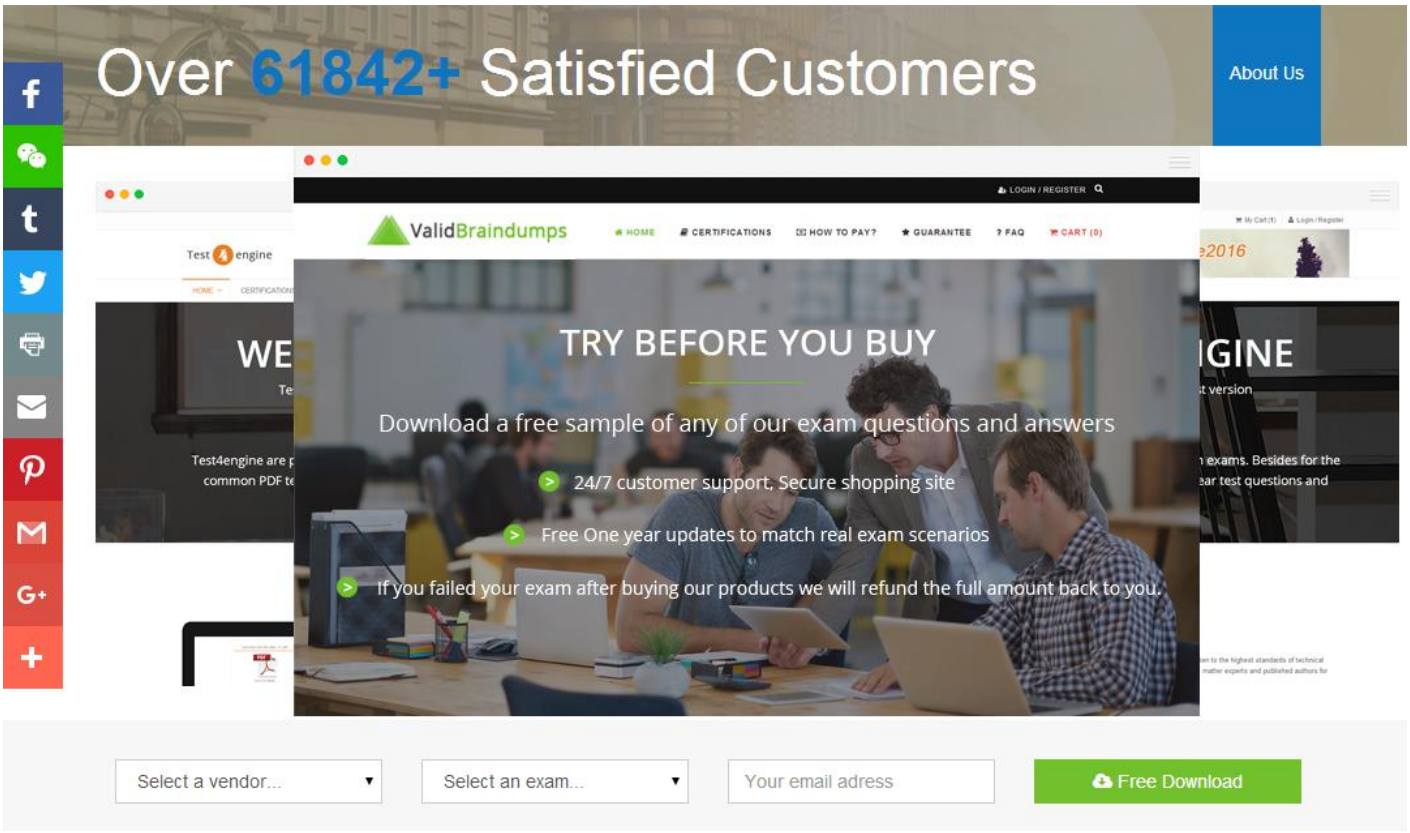
ValidBraindumps



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.

http://www.validbraindumps.com

Free valid test braindumps for IT certification valid exam

Exam : **DOP-C02-JPN**

Title : AWS Certified DevOps Engineer - Professional (DOP-C02日本語版)

Vendor : Amazon

Version : DEMO

QUESTION NO: 1
企業には 20 のサービスが学習されます。各サービス
チームは独自のマイクロサービスを担当します。各サービス
チームは、マイクロサービス用に個別の AWS アカウントと、192 168 0 0/22 CIDR
ブロックを持つ VPC を使用します。同社は AWS Organizations で AWS
アカウントを管理しています。
各サービス チームは、Application Load Balancer の背後にある複数の Amazon EC2
インスタンスでマイクロサービスをホストします。マイクロサービスはパブリック
インターネットを介して相互に通信します。同社のセキュリティチームは、マイクロサービス間のすべての通信にはプライベートネットワーク接続を介したHTTPSを使用する必要があり、パブリックインターネットを通過することはできないという新しいガイドラインを発行した。
DevOps エンジニアは、これらの義務を果たし、各サービス
チームの変更数を最小限に抑えるソリューションを実装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すか?
A. AWS Organizations で新しい AWS アカウントを作成します。 このアカウントに VPC を作成し、AWS Resource Access Manager を使用して、この VPC のプライベート
サブネットを組織と共有します。 サービス
チームに新しいアカウントを起動するように指示します。共有プライベート
サブネットを使用するネットワーク ロード バランサー (NLB) と EC2 インスタンス
マイクロサービス間の通信には NLB DNS 名を使用します。
B. 各マイクロサービス VPC にネットワーク ロード バランサー (NLB) を作成します。 AWS PrivateLink を使用して、NLB の各 AWS アカウントに VPC エンドポイントを作成します。他の各 AWS アカウントに各 VPC エンドポイントへのサブスクリプションを作成します。VPC エンドポイント DNS を使用します。マイクロサービス間の通信の名前。
C. 各マイクロサービス VPC にネットワーク ロード バランサー (NLB) を作成します。
各マイクロサービス VPC 間に VPC ピアリング接続を作成します。 ピアリング
リンクを使用するように各 VPC のルート テーブルを更新します。
マイクロサービス間の通信に NLB DNS 名を使用します。
D. AWS Organizations で新しい AWS
アカウントを作成します。このアカウントにトランジット ゲートウェイを作成し、AWS Resource Access Manager を使用してトランジット
ゲートウェイを組織と共有します。各マイクロサービス VPC 内。共有トランジット
ゲートウェイへのトランジット ゲートウェイ アタッチメントを作成する トランジット
ゲートウェイを使用するように各 VPC のルート テーブルを更新する 各マイクロサービス VPC にネットワーク ロード バランサー (NLB) を作成する マイクロサービス間の通信に NLB DNS 名を使用する
Answer: B
Explanation:
https://aws.amazon.com/blogs/networking-and-content-delivery/connecting-networks-with-overlapping-ip- ranges/ Private link is the best option because Transit Gateway doesn ' t support overlapping CIDR ranges.

