answer: A,B

QUESTION NO: 7
ある企業が、アプリケーションの一連のテストを実行するためのAWS Step Functionsステートマシンを作成しています。テストは、特定のAWS Cloud Formationスタックがデプロイされたときに実行する必要があります。
どの手順の組み合わせがこれらの要件を満たしますか? (2 つ選択してください)。
A. ステートマシンを呼び出す AWS Lambda 関数を作成します。
B. CloudFormation スタックのステータス変更の詳細タイプ、UPDATE_IN_PROGRESS のステータス、および CloudFormation スタックのスタック ID に一致する、デフォルトのバス上に Amazon EventBridge ルールを作成します。
C. Amazon EventBridge Pipes に、デフォルトのイベントバスをソースとするパイプを作成します。Lambda 関数をターゲットとして設定します。詳細タイプとして CloudFormation スタックのステータス変更、ステータスとして UPDATE_IN_PROGRESS、CloudFormation スタックのスタック ID でフィルタリングします。
D. Amazon EventBridge Pipes に、EventBridgeルールのソースを持つパイプを作成します。ターゲットとしてステートマシンを設定します。
E. ステートマシンを EventBridge ルールのターゲットとして追加します。
Answer: A,E
Requirement Summary:
* Trigger an AWS Step Functions state machine (test execution)
* Only when a specific AWS CloudFormation stack is deployed
Option A: Create a Lambda function to invoke the state machine
* # Valid approach: Lambda can be used as an intermediary trigger for Step Functions using the SDK (e.g., StartExecution API).
* Offers flexibility (custom filtering, additional logic).

Option B: Create EventBridge rule filtering on UPDATE_IN_PROGRESS
* # Incorrect: UPDATE_IN_PROGRESS triggers before the stack is fully deployed.
* You need to trigger after deployment, such as UPDATE_COMPLETE or CREATE_COMPLETE.
Option C: EventBridge Pipes with Lambda target filtering on UPDATE_IN_PROGRESS
* # Incorrect for same reason as B (wrong timing).
* Also, EventBridge Pipes are not necessary here if you're using rules directly.
Option D: Pipe with EventBridge Rule as source and Step Functions as target
* # Invalid setup: EventBridge Pipes use event sources, not rules, as input.
* This configuration is unsupported.
Option E: Add the state machine as a target of the EventBridge rule
* # Direct and low-overhead approach.
* EventBridge natively supports Step Functions as a target.
* You can trigger the state machine without a Lambda if the filter matches (e.g., ResourceStatus = CREATE_COMPLETE, with the correct StackId).
* Step Functions as EventBridge target: https://docs.aws.amazon.com/eventbridge/latest/userguide/eventbridge-target-step-functions.html
* EventBridge CloudFormation events: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-listing-event-history.html
* StartExecution API: https://docs.aws.amazon.com/step-functions/latest/apireference/API_StartExecution.html

QUESTION NO: 8
開発者が金融アプリケーションを開発しています。このアプリケーションは、AWS Secrets Manager を使用して Amazon RDS for PostgreSQL データベースのユーザー名とパスワードを管理しています。開発者は、アプリケーションの高可用性を維持しながら、パスワードをローテーションさせる必要があります。これらの要件を、開発工数を最小限に抑えて満たすソリューションはどれでしょうか？
A.
交代ユーザーローテーション戦略を使用してシークレットをローテーションします。認証失敗に対処するために、適切な再試行戦略を使用してアプリケーションを更新します。
B.
PostgreSQLクライアントを使用して、新しいデータベースのユーザー名とパスワードを作成します。新しいシークレット値を追加するには、即時ローテーションを実行します。AWS CLIを使用してRDSデータベースのパスワードを更新します。
Secrets Manager シークレットの即時ローテーションを実行します。
C.
複数値回答ローテーションを使用してシークレットをローテーションします。認証失敗に対処するために、適切な再試行戦略でアプリケーションを更新します。
D.
シングルユーザーローテーション戦略を使用してシークレットをローテーションします。認証失敗に対処するために、適切な再試行戦略を使用してアプリケーションを更新します。
Answer: D

Requirement Summary:
* Secrets managed in AWS Secrets Manager
* DB: Amazon RDS for PostgreSQL
* Need automated password rotation
* Must maintain high availability
* Least development effort
Rotation Strategies:
Single-user rotation strategy
* # Simplest to implement
* The secret contains one set of credentials used by app and rotation logic
* # Supports automated rotation
* AWS provides built-in Lambda rotation templates for RDS
A). Alternating-users strategy
* ## More complex
* Requires application to switch users during rotation window
B). Manual secret + CLI rotation
* # Too much manual work
* Not scalable or reliable
C). Multivalue answer rotation
* # Not a valid strategy in this context
* Doesn't apply to Secrets Manager
* Secrets Manager rotation strategies:
https://docs.aws.amazon.com/secretsmanager/latest/userguide
/rotating-secrets.html
* RDS PostgreSQL secret rotation:
https://docs.aws.amazon.com/secretsmanager/latest/userguide
/rotating-secrets_strategies.html#rotating-secrets-single-user

QUESTION NO: 9
ある企業は、Webアプリケーションのセッション情報をAmazon DynamoDBテーブルにキャッシュしています。このテーブルから古い項目を自動的に削除する方法を探しています。
これを行う最も簡単な方法は何ですか?
A. 古いレコードを削除するスクリプトを作成し、Amazon EC2 インスタンスで cron ジョブとしてスクリプトをスケジュールします。
B. 有効期限を持つ属性を追加し、その属性に基づいて Time To Live 機能を有効にします。
C. 毎日、セッション
データを保持するための新しいテーブルを作成し、前日のテーブルを削除します。
D. 有効期限を持つ属性を追加します。属性の名前は ItemExpiration です。
Answer: B