QUESTION NO: 2
ある企業のアプリケーションチームは、アプリケーションにAWS

CodeCommitリポジトリを使用しています。アプリケーションチームは複数のAWSアカウントにリポジトリを保有しており、すべてのアカウントはAWS Organizations内の同一組織に属しています。
各アプリケーションチームは、外部IDプロバイダー(IdP)で構成されたAWS IAM Identity Center(AWSシングルサインオン)を使用して、開発者IAMロールを引き受けます。開発者ロールにより、アプリケーションチームはGitを使用してリポジトリ内のコードを操作できます。
セキュリティ監査の結果、アプリケーションチームはどのリポジトリのメインブランチでも変更できることが判明しました。DevOpsエンジニアは、アプリケーションチームが管理するリポジトリのメインブランチのみを変更できるようにするソリューションを実装する必要があります。
これらの要件を満たす手順の組み合わせはどれですか？(3つ選択してください。)

A.
SAMLアサーションを更新して、ユーザーのチーム名を渡すようにします。IAMロールの信頼ポリシーを更新して、チーム名を含むaccess-teamセッションタグを追加します。

B.
組織の管理アカウントで、各チーム用の承認ルールテンプレートを作成します。テンプレートをすべてのリポジトリに関連付けます。開発者ロールのARNを承認者として追加します。

C.
各アカウントの承認ルールテンプレートを作成します。テンプレートをすべてのリポジトリに関連付けます。
" aws:ResourceTag/access-team " : " $;{aws:PrincipalTag/access-team} "
条件を承認ルールテンプレートに追加します。

D.
CodeCommitリポジトリごとに、関連付けられたチームの名前を値として設定したaccess-teamタグを追加します。

E. アカウントに SCP を添付します。以下の記述を含めてください。

F. 各アカウントに IAM 権限境界を作成します。次のステートメントを含めます。

Answer: A D F

Explanation:
Short Explanation: To meet the requirements, the DevOps engineer should update the SAML assertion to pass the user's team name, update the IAM role's trust policy to add an access-team session tag that has the team name, create an IAM permissions boundary in each account, and for each CodeCommit repository, add an access-team tag that has the value set to the name of the associated team.
Updating the SAML assertion to pass the user's team name allows the DevOps engineer to use IAM tags to identify which team a user belongs to. This can help enforce fine-grained access control based on the user's team membership1.
Updating the IAM role's trust policy to add an access-team session tag that has the team name allows the DevOps engineer to use IAM condition keys to restrict access based on the session tag value2. For example, the DevOps engineer can use the aws:PrincipalTag condition key to match the access-team tag of the user with the access-team tag of the repository3.
Creating an IAM permissions boundary in each account allows the DevOps engineer to set the maximum permissions that an identity-based policy can grant to an IAM entity. An entity's

permissions boundary allows it to perform only the actions that are allowed by both its identity-based policies and its permissions boundaries4. For example, the DevOps engineer can use a permissions boundary policy to limit the actions that a user can perform on CodeCommit repositories based on their access-team tag5.
For each CodeCommit repository, adding an access-team tag that has the value set to the name of the associated team allows the DevOps engineer to use resource tags to identify which team manages a repository. This can help enforce fine-grained access control based on the resource tag value6.
The other options are incorrect because:
Creating an approval rule template for each team in the Organizations management account is not a valid option, as approval rule templates are not supported by AWS Organizations. Approval rule templates are specific to CodeCommit and can only be associated with one or more repositories in the same AWS Region where they are created7.
Creating an approval rule template for each account is not a valid option, as approval rule templates are not designed to restrict access to modify branches. Approval rule templates are designed to require approvals from specified users or groups before merging pull requests8.
Attaching an SCP to the accounts is not a valid option, as SCPs are not designed to restrict access based on tags. SCPs are designed to restrict access based on service actions and resources across all users and roles in an organization's account9.

QUESTION NO: 3
企業はアプリケーションにフェイルオーバーを実装する必要があります。このアプリケーションには、Amazon CloudFront ディストリビューションと AWS リージョンのパブリック Application Load Balancer (ALB) が含まれています。同社は、ALB をディストリビューションのデフォルトのオリジンとして構成しました。
最近アプリケーションが停止した後、同社は 0 秒の RTO を望んでいます。会社は、アプリケーションをウォーム スタンバイ構成のセカンダリ リージョンにデプロイします。DevOps エンジニアは、HTTP GET リクエストが必要な RTO を満たすように、セカンダリ リージョンへのアプリケーションのフェールオーバーを自動化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すか?
A. セカンダリ ALB をデフォルトのオリジンとする 2 番目の CloudFront ディストリビューションを作成します。フェイルオーバーポリシーを持つ Amazon Route 53 エイリアス レコードを作成し、両方の CloudFront ディストリビューションに対して [ターゲットの健全性の評価] を [はい] に設定します。新しいレコード セットを使用するようにアプリケーションを更新します。
B. セカンダリ ALB のディストリビューション上に新しいオリジンを作成します。新しい起点グループを作成します。元の ALB をプライマリ原点として設定します。HTTP 5xx ステータス コードに対してフェイルオーバーするようにオリジン グループを構成します。元のグループを使用するようにデフォルトの動作を更新します。
C. フェイルオーバーポリシーを持つ Amazon Route 53 エイリアス レコードを作成し、両方の ALB で [ターゲットの健全性の評価] を [はい] に設定します。両方のレコードの TTL を 0 に設定します。新しいレコード

セットを使用するようにディストリビューションのオリジンを更新します。
D. HTTP 5xx ステータス コードを検出する CloudFront 関数を作成します。関数が 5xx ステータス コードを検出した場合に、セカンダリ ALB に 307 一時リダイレクト エラー応答を返すように関数を構成します。オリジン応答を関数に送信するようにディストリビューションのデフォルト動作を更新します。

Answer: B

Explanation:
The best solution to implement failover for the application is to use CloudFront origin groups. Origin groups allow CloudFront to automatically switch to a secondary origin when the primary origin is unavailable or returns specific HTTP status codes that indicate a failure1. This way, CloudFront can serve the requests from the secondary ALB in the secondary Region without any delay or redirection. To set up origin groups, the DevOps engineer needs to create a new origin on the distribution for the secondary ALB, create a new origin group with the original ALB as the primary origin and the secondary ALB as the secondary origin, and configure the origin group to fail over for HTTP 5xx status codes. Then, the DevOps engineer needs to update the default behavior to use the origin group instead of the single origin2.
The other options are not as effective or efficient as the solution in option B. Option A is not suitable because creating a second CloudFront distribution will increase the complexity and cost of the application. Moreover, using Route 53 alias records with a failover policy will introduce some delay in detecting and switching to the secondary CloudFront distribution, which may not meet the zero-second RTO requirement. Option C is not feasible because CloudFront does not support using Route 53 alias records as origins3. Option D is not advisable because using a CloudFront function to redirect the requests to the secondary ALB will add an extra round-trip and latency to the failover process, which may also not meet the zero-second RTO requirement.
1: Optimizing high availability with CloudFront origin failover - Amazon CloudFront
2: Creating an origin group - Amazon CloudFront
3: Values That You Specify When You Create or Update a Web Distribution - Amazon CloudFront

QUESTION NO: 4
DevOps エンジニアは、AWS Cloud Formation カスタム リソースを使用して AD コネクタをセットアップしました。AWS Lambda 関数が実行され、AD コネクタが作成されましたが、Cloud Formation は CREATE_IN_PROGRESS から CREATE_COMPLETE に移行していません。
この問題を解決するにはエンジニアはどのようなアクションをとるべきですか?
A. Lambda 関数コードが正常に終了したことを確認します。
B. Lambda 関数コードが事前署名された URL に応答を返すようにします。
C. Lambda 関数の IAM ロールに、スタック ARN に対する Cloudformation UpdateStack 権限があることを確認します。
D. Lambda 関数の IAM ロールに AWS アカウントに対する ds ConnectDirectory 権限があることを確認します。

Answer: B

Reference: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/crpg-ref-

responses.html

QUESTION NO: 5
ある企業は、自社アプリケーションの環境を複数のAWSアカウントで管理しています。各環境アカウントは、AWS Organizations内の異なるOUに属しています。
DevOpsチームは、複数の環境にわたるアプリケーションのデプロイプロセスを担当します。デプロイプロセスでは、共有サービスアカウント内のAWS CodePipelineパイプラインを使用します。DevOpsチームのメンバーは同じユーザーグループに属しています。チームメンバーは、AWS IAM Identity Centerを通じてすべてのアカウントへの管理者アクセス権を持っています。
開発環境における最近のデプロイ問題により、DevOpsチームは手動での手順を実行する必要に迫られた。その後、本番環境へのデプロイ中にインシデントが発生し、パイプラインが失敗して、数時間にわたり新規デプロイがブロックされた。
DevOpsエンジニアは、パイプラインのみが本番環境へのデプロイを実行できるようにする必要があります。また、緊急時にはDevOpsエンジニアが本番環境にアクセスできる必要があります。
これらの要件を最も効率的に満たすソリューションはどれでしょうか？
A.
本番環境のOU上でDevOpsチームメンバーによるすべての書き込み操作を拒否するSCPを作成します。CodePipelineがプロビジョニングするリソースに特定のタグを付けます。DevOpsエンジニア以外のエンティティによるタグ付けされたリソースの変更を拒否するSCPを追加します。
B. DevOps グループを更新して、本番アカウントに ReadOnlyAccess 権限を設定します。DevOps エンジニア ユーザーに、AdministratorAccess 権限を持ち、パイプライン ロールを引き受けることができる新しい権限セットを設定します。パイプライン ロール以外のエンティティによるリソースの変更を拒否する SCP を追加します。
C. DevOps グループを更新して、本番アカウントのパイプライン ロールを引き受けられるようにします。IAM Identity Center で、DevOps エンジニア用に AdministratorAccess 権限を持つ新しい権限セットを持つ新しいユーザーを設定します。DevOps エンジニア以外のエンティティによるリソースの変更を拒否する SCP を追加します。
D.
本番環境OU上のDevOpsチームメンバーのすべての書き込み操作を拒否するSCPを作成します。
IAM Identity CenterでDevOpsエンジニア用の新しいユーザーを設定し、AdministratorAccess権限を持つ新しい権限セットを付与します。パイプラインロール以外のエンティティによるリソースの変更を拒否するSCPを追加します。
Answer: B

QUESTION NO: 6
あるDevOpsエンジニアがAmazon Linux上でホストされているプロジェクトに取り組んでいるが、セキュリティレビューに不合格となった。
DevOpsマネージャーは、AWS

CodeBuildプロジェクトの会社のbuildspec.yamlファイルをレビューし、推奨事項を提供するよう依頼されました。buildspec.yamlファイルは次のように設定されています。
AWSのセキュリティに関するベストプラクティスに準拠するために、どのような変更を推奨すべきでしょうか？(3つ選択してください。)
A. db_password を AWS Systems Manager Parameter Store に SecureString 値として保存し、その後環境変数から db_password を削除します。
B. 環境変数を「db.-deploy-bucket」Amazon S3 バケットに移動し、変数をダウンロードしてからエクスポートする事前ビルドステージを追加します。
C. 終了前にコンテナから一時ファイルを削除するビルド後コマンドを追加し、他の CodeBuild ユーザーから一時ファイルが見えないようにします。
D. インスタンスに直接secコマンドやsshコマンドを実行するのではなく、AWS Systems Managerのrunコマンドを使用します。
E. CodeBuild プロジェクトロールに必要な権限を追加し、環境変数から AWS 認証情報を削除します。
Answer: A,D,E

QUESTION NO: 7
DevOps エンジニアには、コンテナ イメージを構築して ECR に保存し、イメージの脆弱性をスキャンし、アップストリーム ソース イメージ リポジトリの停止に対して耐性のある、回復力のある CI/CD パイプラインが必要です。
どのソリューションがこれを満たしていますか?
A. プライベート ECR リポジトリを作成し、プッシュ時にイメージをスキャンし、レプリケーション ルールを使用してアップストリーム リポジトリからイメージを複製します。
B. アップストリーム リポジトリからイメージをキャッシュするためのパブリック ECR リポジトリを作成し、イメージを保存するためのプライベート リポジトリを作成し、プッシュ時にイメージをスキャンします。
C. パブリック ECR リポジトリを作成し、プルスルー キャッシュ ルールを構成し、イメージを保存するためのプライベート リポジトリを作成し、基本的なスキャンを有効にします。
D. プライベート ECR リポジトリを作成し、基本スキャンを有効にし、プルスルー キャッシュ ルールを作成します。
Answer: D
* ECR pull-through cache caches images from upstream repositories for resilience.
* Private repo with basic scanning ensures vulnerability detection on pushed images.
* Enabling pull-through caching on a private repo combines caching and vulnerability scanning seamlessly.
* Public repos do not support pull-through caching of upstream images.
* Replication rules are for multi-Region replication, not upstream caching.
References:
Amazon ECR Pull Through Cache
Amazon ECR Image Scanning

QUESTION NO: 8
企業は、e コマース Web
アプリケーションを通じて製品を販売しています。この会社は、製品取引の詳細を円グラフで表示するダッシュボードを必要としています。会社は、ダッシュボードを会社の既存の Amazon CloudWatch
ダッシュボードと統合したいと考えています。これらの要件を最も高い運用効率で満たすソリューションはどれですか?
A. e コマース アプリケーションを更新して、処理されたトランザクションごとに JSON オブジェクトを CloudWatch ログ グループに出力します。CloudWatch Logs Insights を使用してログ グループをクエリし、結果を円グラフ形式で視覚化します。結果を目的の CloudWatch ダッシュボードに添付します。
B. e コマース アプリケーションを更新して、処理されたトランザクションごとに JSON オブジェクトを Amazon S3 バケットに出力します。Amazon Athena を使用して S3 バケットにクエリを実行し、結果を円グラフ形式で視覚化します。Athena から結果をエクスポートします。 結果を目的の CloudWatch ダッシュボードに添付します。
C. インストルメンテーションに AWS X-Ray を使用するように e コマース アプリケーションを更新します。新しい X-Ray サブセグメントを作成します。処理されたトランザクションごとに注釈を追加します。X 線トレースを使用してデータをクエリし、結果を円グラフ形式で視覚化します。結果を目的の CloudWatch ダッシュボードに添付します。
D. 処理されたトランザクションごとに JSON オブジェクトを CloudWatch ログ グループに出力するように e コマース アプリケーションを更新します。結果を集計して Amazon DynamoDB に書き込む AWS Lambda 関数を作成します。ログファイル用の Lambda サブスクリプションフィルターを作成します。結果を目的の CloudWatch ダッシュボードに添付します。
Answer: A
Explanation:
The correct answer is A.
Option A is correct because it meets the requirements with the most operational efficiency. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard. Using CloudWatch Logs Insights to query the log group and to visualize the results in a pie chart format is also a convenient and integrated solution that leverages the existing CloudWatch dashboards. Attaching the results to the desired CloudWatch dashboard is straightforward and does not require any additional steps or services.
Option B is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to an Amazon S3 bucket for each processed transaction is a valid way to store the data, but it requires creating and managing an S3 bucket and its permissions. Using Amazon Athena to query the S3 bucket and to visualize the results in a pie chart format is also a valid way to analyze the data, but it incurs charges based on the amount of data scanned by each query. Exporting the results from Athena and attaching them to the desired CloudWatch dashboard is also an extra step that adds more overhead and latency.
Option C is incorrect because it uses AWS X-Ray for an inappropriate purpose. Updating the

ecommerce application to use AWS X-Ray for instrumentation is a good practice for monitoring and tracing distributed applications, but it is not designed for aggregating product transaction details. Creating a new X-Ray subsegment and adding an annotation for each processed transaction is possible, but it would clutter the X- Ray service map and make it harder to debug performance issues. Using X-Ray traces to query the data and to visualize the results in a pie chart format is also possible, but it would require custom code and logic that are not supported by X-Ray natively. Attaching the results to the desired CloudWatch dashboard is also not supported by X-Ray directly, and would require additional steps or services.
Option D is incorrect because it introduces unnecessary complexity and cost. Updating the ecommerce application to emit a JSON object to a CloudWatch log group for each processed transaction is a simple and cost-effective way to collect the data needed for the dashboard, as in option A. However, creating an AWS Lambda function to aggregate and write the results to Amazon DynamoDB is redundant, as CloudWatch Logs Insights can already perform aggregation queries on log data. Creating a Lambda subscription filter for the log file is also redundant, as CloudWatch Logs Insights can already access log data directly. Attaching the results to the desired CloudWatch dashboard would also require additional steps or services, as DynamoDB does not support native integration with CloudWatch dashboards.
References:
CloudWatch Logs Insights
Amazon Athena
AWS X-Ray
AWS Lambda
Amazon DynamoDB