QUESTION NO: 10
ある開発者がレガシーアプリケーションをAWS Lambda関数に移行しました。この関数は、サードパーティのサービスを利用して、毎月末に一連のAPI呼び出しでデータを取得します。そして、そのデータを処理して月次レポート

を生成します。この関数は今のところ問題なく動作しています。
サードパーティのサービスでは最近、API呼び出しのフィード回数を1分あたりおよび1日あたりに制限する制限が設けられました。API呼び出しが1分あたりまたは1日あたりの制限を超えた場合、サービスはエラーを生成します。APIは、レスポンスヘッダーで分数制限と1日あたりの制限も提供します。この制限により、プロセスが利用可能な制限を超えるAPI呼び出しを消費するため、プロセス全体が複数日に及ぶ可能性があります。
この変更に対応するためにサーバーレス
アプリケーションをリファクタリングする最も運用効率の高い方法は何ですか?

A. AWS Step Functions ステートマシンを使用して API 障害を監視します。Wait ステートを使用して Lambda 関数の呼び出しを遅延させます。

B. Amazon Simple Queue Service (Amazon SQS) キューを使用して API 呼び出しを保持します。Lambda 関数を設定して、API しきい値制限内でキューをポーリングします。

C. Amazon CloudWatch Logs メトリクスを使用して API 呼び出し回数をカウントします。メトリクスが API しきい値制限を超えた場合に、現在実行中の Lambda 関数のインスタンスをフラットスロップする Amazon CloudWatch アラームを設定します。

D. Amazon Kinesis Data Firehose を使用して API 呼び出しをバッチ処理し、イベント通知で Amazon S3 バケットに配信して Lambda 関数を呼び出します。

Answer: A

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

QUESTION NO: 11

データ可視化企業は、コアアプリケーションのセキュリティ強化を目指しています。アプリケーションは、開発ステージング、プレプロダクション、本番環境全体にわたってAWS上にデプロイされています。企業は、保管されているすべての機密認証情報を暗号化する必要があります。機密認証情報は自動的にローテーションする必要があります。機密認証情報は環境ごとに保存する必要があります。これらの要件を、運用効率を最も高める方法で満たすソリューションはどれでしょうか？

A. AWS Secrets Managerのバージョンを設定して、複数の環境に同じ認証情報の異なるコピーを保存する

B. 各環境の AWS Systems Manager パラメータストアに新しいパラメータバージョンを作成し、環境固有の認証情報をパラメータバージョンに保存します。

C.
アプリケーションコードで環境変数を設定します。環境タイプごとに異なる名前を使用しま

す。
D. AWS Secrets
Managerを設定して、環境タイプごとに新しいシークレットを作成します。環境固有の認証情報をシークレットに保存します。
Answer: D
* Secrets Management: AWS Secrets Manager is designed specifically for storing and managing sensitive credentials.
* Environment Isolation: Creating separate secrets for each environment (development, staging, etc.) ensures clear separation and prevents accidental leaks.
* Automatic Rotation: Secrets Manager provides built-in rotation capabilities, enhancing security posture.
* Versioning: Tracking changes to secrets is essential for auditing and compliance.
References:
AWS Secrets Manager: https://aws.amazon.com/secrets-manager/
Secrets Manager Rotation:
https://docs.aws.amazon.com/secretsmanager/latest/userguide/rotating-secrets.html

QUESTION NO: 12
ある企業は、AWS
Lambda関数を使用してサーバーレスアプリケーションを開発しています。Lambda関数の1つは、Amazon RDS
DBインスタンスにアクセスする必要があります。このDBインスタンスは、VPC内のプライベートサブネットにあります。
企業はDBインスタンスへのアクセスに必要な権限を含むロールを作成します。そして、そのロールをLambda関数に割り当てます。開発者は、Lambda関数にDBインスタンスへのアクセスを許可するために追加のアクションを実行する必要があります。
これらの要件を満たすために開発者は何をすべきでしょうか?
A.
DBインスタンスにパブリックIPアドレスを割り当てます。DBインスタンスのセキュリティグループを変更し、Lambda関数のIPアドレスからのインバウンドトラフィックを許可します。
B. Lambda 関数と DB インスタンスの間に AWS Direct Connect 接続を設定します。
C. Amazon CloudFront ディストリビューションを設定して、Lambda 関数と DB インスタンス間の安全な接続を作成します。
D.
VPC内のプライベートサブネットに接続するようにLambda関数を設定します。Lambda関数からDBインスタンスへのトラフィックを許可するためのセキュリティグループルールを追加します。
Answer: D

QUESTION NO: 13
開発者がAmazon Elastic Container Service (Amazon ECS)
に新しいアプリケーションをデプロイしようとしています。開発者は、様々な種類の変数を安全に保存および取得する必要があります。これらの変数には、リモートAPIの認証情報、APIのURL、認証情報が含まれます。認証情報とAPI

URLは、開発環境、テスト環境、本番環境のすべてにおいて、現在および将来デプロイされるすべてのバージョンのアプリケーションで利用可能である必要があります。
開発者は、アプリケーションの変更を最小限に抑えながら変数を取得するにはどうすればよいでしょうか?