QUESTION NO: 9
ある企業は、AWS CDKとCodePipeline、CodeBuildを使用してアプリケーションをデプロイしています。デプロイ前にユニットテストを実施し、テストに合格した場合にのみデプロイを進めたいと考えています。
これを実施する手順はどれですか? (2 つ選択してください)。
A. CodeBuild ビルド コマンドを更新してテストを実行し、デプロイし、OnFailure を ABORT に設定します。
B. CodeBuild コマンドを更新してテストを実行してからデプロイし、cdk deploy に --rollback true を追加します。
C. CodeBuild コマンドを更新してテストを実行し、デプロイし、--require-approval any-change フラグを追加します。
D. template.hasResourceProperties アサーションを使用して、AWS CDK アサーション モジュールでテストを作成します。
E. cdk diff を使用し、リソースの変更が検出された場合は失敗するテストを作成します。
Answer: A D
* Running unit tests in the build phase and aborting on failure (OnFailure=ABORT) prevents deployment if tests fail.
* AWS CDK assertions module provides programmatic unit tests against synthesized

templates (Option D).
* The --rollback flag relates to CloudFormation stack rollback, not test gating.
* The --require-approval flag controls manual approvals, not test outcomes.
* cdk diff checks for changes but is not a unit test and may not catch logical errors.
References:
Testing AWS CDK Applications
CodeBuild Buildspec OnFailure

QUESTION NO: 10
DevOpsエンジニアは、AWS Fargate上のAmazon Elastic Container Service（Amazon ECS）クラスターで実行されるJavaベースのアプリケーションを管理しています。このアプリケーションには自動スケーリングが設定されていません。DevOpsエンジニアは、Java仮想マシン（JVM）のスレッド数が、アプリケーションのスケーリングタイミングを判断する上で優れた指標であると判断しました。このアプリケーションは、ポート8080で顧客トラフィックを処理し、ポート9404でJVMメトリクスを提供しています。最近、アプリケーションの利用が増加しています。DevOpsエンジニアは、このアプリケーションの自動スケーリングを設定する必要があります。これらの要件を、運用オーバーヘッドが最も少ないソリューションで満たせるでしょうか？

A. Amazon CloudWatch エージェントをコンテナサイドカーとしてデプロイします。CloudWatch エージェントを設定して、ポート 9404 から JVM メトリクスを取得します。JVM スレッド数メトリクスに CloudWatch アラームを作成し、アプリケーションをスケールします。Fargate にステップスケーリングポリシーを追加し、CloudWatch アラームに基づいてスケールアップおよびスケールダウンします。

B. Amazon CloudWatch エージェントをコンテナサイドカーとしてデプロイします。CloudWatch エージェントの CloudWatch ロググループに、JVM スレッド数メトリクスのメトリクスフィルターを設定します。Fargate にターゲット追跡ポリシーを追加します。メトリクスフィルターから、スケールターゲットとしてメトリクスを選択します。

C. Amazon Managed Service for Prometheus ワークスペースを作成します。AWS Distro for OpenTelemetry をコンテナサイドカーとしてデプロイし、ポート 9404 から JVM メトリクスを Prometheus ワークスペースに公開します。ワークスペースのルールを設定し、JVM スレッド数メトリクスを使用してアプリケーションをスケーリングします。Fargate にステップスケーリングポリシーを追加します。スケールアップとスケールダウンに使用する Prometheus ルールを選択します。

D. Amazon Managed Service for Prometheus ワークスペースを作成します。AWS Distro for OpenTelemetry をコンテナサイドカーとしてデプロイし、ポート 9404 から JVM メトリクスを取得し、ポート 9404 から Prometheus ワークスペースに公開します。Fargate にターゲット追跡ポリシーを追加します。スケールターゲットとして Prometheus メトリクスを選択します。

Answer: A

QUESTION NO: 11
ある企業は、AWS Organizations の組織を使用して多数の AWS
アカウントを管理しています。この組織ではすべての機能が有効化されています。アカウントへの設定のデプロイには AWS CloudFormation StackSets を使用しています。Amazon S3 バケットのモニタリングには AWS Config を使用しています。S3
バケットへのすべてのオブジェクトのアップロードに AWS Key Management Service (AWS KMS)
暗号化が使用されていることを確認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. s3-bucket-server-side-encryption-enabled ルールを含む AWS Config
適合パックを作成します。適合パックをアカウントにデプロイします。Amazon Simple Notification Service (Amazon SNS) トピックをターゲットとするルールを設定します。

B. s3:createBucket アクションの deny ステートメントと、s3:x-amz-server-side-encryption が aws:kms ではないという条件ステートメントを含む SCP を作成します。この SCP
を組織のルートにアタッチします。

C. AWS CloudFormation スタックセットを作成し、AWS CloudTrail 証跡を使用して組織の S3 データイベントをキャプチャできるようにします。このスタックセットで、AWS KMS 暗号化を使用しない S3 PutObject イベントに一致する Amazon EventBridge
ルールを作成します。Amazon Simple Notification Service (Amazon SNS)
トピックをターゲットとするルールを設定します。

D. s3:putObject アクションの拒否ステートメントと、s3:x-amz-server-side-encryption が aws:kms ではないという条件を含む SCP を作成します。この SCP
を組織のルートにアタッチします。

Answer: D

QUESTION NO: 12
ある企業は、新機能の開発時間を短縮したいと考えています。アプリケーションの構築とデプロイにはAWS CodeBuildとAWS
CodeDeployを使用しています。また、各マイクロサービスを独自のCI/CDパイプラインでデプロイするためにAWS
CodePipelineを使用しています。新機能のリリース間隔の平均時間と、デプロイ失敗後の復旧時間の平均について、より詳細な可視性が必要です。この可視性を、最小限の設定作業で実現できるソリューションはどれでしょうか？

A. 各パイプラインの成功実行と失敗実行に関する情報を含む Amazon CloudWatch
カスタムメトリクスを作成する AWS Lambda 関数をプログラムします。5 分ごとに Lambda 関数を呼び出す Amazon EventBridge
ルールを作成します。これらのメトリクスを使用して、CloudWatch
ダッシュボードを構築します。

B. 各パイプラインの成功実行と失敗実行に関する情報を含む Amazon CloudWatch
カスタムメトリクスを作成する AWS Lambda 関数をプログラムします。Amazon EventBridge ルールを作成し、成功実行と失敗実行のたびに Lambda
関数を起動します。これらのメトリクスを使用して、CloudWatch
ダッシュボードを構築します。

C. 成功した実行と失敗した実行に関する情報を Amazon DynamoDB に書き込む AWS

Lambda 関数をプログラムします。Amazon EventBridge
ルールを作成し、成功した実行と失敗した実行のたびに Lambda
関数を起動します。DynamoDB からの情報を表示する Amazon QuickSight
ダッシュボードを構築します。

D. 成功した実行と失敗した実行に関する情報を Amazon DynamoDB に書き込む AWS Lambda 関数をプログラムします。5 分ごとに Lambda 関数を呼び出す Amazon EventBridge ルールを作成します。DynamoDB からの情報を表示する Amazon QuickSight ダッシュボードを構築します。

Answer: B

QUESTION NO: 13

企業は AWS Organizations を使用して部門ごとに個別の AWS
アカウントを作成しています。この企業は次のタスクを自動化する必要があります。
* Linux AMI を新しいパッチで定期的に更新し、ゴールデン イメージを生成します
* ゴールデン イメージの Chef エージェントに新しいバージョンをインストールできます。
* 新しく生成された AMI を部門のアカウントに提供します
最小限の管理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すか?

A. 以前のゴールデン イメージから Amazon EC2
インスタンスを起動するスクリプトを作成します。 パッチ更新を適用します。
新しいバージョンの Chef エージェントをインストールし、新しいゴールデン
イメージを生成し、新しいイメージのみを共有するように AMI
アクセス許可を変更します。部門のアカウントの画像。

B. Amazon EC2 Image Builder を使用して、ベース Linux AMI と Chef
エージェントをインストールするコンポーネントで構成されるイメージ
パイプラインを作成します。AWS Resource Access Manager を使用して、EC2 Image Builder イメージを部門のアカウントと共有します。

C. AWS Systems Manager Automation Runbook を使用して、前のイメージを使用して Linux AMI を更新します。 Chef エージェントを更新するスクリプトの URL を指定します。AWS Organizations を使用して、部門のアカウント内の以前のゴールデン
イメージを置き換えます。

D. Amazon EC2 Image Builder を使用して、ベース Linux AMI と Chef
エージェントをインストールするコンポーネントで構成されるイメージ
パイプラインを作成します。AWS Systems Manager パラメータ
ストアにパラメータを作成して、次の AMI ID で参照できる新しい AMI ID
を保存します。部門のアカウント

Answer: B

Explanation:

Amazon EC2 Image Builder is a service that automates the creation, management, and deployment of customized, secure, and up-to-date server images that are pre-installed with software and configuration settings tailored to meet specific IT standards. EC2 Image Builder simplifies the creation and maintenance of golden images, and makes it easy to generate images for multiple platforms, such as Amazon EC2 and on- premises. EC2 Image Builder also integrates with AWS Resource Access Manager, which allows you to share your images across accounts within your organization or with external AWS accounts. This solution meets the requirements of automating the tasks of updating the Linux AMIs, installing the Chef

agent, and providing the images to the department's accounts with the least management overhead. References:
Amazon EC2 Image Builder
Sharing EC2 Image Builder images

QUESTION NO: 14
ビデオ共有会社はビデオを Amazon S3
に保存しています。同社はビデオへのアクセス要求の突然の増加を観察しましたが、どのビデオが最も人気があるのかはわかりません。会社はビデオ ファイルの一般的なアクセス パターンを特定する必要があります。このパターンには、特定の日に特定のファイルにアクセスするユーザーの数と、無感覚なユーザーの数が含まれます。 DevOps
エンジニアは、Amazon EC2 で実行される大規模な商用 Web サイトを管理しています。Web サイトは、Amazon Kinesis Data Streams を使用して Web togs
を収集および処理します。 DevOpsエンジニアは Kinesis コンシューマ
アプリケーションを管理します。
企業はどうすれば最小限の労力でこれらの要件を満たすことができるでしょうか?
A. S3 サーバーのアクセスログを有効にします。アクセス ログを Amazon Aurora
データベースにインポートします。Aurora SQL クエリを使用してアクセス
パターンを分析します。
B. S3 サーバーのアクセスログを有効にします。Amazon Athena を使用して、ログ
ファイルを含む外部テーブルを作成します。Athena を使用して SQL
クエリを作成し、アクセス パターンを分析します。
C. S3 オブジェクト アクセス イベントごとに AWS Lambda
関数を呼び出します。ユーザーなどのファイルアクセス情報を書き込むように Lambda
関数を設定します。S3 バケットとファイルキーを Amazon Aurora
データベースに追加します。Aurora SQL クエリを使用してアクセス
パターンを分析します。
D. すべての S3 オブジェクト アクセス イベントの Amazon CloudWatch Logs ログ
メッセージを記録します。ユーザー、S3
バケット、ファイルキーなどのファイルアクセス情報を Amazon Kinesis Data Analytics for SQL アプリケーションに書き込むように CloudWatch Logs
ログストリームを設定します。スライディング ウィンドウ分析を実行します。
Answer: B
Explanation:
Activating S3 server access logging and using Amazon Athena to create an external table with the log files is the easiest and most cost-effective way to analyze access patterns. This option requires minimal setup and allows for quick analysis of the access patterns with SQL queries. Additionally, Amazon Athena scales automatically to match the query load, so there is no need for additional infrastructure provisioning or management.

QUESTION NO: 15
ある企業では、Auto Scaling グループ内の Amazon EC2
インスタンス上で実行される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、Amazon Simple Queue Service (Amazon SQS)
キューからの大量のメッセージを処理します。

DevOpsエンジニアは、アプリケーションがSQSキューからの一連のメッセージを処理するのに数時間かかっていることに気付きました。メッセージ処理時のAuto Scalingグループの平均CPU使用率は、ターゲットトラッキングスケーリングポリシーのしきい値を超えませんでした。SQSキューを処理するアプリケーションは、Amazon CloudWatch Logsに口グを発行します。
DevOps エンジニアは、キューが迅速に処理されるようにする必要があります。
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すか?

A. AWS Lambda 関数を作成します。SQS キュー属性 approximateNumberOfMessagesVisible と Auto Scaling グループ属性 GroupIn-ServiceInstances を使用してカスタムメトリクスを発行し、各インスタンスのキューメッセージをパブリッシュするように Lambda 関数を設定します。Amazon EventBridge ルールをスケジュールし、Lambda 関数を 1 時間ごとに実行します。カスタムメトリクスを使用してスケールインおよびスケールアウトする Auto Scaling グループのターゲット追跡スケーリングポリシーを作成します。

B. AWS Lambda 関数を作成します。SQS キュー属性 approximateNumberOfMessagesVisible と Auto Scaling グループ属性 GroupIn-ServiceInstances を使用してカスタムメトリクスを発行するように Lambda 関数を設定し、各インスタンスのキューメッセージを公開します。Lambda 関数をターゲットとして、アプリケーション口グ用の CloudWatch サブスクリプションフィルターを作成します。カスタムメトリクスを使用してスケールインおよびスケールアウトする Auto Scaling グループのターゲット追跡スケーリングポリシーを作成します。

C. Auto Scaling グループのターゲット追跡スケーリングポリシーを作成します。ターゲット追跡ポリシーでは、SQS キュー属性 approximateNumberOfMessagesVisible と Auto Scaling グループ属性 GroupIn-ServiceInstances を使用して、メトリック計算によりインスタンス数ごとにキュー内のメッセージ数を計算します。計算された属性を使用してスケールインおよびスケールアウトを行います。

D. SQSキューの approximateNumberOfMessagesVisible 属性をCloudWatch Logs口グループに記録するAWS Lambda関数を作成します。Amazon EventBridgeルールをスケジュールし、Lambda関数を5分ごとに実行します。CloudWatch Logsグループからの口グイベント数をカウントするメトリクスフィルターを作成します。カスタムメトリクスを使用してスケールイン/スケールアウトする、Auto Scalingグループのターゲット追跡スケーリングポリシーを作成します。