A. AWS Systems Manager
パラメータストアから変数を取得するようにアプリケーションを更新します。パラメータストアでは、各環境の各変数に固有のパスを使用します。認証情報は各環境の AWS Secrets Manager に保存します。

B. AWS Key Management Service (AWS KMS)
から変数を取得するようにアプリケーションを更新します。
API URL と資格情報を環境ごとに一意のキーとして保存します。

C.
アプリケーションを更新して、アプリケーションに保存されている暗号化されたファイルから変数を取得します。
API URL と資格情報を環境ごとに固有のファイルに保存します。

D.
デプロイされた各環境から変数を取得するようにアプリケーションを更新します。ECSタスク定義内の認証情報とAPI URLを、デプロイプロセス中に一意の名前で定義します。

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

QUESTION NO: 14
開発者は、通常ユーザーとゲストユーザーの2種類のユーザー向けにWebおよびモバイルアプリケーションを構築しています。通常ユーザーはログインが必要ですが、ゲストユーザーはログインする必要はありません。ユーザーは認証の有無にかかわらず、自分のデータのみを閲覧できる必要があります。ユーザーがAWSリソースにアクセスするには、AWS認証情報が必要です。

A. Amazon Cognito ID
プールを使用して、必要なリソースにアクセスできる認証されていないロールにリンクされた一時的な AWS 認証情報を生成します。

B.
必要なリソースへの権限を持つIAMユーザーを設定します。Webアプリケーションとモバイルアプリケーションに IAM認証情報をハードコードします。

C. AWS KMS

に保存される一時キーを生成します。必要なリソースにアクセスするには、この一時キーを使用します。
D. 一時的な認証情報を生成します。AWS Secrets Manager
に一時的な認証情報を保存します。必要なリソースにアクセスするには、この一時的な認証情報を使用します。
Answer: A
Comprehensive and Detailed Step-by-Step Explanation:
* Option A: Amazon Cognito Identity Pool with Unauthenticated Role
* Cognito identity pools can generate temporary AWS credentials for both authenticated and unauthenticated users.
* For guest users, Cognito assigns an unauthenticated role with limited permissions, ensuring secure access to only their resources.
* This is the most secure and efficient solution for managing AWS credentials dynamically without hardcoding or storing them.
* Why Other Options Are Incorrect:
* Option B: Hardcoding IAM credentials in the application is insecure and violates best practices.
* Option C and D: Temporary keys stored in KMS or Secrets Manager require additional implementation overhead and do not inherently manage user-specific access.
References:
Amazon Cognito Identity Pools

QUESTION NO: 15
開発者は、VPC 内のプライベート リソースへのネットワーク アクセスを必要とする AWS Lambda 関数を作成しています。
A.
Lambda関数をプライベートサブネット経由でVPCにアタッチします。プライベートリソースへのネットワークアクセスを許可するセキュリティグループを作成し、そのセキュリティグループをLambda関数に関連付けます。
B.
Lambda関数を設定して、トラフィックをVPN接続経由でルーティングします。プライベートリソースへのネットワークアクセスを許可するセキュリティグループを作成し、Lambda関数に関連付けます。
C.
Lambda関数のVPCエンドポイント接続を設定します。VPCエンドポイントがNATゲートウェイ経由でトラフィックをルーティングするように設定します。
D. プライベートリソース用のAWS
PrivateLinkエンドポイントを設定します。PrivateLinkエンドポイントを参照するようにLambda関数を設定します。
Answer: A
Comprehensive Detailed Step by Step Explanation with All AWS Developer References:
When you need to provide an AWS Lambda function access to private resources in a VPC, the most common and straightforward approach is to attach the Lambda function to a VPC via private subnets. Once the Lambda function is associated with the VPC, you need to configure appropriate security groups to control the access to the private resources.

* Lambda with VPC Access: Lambda functions can be attached to private subnets in a VPC, allowing them to access resources like RDS, EC2, or internal services within that VPC.
* Security Groups: A security group acts as a virtual firewall for the Lambda function, ensuring that it can access only the necessary resources and ports in the VPC.
* Alternatives:
* Option B involves routing traffic through a VPN, which adds unnecessary complexity and operational overhead compared to simply attaching the Lambda to the VPC.
* Option C requires configuring a VPC endpoint and a NAT gateway, which can be complex and costly.
* Option D refers to AWS PrivateLink, which is used to access services over private connections, but it's unnecessary in this scenario unless you need a cross-VPC connection.
:
Lambda functions in a VPC

QUESTION NO: 16
ある企業の開発者は、毎日指定された時間に同じAPI呼び出しを1回行う小規模なアプリケーションを作成する必要があります。同社はまだAWSクラウドにインフラストラクチャを整備していませんが、この機能をAWS上に実装したいと考えています。
これらの要件を最も運用効率よく満たすソリューションはどれですか?
A. Amazon Elastic Kubernetes Service (Amazon EKS) で実行される Kubernetes cron ジョブを使用します。
B. Amazon EC2 で実行される Amazon Linux crontab スケジュールジョブを使用します。
C. Amazon EventBridge のスケジュールされたイベントによって呼び出される AWS Lambda 関数を使用します。
D. AWS Batch ジョブキューに送信された AWS Batch ジョブを使用します。
Answer: C
The correct answer is C. Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event.
C). Use an AWS Lambda function that is invoked by an Amazon EventBridge scheduled event. This is correct.
AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling, and logging1. Amazon EventBridge is a serverless event bus service that enables you to connect your applications with data from a variety of sources2. EventBridge can create rules that run on a schedule, either at regular intervals or at specific times and dates, and invoke targets such as Lambda functions3. This solution meets the requirements of creating a small application that makes the same API call once each day at a designated time, without requiring any infrastructure in the AWS Cloud or any operational overhead.
A). Use a Kubernetes cron job that runs on Amazon Elastic Kubernetes Service (Amazon EKS). This is incorrect. Amazon EKS is a fully managed Kubernetes service that allows you to run containerized applications on AWS4. Kubernetes cron jobs are tasks that run periodically on a given schedule5. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated

time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EKS cluster, which would incur additional costs and complexity.
B). Use an Amazon Linux crontab scheduled job that runs on Amazon EC2. This is incorrect. Amazon EC2 is a web service that provides secure, resizable compute capacity in the cloud6. Crontab is a Linux utility that allows you to schedule commands or scripts to run automatically at a specified time or date7. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to provision and manage an EC2 instance, which would incur additional costs and complexity.
D). Use an AWS Batch job that is submitted to an AWS Batch job queue. This is incorrect. AWS Batch enables you to run batch computing workloads on the AWS Cloud8. Batch jobs are units of work that can be submitted to job queues, where they are executed in parallel or sequentially on compute environments9. This solution could meet the functional requirements of creating a small application that makes the same API call once each day at a designated time, but it would not be the most operationally efficient manner. The company would need to configure and manage an AWS Batch environment, which would incur additional costs and complexity.
References:
1: What is AWS Lambda? - AWS Lambda
2: What is Amazon EventBridge? - Amazon EventBridge
3: Creating an Amazon EventBridge rule that runs on a schedule - Amazon EventBridge
4: What is Amazon EKS? - Amazon EKS
5: CronJob - Kubernetes
6: What is Amazon EC2? - Amazon EC2
7: Crontab in Linux with 20 Useful Examples to Schedule Jobs - Tecmint
8: What is AWS Batch? - AWS Batch
9: Jobs - AWS Batch

QUESTION NO: 17
開発者はAWS
Lambda関数を作成しています。このLambda関数は、サードパーティのソリューションに接続するために外部ライブラリを必要とします。外部ライブラリは、合計サイズが100MBのファイルの集合です。開発者は、外部ライブラリをLambda実行環境で利用できるようにし、Lambdaパッケージの容量を削減する必要があります。これらの要件を満たし、運用オーバーヘッドが最も少ないソリューションはどれでしょうか？
A.
外部ライブラリを保存するためのLambdaレイヤーを作成する。そのレイヤーを使用するようにLambda関数を設定する。
B. Amazon S3 バケットを作成します。外部ライブラリを S3
バケットにアップロードします。Lambda 関数で S3
バケットフォルダをマウントします。マウントポイントの適切なフォルダを使用してライブラリをインポートします。
C.
Lambdaパッケージのデプロイ中に、外部ライブラリをLambda関数の/tmpディレクトリに口

ードします。/tmpディレクトリからライブラリをインポートします。
D. Amazon Elastic File System (Amazon EFS) ボリュームを作成します。外部ライブラリを EFS ボリュームにアップロードし、Lambda 関数で EFS
ボリュームをマウントします。マウントポイント内の適切なフォルダを使用してライブラリをインポートします。
Answer: A
* Lambda Layers: These are designed to package dependencies that you can share across functions.
* How to Use:
* Create a layer, upload your 100MB library as a zip.
* Attach the layer to your function.
* In your function code, import the library from the standard layer path.
References:
Lambda Layers: https://docs.aws.amazon.com/lambda/latest/dg/configuration-layers.html

QUESTION NO: 18
開発者は、Amazon CloudFront
を使用してコンテンツを配信するウェブサイトを管理しています。ウェブサイトの静的アーティファクトは、Amazon S3 バケットに保存されています。
開発者はいくつかの変更をデプロイし、S3バケットで新しいアーティファクトを確認できます。ただし、変更内容はCloudFrontディストリビューションが配信するウェブページには表示されません。
開発者はこの問題をどのように解決すべきでしょうか?
A. S3 オブジェクトが更新されるたびに、ファイルの最新バージョンに更新されるように S3 オブジェクト ロックを設定します。
B.
新しいアーティファクトがアップロードされる前に、バケットからすべての古いオブジェクトをクリアするように S3 バケットを設定します。
C. アーティファクトが Amazon S3 にデプロイされた後、キャッシュを無効にするように CloudFront を設定します。
D. アーティファクトが Amazon S3
にデプロイされた後、ディストリビューションオリジンを変更するように CloudFront
を設定します。
Answer: C

QUESTION NO: 19
Amazon EC2 インスタンスでホストされているアプリケーションは、Amazon S3
バケットに保存されているファイルにアクセスする必要があります。アプリケーションは、S3
バケットに保存されているオブジェクトを一覧表示し、ユーザーにテーブルを表示します。
テスト中に、開発者はアプリケーションのリストにオブジェクトがまったく表示されないことに気付きました。
この問題を解決する最も安全な方法は何ですか?
A. EC2 インスタンスにアタッチされている IAM インスタンス
プロファイルを更新して、S3 バケットの S3:* 権限を含めます。

B. EC2 インスタンスにアタッチされている IAM インスタンス プロファイルを更新して、S3 バケットの S3:ListBucket 権限を含めます。
C. 開発者のユーザー権限を更新して、S3 バケットの S3:ListBucket 権限を含めます。
D. S3:ListBucket 権限を追加し、Principal 要素を設定して EC2 インスタンスのアカウント番号を指定して、S3 バケット ポリシーを更新します。
Answer: B
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

QUESTION NO: 20
ある大企業では、アプリケーションコンポーネントが複数のAWSアカウントに分散しており、これらのアカウント間でトレースデータを収集・可視化する必要があります。これらの要件を満たすには何を使用すればよいでしょうか?
A. AWS X-Ray
B. Amazon CloudWatch
C. Amazon VPC フローログ
D. Amazon OpenSearch サービス
Answer: A