Answer: C

Explanation:
The default CPU utilization metric does not reflect the processing backlog in the SQS queue, so the Auto Scaling group is not scaling properly to handle the workload.
To scale the Auto Scaling group based on queue length, you can create a target tracking scaling policy that uses a custom metric that combines the SQS queue ' s ApproximateNumberOfMessagesVisible and the number of instances (GroupIn-ServiceInstances) metric using CloudWatch metric math. This allows the scaling policy to calculate the average number of messages per instance and scale accordingly.
This approach requires no additional Lambda functions or log processing, thus minimizing

operational overhead.
Option A and B require Lambda functions to publish custom metrics, which increases operational complexity.
Option D also adds complexity with logging and metric filters.
Reference:
Scaling based on SQS queue length using metric math: " You can create CloudWatch metric math expressions combining SQS and Auto Scaling group metrics to enable target tracking scaling policies that respond to queue backlog. " (AWS Auto Scaling with SQS) Target Tracking Scaling Policies: " Target tracking policies can use metric math expressions as a source to make scaling decisions. " (AWS Auto Scaling Target Tracking)

QUESTION NO: 16
DevOpsエンジニアは、AWSアカウントにCI/CDパイプラインを実装する必要があります。このパイプラインは、AWS Systems Managerパラメータストアパラメータに保存されている機密性の高いデータベース認証情報を使用する必要があります。パラメータストアパラメータは、別の中央アカウントに保存されています。DevOpsエンジニアは、このパラメータを作成し、CI/CDアカウントに統合する必要があります。
これらの要件を満たす手順の組み合わせはどれですか? (3 つ選択してください)。
A. 高度なレベルのパラメータ ストア
パラメータを使用して、データベース認証情報を中央の AWS アカウントに保存します。
B.
CI/CDパイプラインをホストするAWSアカウントにIAMロールを作成します。パラメータの完全なARNを、IAMロールに関連付けられたIAMポリシーに追加します。
C. 標準層のパラメータ ストア パラメータを使用して、データベース認証情報を中央の AWS アカウントに保存します。
D. AWS KMS 管理キーを使用してパラメータを暗号化します。CI/CD
パイプラインをホストする AWS アカウントに、KMS キーの復号権限を付与します。
E. パラメータを暗号化するには、カスタマー管理のAWS KMSキーを使用します。CI/CDパイプラインをホストするAWSアカウントに、カスタマー管理キーの復号権限を付与します。
F. 中央の AWS アカウントに AWS Resource Access Manager (AWS RAM)
リソース共有を作成します。CI/CD
パイプラインをホストするアカウントとパラメータを共有します。
Answer: A E F
Explanation:

###
The requirement is to securely consume cross-account sensitive parameters from AWS Systems Manager Parameter Store in a CI/CD pipeline. AWS imposes specific constraints and features for cross-account parameter access, and the correct solution must align with those constraints.
First, cross-account sharing of Parameter Store parameters is supported only for Advanced tier parameters.
Standard tier parameters cannot be shared across accounts. Therefore, Option A is required,

and Option C is invalid.
Second, cross-account access to Parameter Store parameters is implemented using AWS Resource Access Manager (AWS RAM). RAM allows the central account to explicitly share the parameter resource with the CI
/CD account. Without RAM, the parameter is not visible or accessible across accounts. This makes Option F mandatory.
Third, because the parameter stores sensitive credentials, encryption must be handled securely. When sharing encrypted parameters across accounts, AWS requires the use of a customer managed AWS KMS key, not an AWS managed key. The key policy must explicitly grant `kms:Decrypt` permission to the consuming account. AWS managed keys cannot be shared across accounts, which makes Option D invalid and Option E correct.
Although the CI/CD pipeline's IAM role must ultimately have permission to read the parameter, that permission is implicit in the consuming account once the parameter is shared and the KMS key policy allows decryption. The critical cross-account enablers are Advanced tier, RAM sharing, and a customer managed KMS key.
Therefore, the correct combination is A, E, and F.

QUESTION NO: 17
ある企業は、AWS CodePipeline
のパイプラインを使用してアプリケーションをデプロイしています。アプリケーションの耐障害性をテストするために、AWS Fault Injection Service (AWS FIS)
の実験テンプレートを作成しました。DevOps
エンジニアは、この実験をパイプラインに統合する必要があります。
どのソリューションがこの要件を満たすでしょうか?
A. AWS FIS アクションを含むパイプラインの新しいステージを設定します。AWS FIS 実験テンプレートを参照するようにアクションを設定します。実験を開始するためのアクセス権限をパイプラインに付与します。
B. Amazon EventBridge スケジューラを作成します。スケジューラに AWS FIS 実験を開始する権限を付与します。パイプラインに、EventBridge スケジューラを呼び出すアクションを含む新しいステージを設定します。
C. AWS FIS 実験を開始するための AWS Lambda 関数を作成します。Lambda 関数に実験を開始する権限を付与します。パイプラインに Lambda アクションを含む新しいステージを作成します。Lambda 関数を呼び出すアクションを設定します。
D. AWS FIS 実験テンプレートを Amazon S3 バケットにエクスポートします。AWS FIS 実験を開始するビルドスペックを含む AWS CodeBuild ユニットテストプロジェクトを作成します。実験を開始するためのアクセス権限を CodeBuild プロジェクトに付与します。パイプラインに、CodeBuild ユニットテストプロジェクトを実行するアクションを含む新しいステージを設定します。
Answer: C

QUESTION NO: 18
IT
チームは、社内の他のユーザーがアプリケーションを迅速かつ確実にデプロイおよび終了できるように AWS CloudFormation

テンプレートを構築しました。このテンプレートは、アプリケーションをインストールするためのユーザー データ スクリプトを含む Amazon EC2
インスタンスと、アプリケーションが実行中に静的 Web ページを提供するために使用する Amazon S3 バケットを作成します。
CloudFormation
スタックが削除されるときは、すべてのリソースを削除する必要があります。ただし、チームは、CloudFormation
がスタックの削除中にエラーを報告し、スタックによって作成された S3
バケットが削除されていないことを観察しました。
すべてのリソースがエラーなく削除されるように、チームは最も効率的な方法でエラーを解決するにはどうすればよいでしょうか?
A. DelelionPolicy 属性を S3 バケット リソースに追加します。その値は Delete
で、スタックの削除時にバケットが強制的に削除されます。
B. S3 バケットを指定する dependsOn 属性を持つ AWS Lambda 関数と IAM
ロールを含むカスタム リソースを追加します。RequestType が Delete
の場合にバケットからすべてのオブジェクトを削除する Lambda 関数を作成します。
C. 削除されなかったリソースを特定します。S3
バケットを手動で空にしてから削除します。
D. EC2 および S3 バケット リソースを単一の AWS OpsWorks スタック
リソースに置き換えます。EC2 インスタンスと S3
バケットを作成および削除するスタックのカスタム レシピを定義します。
Answer: B
Explanation:
https://aws.amazon.com/premiumsupport/knowledge-center/cloudformation-s3-custom-resources/