QUESTION NO: 21
開発者は、新しいデプロイによってエラーが発生した場合に、AWS Lambda関数を以前のバージョンにロールバックできる機能を望んでいます。ユーザーへの影響を最小限に抑えながら、これを実現するにはどうすればよいでしょうか？
A.
アプリケーションを変更し、現在のバージョンを指すエイリアスを使用します。新しいバージョンのコードをデプロイし、新しくデプロイしたバージョンを使用するようにエイリアスを更新します。エラーが多すぎる場合は、エイリアスを以前のバージョンに戻します。
B.
アプリケーションを変更し、現在のバージョンを指すエイリアスを使用します。新しいバージョンのコードをデプロイします。エイリアスを更新し、ユーザーの10%を新しくデプロイしたバージョンに誘導します。エラーが多すぎる場合は、トラフィックの100%を以前のバージョンに送信します。
C.
アプリケーションに変更を加えないでください。新しいバージョンのコードをデプロイしてください。エラーが多すぎる場合は、Amazon リソースネーム (ARN) のバージョン番号を使用して、アプリケーションを以前のバージョンに戻してください。
D.
新規、既存、ルーターの3つのエイリアスを作成します。既存のエイリアスを現在のバージ

ョンに関連付けます。ルーターエイリアスによって、すべてのユーザーが既存のエイリアスに誘導されるようにします。アプリケーションを更新して、ルーターエイリアスを使用するようにします。
新しいバージョンのコードをデプロイします。新しいエイリアスをこのバージョンに向けます。ルーターエイリアスを更新して、
ユーザーの10%を新しいエイリアスに振り分けます。エラーが多すぎる場合は、トラフィックの100%を既存のエイリアスに振り分けます。
Answer: A

QUESTION NO: 22
ある企業は、開発のスピードと俊敏性を向上させるために、AWS
サービスを活用したスケーラブルなデータ管理ソリューションを構築しています。このソリューションは、様々なソースから大量のデータを取り込み、複数のビジネスルールと変換処理を通してデータを処理します。
このソリューションでは、ビジネスルールを順番に実行し、ビジネスルール実行中にエラーが発生した場合にデータの再処理を実行する必要があります。このソリューションは拡張性が高く、メンテナンスの必要性が最小限であることが求められています。
これらの要件を満たすために、データフローのオーケストレーションを管理および自動化するには、どの AWS サービスを使用する必要がありますか?
A. AWS バッチ
B. AWS ステップ関数
C. AWS グルー
D. AWS Lambda
Answer: B
https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html

QUESTION NO: 23
ある企業は、ファームウェアのアップデートを世界中の顧客に配布する必要があります。
ダウンロードへのアクセスを最も低コストで簡単かつ安全に制御できるサービスはどれですか?
A. Amazon S3 の署名付き URL で Amazon CloudFront を使用します。
B. 顧客ごとに専用の Amazon CloudFront ディストリビューションを作成します。
C. AWS Lambda@Edge で Amazon CloudFront を使用します。
D. Amazon API Gateway と AWS Lambda を使用して、S3
バケットへのアクセスを制御します。
Answer: A
This solution allows easy and secure control of access to the downloads at the lowest cost because it uses a content delivery network (CDN) that can cache and distribute firmware updates to customers around the world, and uses a mechanism that can restrict access to specific files or versions. Amazon CloudFront is a CDN that can improve performance, availability, and security of web applications by delivering content from edge locations closer to customers. Amazon S3 is a storage service that can store firmware updates in buckets and objects. Signed URLs are URLs that include additional information, such as an expiration date and time, that give users temporary access to specific objects in S3 buckets. The

developer can use CloudFront to serve firmware updates from S3 buckets and use signed URLs to control who can download them and for how long.
Creating a dedicated CloudFront distribution for each customer will incur unnecessary costs and complexity.
Using Amazon CloudFront with AWS Lambda@Edge will require additional programming overhead to implement custom logic at the edge locations. Using Amazon API Gateway and AWS Lambda to control access to an S3 bucket will also require additional programming overhead and may not provide optimal performance or availability.
Reference: [Serving Private Content through CloudFront], [Using CloudFront with Amazon S3]

QUESTION NO: 24
開発者は、サーバーレスコンポーネントを使用して、高度に安全なヘルスケアアプリケーションを構築しています。このアプリケーションでは、AWS Lambda関数を使用して/Impストレージに一時データを書き込む必要があります。
開発者はこのデータをどのように暗号化すればよいでしょうか?
A. Lambda 関数の設定で AWS KMS キーを使用して Amazon EBS ボリュームの暗号化を有効にし、Lambda 関数に接続されているすべてのストレージが暗号化されるようにします。
B. AWS KMSキーにアクセスするためのロールとキーポリシーを使用してLambda関数を設定します。このキーを使用して、Ampストレージに書き込む前にすべてのデータを暗号化するためのデータキーを生成します。
C. Lambda の起動時に OpenSSL を使用して対称暗号化キーを生成します。このキーを使用して、/tmp に書き込む前にデータを暗号化します。
D. オンプレミスのハードウェア セキュリティ モジュール (HSM) を使用してキーを生成します。Lambda 関数は HSM からデータ キーを要求し、それを使用して関数へのすべてのリクエストのデータを暗号化します。
Answer: B

QUESTION NO: 25
開発者は、AWS Lambda関数を用いてデータを処理するアプリケーションを構築しています。このアプリケーションでは、レイテンシを最小限に抑える必要があります。Lambda関数の起動時間は予測可能でなければなりません。実行環境のすべてのセットアップアクティビティは、Lambda関数の呼び出し前に完了している必要があります。
これらの要件を満たすソリューションはどれでしょうか?
A. Lambda関数のメモリを最大量まで増やします。Amazon EventBridgeルールを設定して、Lambda関数の呼び出しを1分ごとにスケジュールし、実行環境をアクティブに保ちます。
B.
新しい実行環境を初めて準備する際に実行される静的初期化コードを最適化します。Lambda関数パッケージとインポートされたライブラリおよび依存関係のサイズを縮小・圧縮します。

C.
Lambda関数の予約済み同時実行数を、未予約アカウント同時実行数の最大値まで増加します。Lambda関数の最初の呼び出し前に、必要なセットアップアクティビティを手動で実行してください。
D.
Lambda関数の新しいバージョンを公開します。必要な最小数の実行環境で、Lambda関数のプロビジョニングされた同時実行性を設定します。
Answer: D

QUESTION NO: 26
ある企業には、AWS
Lambda関数で構成されたマイクロサービスが多数存在します。社内の複数のチームがマイクロサービスの所有権を分割しています。
アプリケーションは、Lambda関数に含まれる環境変数から設定値を読み取ります。セキュリティ監査中に、一部の環境変数に機密情報が含まれていることが判明しました。
同社のセキュリティポリシーでは、各チームがそれぞれのマイクロサービスに使用するAWS KMS キーのローテーションを完全に制御することが求められています。
A.
すべてのLambda関数にAWSマネージドキーを作成します。新しいAWSマネージドキーを使用して環境変数を暗号化します。Lambda関数実行ロールにkms:Decrypt権限を追加します。
B.
すべてのLambda関数にカスタマーマネージドキーを作成します。新しいカスタマーマネージドキーを使用して環境変数を暗号化します。Lambda関数実行ロールに kms:Decrypt 権限を追加します。
C.
すべてのLambda関数にカスタマーマネージドキーを作成します。新しいカスタマーマネージドキーを使用して環境変数を暗号化します。Lambda関数実行ロールに kms:CreateGrant 権限と kms:Encrypt 権限を追加します。
D.
すべてのLambda関数にAWSマネージドキーを作成します。新しいAWSマネージドキーを使用して環境変数を暗号化します。Lambda関数実行ロールにkms:CreateGrant権限とkms:Encrypt権限を追加します。
Answer: B
Comprehensive and Detailed Step-by-Step Explanation:
* Customer Managed Keys (CMK) for Granular Control (Option B):
* Customer-managed KMS keys are required to meet the security policy requirement of team- specific control over KMS key rotation. Each team can manage the lifecycle of its own key.
* The kms:Decrypt permission allows the Lambda function execution roles to decrypt the environment variables during runtime.
* This solution adheres to the principle of least privilege and satisfies the need for team-specific key control.
* Why Other Options Are Incorrect:
* Option A: AWS-managed keys cannot provide team-specific control or support the custom rotation policy required by the teams.

* Option C: Adding kms:CreateGrant and kms:Encrypt permissions to Lambda roles is unnecessary for this scenario. The key usage is limited to decryption at runtime.
* Option D: AWS-managed keys still lack team-specific control, and adding kms:CreateGrant and kms:Encrypt is redundant.
References:
AWS Lambda Environment Variables
AWS Key Management Service Documentation

QUESTION NO: 27
AWS Code Deploy
を使用したデプロイメントの場合、インプレースデプロイメントのフックの実行順序は何ですか?
A. BeforeInstall -> ApplicationStop -> ApplicationStart -> AfterInstall
B. アプリケーション停止 -> インストール前 -> インストール後 -> アプリケーション開始
C. BeforeInstall -> ApplicationStop -> ValidateService -> ApplicationStart
D. アプリケーション停止 -> インストール前 -> 検証サービス -> アプリケーション開始
Answer: B
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]

QUESTION NO: 28
ある企業では、AWS Elastic Beanstalk
上にデプロイされたアプリケーションを運用しています。このアプリケーションはユーザー固有の PDF を生成し、Amazon S3 バケットに保存します。その後、Amazon Simple Email Service (Amazon SES) を使用して、PDF を購読者にメールで送信します。
PDFが生成されてから90日後、ユーザーはPDFにアクセスできなくなります。S3バケットはバージョン管理されておらず、多くの古いPDFが含まれています。
開発者は、90 日以上経過した PDF を削除して、S3
バケット内のファイル数を減らす必要があります。
最も少ない開発労力でこの要件を満たすソリューションはどれでしょうか?
A.
アプリケーションコードを更新します。コードに、S3バケット内のすべてのオブジェクトを

毎日スキャンし、90日後にオブジェクトを削除するルールを追加します。
B. AWS
Lambda関数を作成します。S3バケット内のすべてのオブジェクトを毎日スキャンし、90日後にオブジェクトを削除するようにLambda関数をプログラムします。
C. S3 バケットの S3 ライフサイクル ルールを作成し、90
日後にオブジェクトを期限切れにします。
D.
S3オブジェクトを<年>/<月>/<日>キープレフィックスでパーティション分割します。有効期限に達したプレフィックスを持つオブジェクトを削除するAWS
Lambda関数を作成します。
Answer: C

QUESTION NO: 29
開発者は、複数のコンピューティング環境で実行されるマイクロサービス
アプリケーションを作成しています。
アプリケーションは、AWS Secrets Manager
に保存されているシークレットに、最小限のネットワークレイテンシーで安全にアクセスする必要があります。開発者は、Secrets Manager
への直接呼び出し回数を減らし、環境間のシークレット管理を簡素化するソリューションを求めています。これらの要件を満たし、運用オーバーヘッドが最も少ないソリューションはどれでしょうか？
A. Secrets Manager からシークレットを直接取得し、各コンピューティング環境のローカルデータベースにシークレットをキャッシュするカスタム スクリプトを作成します。
B. 各コンピューティング環境に Secrets Manager
エージェントをインストールします。エージェントがシークレットをローカルにキャッシュするように設定し、必要に応じて Secrets Manager からシークレットを安全に取得します。
C. アプリケーションに遅延読み込みロジックを実装して、Secrets Manager
からシークレットを直接取得し、Redis にシークレットをキャッシュします。
D. シークレットを Amazon S3
バケットに保存します。各コンピューティング環境のアプリケーション起動時に、シークレットを環境変数として取得して読み込みます。
Answer: B
The Secrets Manager Agent provides an out-of-the-box solution for securely caching secrets locally, reducing latency and operational overhead.
* Why Option B is Correct:
* Caching: The agent securely caches secrets locally, minimizing Secrets Manager API calls.
* Security: Secrets remain secure during retrieval and storage.
* Low Operational Overhead: Managed solution eliminates the need for custom logic.
* Why Not Other Options:
* Option A: Custom scripts introduce complexity and require ongoing maintenance.
* Option C: Using Redis requires managing an additional service, increasing overhead.
* Option D: Storing secrets in S3 lacks the fine-grained security controls of Secrets Manager.
:
Caching Secrets in AWS Secrets Manager