QUESTION NO: 19
ある企業が、Amazon EC2 ノードグループを使用して Amazon Elastic Kubernetes Service (Amazon EKS) クラスターをデプロイしました。同社の DevOps チームは Kubernetes Horizontal Pod Autoscaler を使用しており、最近、サポート対象の EKS クラスター Autoscaler をインストールしました。
DevOps チームは、EKS
クラスターのメトリクスとログを収集し、パフォーマンスのベースラインを確立するためのソリューションを実装する必要があります。DevOps
チームは、特定のメトリクスの初期しきい値セットを作成し、クラスターの使用状況に応じてしきい値を更新します。初期しきい値セットを超えた場合、または EKS
クラスターのオートスケーラーが正常に機能していない場合は、DevOps チームが Amazon Simple Notification Service (Amazon SNS) からのメール通知を受信する必要があります。
このソリューションでは、クラスター、ノード、ポッドのメトリクスを収集する必要があります。また、Amazon CloudWatch でログをキャプチャする必要があります。
これらの要件を満たすために、DevOps
チームはどのような組み合わせの手順を実行する必要がありますか? (3
つ選択してください。)
A. CloudWatchエージェントとFluent
Bitをクラスターにデプロイします。EKSクラスターに、メトリクスとログをCloudWatchに

送信するための適切な権限があることを確認します。
B. AWS Distro for OpenTelemetry をクラスターにデプロイします。EKS クラスターにメトリクスとログを CloudWatch に送信するための適切な権限があることを確認します。
C. クラスターの CPU、メモリ、ノード障害のメトリクスを監視するための CloudWatch アラームを作成します。しきい値を超えた場合に DevOps チームに SNS 電子メール通知を送信するようにアラームを設定します。
D. クラスターのCPU、メモリ、ノードメトリクスのメトリクスログフィルターを監視するためのCloudWatch複合アラームを作成します。異常が検出されると、DevOpsチームにSNSメール通知を送信するようにアラームを設定します。
E. CloudWatch アラームを作成し、Autoscaler デプロイメントのログでエラーを監視します。しきい値を超えた場合に SNS メール通知を DevOps チームに送信するようにアラームを設定します。
F. Autoscaler デプロイメントのメトリクス ログ フィルターのエラーを監視するための CloudWatch アラームを作成します。しきい値を超えた場合に DevOps チームに SNS メール通知を送信するようにアラームを設定します。
Answer: A C F
Explanation:
Deploy CloudWatch Agent + Fluent Bit (supported by Amazon EKS integration) to forward metrics and logs.
Create CloudWatch Alarms for CPU/memory/node metrics and metric log filters for Autoscaler logs, triggering SNS notifications when anomalies occur. This pattern matches AWS guidance on "Monitoring EKS clusters with CloudWatch Container Insights and alarms. "

QUESTION NO: 20
ある成長企業は、AWS Organizations の組織内で 50 を超えるアカウントを管理しています。同社は、ログを Amazon CloudWatch Logs に送信するようにアプリケーションを構成しました。
DevOps エンジニアは、企業が将来のセキュリティ インシデントに対応するためにログを迅速に検索できるように、ログを集約する必要があります。DevOps エンジニアは、集中監視用に新しい AWS アカウントを作成しました。
監視アカウントからアプリケーション ログを検索できるようにするには、DevOps エンジニアはどの手順を組み合わせる必要がありますか? (3 つ選択してください。)
A. モニタリング アカウントで、組織で使用するために CloudWatch から AWS CloudFormation テンプレートをダウンロードします。組織の管理アカウントで CloudFormation StackSets を使用して、CloudFormation テンプレートを組織全体にデプロイします。
B. IAM ロールを定義する AWS CloudFormation テンプレートを作成します。aws:ResourceAccount プロパティがモニタリングアカウント ID と等しい場合、logs-amazonaws.com が logs:Link アクションを実行できるようにロールを設定します。組織の管理アカウントで CloudFormation StackSets を使用して、CloudFormation

テンプレートを組織全体にデプロイします。
C. モニタリング アカウントに IAM ロールを作成します。aws:PrincipalOrgId プロパティが組織 ID と等しい場合に、logs.amazonaws.com が iam:CreateSink アクションを実行できるようにする信頼ポリシーをアタッチします。
D. 組織の管理アカウントで、組織のログ ポリシーを有効にします。
E. モニタリング アカウントで CloudWatch Observability Access Manager を使用してシンクを作成します。ログを監視アカウントと共有できるようにします。組織 ID から可観測性データを表示するには、監視アカウント データの選択を構成します。
F. モニタリング アカウントで、ログの検索を想定できる IAM ロールに CloudWatchLogsReadOnlyAccess AWS 管理ポリシーをアタッチします。
Answer: B C F
Explanation:
To aggregate logs from multiple accounts in an organization, the DevOps engineer needs to create a cross- account subscription1 that allows the monitoring account to receive log events from the sharing accounts.
To enable cross-account subscription, the DevOps engineer needs to create an IAM role in each sharing account that grants permission to CloudWatch Logs to link the log groups to the destination in the monitoring account2. This can be done using a CloudFormation template and StackSets3 to deploy the role to all accounts in the organization.
The DevOps engineer also needs to create an IAM role in the monitoring account that allows CloudWatch Logs to create a sink for receiving log events from other accounts4. The role must have a trust policy that specifies the organization ID as a condition.
Finally, the DevOps engineer needs to attach the CloudWatchLogsReadOnlyAccess policy5 to an IAM role in the monitoring account that can be used to search the logs from the cross-account subscription.
1: Cross-account log data sharing with subscriptions 2: Create an IAM role for CloudWatch Logs in each sharing account 3: AWS CloudFormation StackSets 4: Create an IAM role for CloudWatch Logs in your monitoring account 5: CloudWatchLogsReadOnlyAccess policy

QUESTION NO: 21
ある企業がAmazon Elastic Kubernetes Service (Amazon EKS) 上でマイクロサービスアプリケーションを運用しています。最近、ユーザーから、特に営業時間のピーク時にアカウント概要機能へのアクセスに大幅な遅延が発生するという報告がありました。
DevOpsエンジニアはAmazon CloudWatchのメトリクスとログを使用して問題のトラブルシューティングを行いました。ログにはEKSノードのCPUとメモリの使用率が正常であることが示されていましたが、マイクロサービスアーキテクチャ内のどこで遅延が発生しているかを特定することはできませんでした。
DevOps エンジニアは、遅延が発生している場所を正確に特定するために、アプリケーションの可観測性を高める必要があります。
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. AWS X-RayデーモンをDaemonSetとしてEKSクラスターにデプロイします。X-Ray SDKを使用してアプリケーションコードをインストルメントします。アプリケーションを再

デプロイします。
B. EKSクラスターでCloudWatch Container Insightsを有効にします。Container Insightsのデータを使用して遅延を診断します。
C. 既存の CloudWatch メトリクスに基づいてアラームを作成します。Amazon Simple Notification Service (Amazon SNS) トピックを設定して、メールアラートを送信します。
D. ネットワーク操作のアプリケーション
コードのタイムアウト設定を増やして、操作が完了するまでの時間を長くします。
Answer: A
Explanation:
AWS X-Ray provides distributed tracing for microservice-based applications. Deploying the X-Ray daemon as a DaemonSet in the EKS cluster and instrumenting the application with the X-Ray SDK enables end-to-end tracing across microservices, helping identify performance bottlenecks. This method is documented in "Using AWS X-Ray with Amazon EKS" (AWS Observability Guide).