QUESTION NO: 30
アプリケーションはAmazon
Kinesisデータストリームからデータを取り込みます。データストリーム内のシャードは通常のトラフィック用に設定されています。
ピークトラフィックのテスト中は、アプリケーションはゆっくりとデータを取り込みます。
開発者は、ピークトラフィックに対応できるようにデータストリームを調整する必要があります。
この要件を最もコスト効率よく満たすために、開発者は何をすべきでしょうか?
A. データ ストリームにデータを組み込むには、Kinesis Producer Library (KPL) をインストールします。
B.
データストリームをオンデマンド容量モードに切り替えます。データストリームにデータを書き込む際にパーティションキーを指定します。
C. DecreaseStreamRetention Period API 操作を使用して、データ ストリームにデータが保持される時間を短縮します。
D. UpdateShardCount API 操作を使用して、データ ストリーム内のシャード数を増やします。
Answer: D

QUESTION NO: 31
あるeコマースアプリケーションがApplication Load Balancerの背後で実行されています。開発者は、ピーク時以外でアプリケーションに予期せぬ負荷が発生していることに気づきました。開発者は、アプリケーションを使用するクライアントIPアドレスのパターンを分析したいと考えています。この分析にはどのHTTPヘッダーを使用すべきでしょうか？
A. X-Forwarded-Protoヘッダー
B. XF Forwarded-Host ヘッダー
C. X-Forwarded-For ヘッダー
D. X-Forwarded-Port ヘッダー
Answer: C
The HTTP header that the developer should use for this analysis is the X-Forwarded-For header. This header contains the IP address of the client that made the request to the Application Load Balancer. The developer can use this header to analyze patterns for the client IP addresses that use the application. The other headers either contain information about the protocol, host, or port of the request, which are not relevant for the analysis.
Reference: How Application Load Balancer works with your applications

QUESTION NO: 32
開発者は、複数のサードパーティの支払い処理 API と通信する電子商取引プラットフォームに取り組んでいます。サードパーティの支払いサービスでは、テスト環境が提供されていません。
開発者は、eコマースプラットフォームとサードパーティの決済処理APIの統合を検証する必要があります。開発者は、サードパーティの決済処理APIを呼び出さずに、API統合コードをテストする必要があります。

これらの要件を満たすソリューションはどれでしょうか?
A. ステータスコード 200 に設定されたゲートウェイ応答を使用して Amazon API Gateway REST API を設定し、実際のサードパーティ API
からキャプチャされたサンプル応答を含む応答テンプレートを追加します。
B. 各サードパーティ API 用に設定されたデータソースを使用して AWS AppSync GraphQL API をセットアップし、統合タイプとして Mock を指定します。実際のサードパーティ API からキャプチャされたサンプル応答を使用して、統合応答を構成します。
C. サードパーティAPIごとにAWS
Lambda関数を作成します。実際のサードパーティAPIからキャプチャしたレスポンスを埋め込みます。各Lambda関数のAmazonリソースネーム（ARN）に対応するインバウンドエンドポイントをAmazon Route 53 Resolverに設定します。
D. 各サードパーティ API に対して Amazon API Gateway REST API
を設定する統合リクエストタイプとして Mock を指定する実際のサードパーティ API からキャプチャしたサンプルレスポンスを使用して統合レスポンスを構成する
Answer: D
* Mocking API Responses: API Gateway's Mock integration type enables simulating API behavior without invoking backend services.
* Testing with Sample Data: Using captured responses from the real third-party API ensures realistic testing of the integration code.
* Focus on Integration Logic: This solution allows the developer to isolate and test the application's interaction with the payment APIs, even without a test environment from the third-party providers.
References:
Amazon API Gateway Mock Integrations:
https://docs.aws.amazon.com/apigateway/latest/developerguide
/how-to-mock-integration.html

QUESTION NO: 33
ある金融会社は、法的理由により、顧客の原本記録を10年間保管する必要があります。完全な記録には、個人を特定できる情報（PII）が含まれます。現地の規制により、PIIは社内の特定の担当者のみがアクセスでき、第三者と共有することはできません。会社は、PIIを共有することなく、統計分析のために記録を第三者機関に提供する必要があります。
開発者は、元の不変レコードをAmazon
S3に保存したいと考えています。S3ドキュメントにアクセスするユーザーに応じて、ドキュメントをそのまま返すか、すべてのPIIを削除して返す必要があります。開発者は、ドキュメントからPIIを削除するAWS
Lambda関数を作成しました。この関数の名前はremovePiiです。
企業がドキュメントのコピーを 1 つだけ保持しながら PII
要件を満たすためには、開発者は何をすべきでしょうか?
A. S3 GET リクエストが行われたときに removePii 関数を呼び出す S3
イベント通知を設定します。
PII なしでオブジェクトにアクセスするには、GET リクエストを使用して Amazon S3 を呼び出します。
B. S3 PUT リクエストが行われたときに removePii 関数を呼び出す S3
イベント通知を設定します。

PUT リクエストを使用して Amazon S3 を呼び出し、PII
なしでオブジェクトにアクセスします。
C.
S3コンソールからS3オブジェクトLambdaアクセスポイントを作成します。removePii関数を選択します。S3アクセスポイントを使用して、PIIなしでオブジェクトにアクセスします。
D.
S3コンソールからS3アクセスポイントを作成します。アクセスポイント名を使用して、Get ObjectLegalHold S3
API関数を呼び出します。PIIなしでオブジェクトにアクセスするには、removePii関数名を渡します。
Answer: C
S3 Object Lambda allows you to add your own code to process data retrieved from S3 before returning it to an application. You can use an AWS Lambda function to modify the data, such as removing PII, redacting confidential information, or resizing images. You can create an S3 Object Lambda access point and associate it with your Lambda function. Then, you can use the access point to request objects from S3 and get the modified data back. This way, you can maintain only one copy of the original document in S3 and apply different transformations depending on who accesses it. Reference: Using AWS Lambda with Amazon S3

QUESTION NO: 34
開発者は、IAM ユーザーを含むテンプレートからアプリケーション用の AWS CloudFormation スタックをデプロイする準備をしています。
開発者は、IAM
ユーザーの作成が成功した後、そのユーザーを保持するようにアプリケーションのリソースを構成する必要があります。
ただし、開発者は、スタックがロールバックした場合に IAM
ユーザーを削除するようにアプリケーションを構成する必要もあります。
A. 次の削除ポリシーを使用して CloudFormation テンプレートを更新します。
AWSテンプレートフォーマットバージョン: '2010-05-09'
リソース：
アプリユーザー:
タイプ: AWS::IAM::User
削除ポリシー: 保持
B. 次の削除ポリシーを使用して CloudFormation テンプレートを更新します。
AWSテンプレートフォーマットバージョン: '2010-09-09'
リソース：
アプリユーザー:
タイプ: AWS::IAM::User
削除ポリシー: RetainExceptOnCreate
C. CloudFormation サービスロールを更新して、次のポリシーを追加します。
{
"Version": "2012-10-17",
"Statement": [{
"Effect": "Allow",

"Action": ["cloudformation:UpdateTerminationProtection"],
"Resource": "*"
}]
}
D. Update the stack policy to include the following statements:
{
"Statement": [{
"Effect": "Deny",
"Action": "Update:*",
"Principal": "*",
"Resource": "*",
"Condition": {
"StringEquals": {
"ResourceType": "AWS::IAM::User"
}
}
}]
}
Answer: B
* Why Option B is Correct:The RetainExceptOnCreate deletion policy ensures that the IAM user is retained after successful stack creation but is deleted if the stack creation fails or rolls back. This meets both requirements.
* Why Other Options are Incorrect:
* Option A: The Retain policy retains the resource regardless of stack status and does not delete the IAM user upon rollback.
* Option C: Updating the service role for termination protection does not address the specific deletion behavior for the IAM user.
* Option D: Stack policy controls updates, not resource deletion behavior during rollbacks.
* AWS Documentation References:
* CloudFormation DeletionPolicy Attribute

QUESTION NO: 35
あるオンライン食品会社は、パートナーからの注文受付用にAmazon API Gateway HTTP APIを提供しています。このAPIはAWS Lambda関数と統合されており、Lambda関数は注文をAmazon DynamoDBテーブルに保存します。
同社は今後、追加のパートナーをオンボードする予定です。一部のパンサーは注文を受け取るために追加のLambda関数を必要とします。同社はAmazon S3バケットを作成しました。将来の分析のために、すべての注文と更新をS3バケットに保存する必要があります。開発者は、最小限の開発労力で注文と更新をAmazon S3に保存するにはどうすればよいでしょうか？
A. 新しいLambda関数と新しいAPI Gateway APIエンドポイントを作成します。新しいLambda関数はS3バケットに書き込むように設定します。元のLambda関数を変更し、新しいAPIエンドポイントに更新を送信します。
B. Amazon Kinesis Data Streams

を使用して新しいデータストリームを作成します。Lambda 関数を変更して、注文を oats ストリームにパブリッシュし、データストリームで S3 バケットに書き込むように設定します。

C. DynamoOB テーブルで DynamoDB ストリームを有効にします。新しい Lambda 関数を作成します。ストリームの Amazon リソースネーム (ARN) を Lambda 関数に関連付けます。テーブルのストリームにレコードが表示されたら S3 バケットに書き込むように Lambda 関数を設定します。

D.

Lambda関数を変更して、新しいAmazonにペナルティを課します。シンプルなLambda関数で注文を受け取ります。新しいLambda関数をトピックにサブスクライブします。トピックを通じて更新が届いたときにS3/バケットに書き込むように、新しいLambda関数を設定します。

Answer: C

This solution will ensure that all orders and updates are stored to Amazon S3 with the least development effort because it uses DynamoDB Streams to capture changes in the DynamoDB table and trigger a Lambda function to write those changes to the S3 bucket. This way, the original Lambda function and API Gateway API endpoint do not need to be modified, and no additional services are required. Option A is not optimal because it will require more development effort to create a new Lambda function and a new API Gateway API endpoint, and to modify the original Lambda function to post updates to the new API endpoint. Option B is not optimal because it will introduce additional costs and complexity to use Amazon Kinesis Data Streams to create a new data stream, and to modify the Lambda function to publish orders to the data stream. Option D is not optimal because it will require more development effort to modify the Lambda function to publish to a new Amazon SNS topic, and to create and subscribe a new Lambda function to the topic.

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