QUESTION NO: 22
ある企業は、AWS CodeConnections 対応の Git
リポジトリにアプリケーションコードを保存しています。プルリクエストが開かれたときにユニットテストを実行するように設定したいと考えています。また、テスト完了時にプルリクエストでテストステータスが表示されるようにしたいと考えています。さらに、テスト完了後に、テストで生成された出力データファイルを Amazon S3
バケットに保存したいと考えています。これらの要件を満たすソリューションの組み合わせはどれですか？ (3 つ選択してください。)
A. テストの実行に必要なリソースへのアクセスを許可する IAM サービス
ロールを作成します。
B. AWS CodePipeline
でテストステージを含むパイプラインを作成します。プルリクエストが作成または更新されたときにパイプラインを実行するトリガーを作成します。テスト結果をレポートするためのソースアクションを追加します。
C. テストを実行するためのAWS
CodeBuildプロジェクトを作成します。プルリクエストが作成または更新されたときにテストを実行するために、Webhookトリガーを有効にします。テスト結果を報告するために、ビルドステータスレポートを有効にします。
D. テストの実行が完了したときに出力ファイルをアップロードするためのレポート
セクションを含む buildspec.yml ファイルを作成します。
E. テストの実行が完了したときに成果物をアップロードするためのアーティファクト
セクションを含む buildspec.yml ファイルを作成します。
F. テストの実行が完了したときに出力ファイルをアップロードするためのファイル
セクションを持つ appspec.yml ファイルを作成します。
Answer: A C E

QUESTION NO: 23
ある企業は、コンテナ化されたインフラストラクチャのすべてのイメージにAmazon Elastic Container Registry (Amazon

ECR) を使用しています。外部イメージレジストリからイメージを取得する際のスロットリングを回避するため、/externalプレフィックスを付与したプルスルーキャッシュ機能を使用しています。また、アカウントにはAWS Organizationsを使用しています。
レジストリ内のすべてのイメージは、事前にプロビジョニングされた特定のAWS Key Management Service (AWS KMS)
キーで暗号化する必要があります。社内で作成されたイメージは、すでにこのポリシーに準拠しています。
ただし、キャッシュされた外部イメージでは、Amazon S3 管理キー (SSE-S3)
によるサーバー側暗号化が使用されます。
企業は、非準拠のキャッシュリポジトリを削除する必要があります。また、すべての新しいプルスルーキャッシュリポジトリが必要なKMSキーで自動的に暗号化されるように、安全なソリューションを実装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. AWS Config を設定します。Guard
構文を使用するカスタムルールを追加します。新しいリポジトリで KMS
暗号化を有効にするルールを記述します。

B.
プレフィックスのECRリポジトリ作成テンプレートを設定します。KMSキーを指定します。リポジトリが再作成されるまで待ちます。

C. すべての ECR リポジトリを KMS で暗号化することを要求する SCP をすべての AWS アカウントに設定します。

D. すべての「ECR プルスルーキャッシュアクション」イベントでトリガーされる新しい Amazon EventBridge ルールを作成します。ルールのターゲットとして AWS KMS
を設定します。

Answer: B

Explanation:
For pull through cache repositories, Amazon ECR now supports repository creation templates that can be applied to a registry prefix, such as /external . These templates define default settings, including encryption configuration with a specific KMS key , tag immutability, scan on push, and more. When new cache repositories are auto-created under that prefix, they inherit the template settings automatically.
In this scenario, existing external cache repositories are noncompliant because they use SSE-S3. The company can delete those repositories (removing the noncompliant caches) and configure an ECR repository creation template for the /external prefix that specifies the required customer managed KMS key. As new images are pulled, ECR recreates the cache repositories under that prefix with KMS encryption using the specified key , guaranteeing compliance going forward.
Option A (AWS Config) would only detect noncompliance after creation and cannot enforce encryption at creation time. Option C (SCP) cannot directly control repository encryption properties. Option D misuses EventBridge; KMS cannot be a "target" that retroactively encrypts repositories.
Therefore, using an ECR repository creation template with the desired KMS key is the correct, automatic, and secure solution.

QUESTION NO: 24

ある企業はAWS Organizationsに組織を持っています。DevOpsエンジニアは、組織内の異なるOUに属する複数のAWSアカウントを管理する必要があります。アカウント内の1AMポリシーやAmazon S3ポリシーを含むすべてのリソースは、AWS CloudFormationを通じてデプロイされます。すべてのテンプレートとコードはAWS CodeCommitリポジトリで管理されています。最近、一部の開発者が組織内の一部のアカウントからS3バケットにアクセスできない問題が発生しています。
次のポリシーが S3 バケットにアタッチされています。
このアクセス問題を解決するために DevOps エンジニアは何をすべきでしょうか?

A.
S3バケットポリシーを変更します。S3バケットのS3ブロックパブリックアクセス設定をオフにします。S3ポリシーにawsSourceAccount条件を追加します。問題が発生しているすべての開発者のAWSアカウントIDを追加します。

B. 1AM 権限境界によって開発者の S3 バケットへのアクセスが拒否されていないことを確認します。IAM 権限境界に必要な変更を加えます。問題が発生している個々の開発者アカウントで AWS Config レコーダーを使用して、アクセスをブロックしている変更を元に戻します。修正をCodeCommitリポジトリにコミットし、Cloud Formationを介してデプロイを呼び出して変更を適用します。

C.
開発者OU内の1AMリソースを誰も変更できないようにするSCPを設定します。S3ポリシーにawsSourceAccount条件を追加します。問題が発生しているすべての開発者のAWSアカウントIDを追加します。修正をCodeCommitリポジトリにコミットします。変更を適用するために、CloudFormationを介してデプロイメントを呼び出します。

D. SCP が開発者の S3 バケットへのアクセスをブロックしていないことを確認する 1AM ポリシーの権限境界が開発者の 1AM ユーザーへのアクセスを拒否していないことを確認する CodeCommit リポジトリの SCP と 1AM ポリシーの権限境界に必要な変更を加える 変更を適用するために CloudFormation を介してデプロイメントを呼び出す

Answer: D

Explanation:
Verify No SCP Blocking Access:
Ensure that no Service Control Policy (SCP) is blocking access for developers to the S3 bucket. SCPs are applied at the organization or organizational unit (OU) level in AWS Organizations and can restrict what actions users and roles in the affected accounts can perform.
Verify No IAM Policy Permissions Boundaries Blocking Access:
IAM permissions boundaries can limit the maximum permissions that a user or role can have. Verify that these boundaries are not restricting access to the S3 bucket.
Make Necessary Changes to SCP and IAM Policy Permissions Boundaries:
Adjust the SCPs and IAM permissions boundaries if they are found to be the cause of the access issue. Make sure these changes are reflected in the code maintained in the AWS CodeCommit repository.
Invoke Deployment Through CloudFormation:
Commit the updated policies to the CodeCommit repository.

Use AWS CloudFormation to deploy the changes across the relevant accounts and resources to ensure that the updated permissions are applied consistently.
By ensuring no SCPs or IAM policy permissions boundaries are blocking access and making necessary changes if they are, the DevOps engineer can resolve the access issue for developers trying to access the S3 bucket.
References:
AWS SCPs
IAM Permissions Boundaries
Deploying CloudFormation Templates

QUESTION NO: 25
ある企業には、単一の共有 AWS
アカウントで作業する複数の開発グループがあります。グループのシニア
マネージャーは、リソースの作成がアカウントのサービス制限に近づいたときに、サードパーティ API 呼び出しを介してアラートを受け取りたいと考えています。
最小限の開発労力でこれを達成できるソリューションはどれですか?
A. 定期的に実行され、AWS Lambda 関数をターゲットとする Amazon CloudWatch Event ルールを作成します。Lambda 関数内で、AWS
環境の現在の状態を評価し、デプロイされたリソースの値をアカウントのリソース制限と比較します。アカウントがサービス制限に近づいている場合は、シニア
マネージャーに通知してください。
B. AWS Trusted Advisor チェックを更新する AWS Lambda 関数をデプロイし、Lambda 関数を定期的に実行するように Amazon CloudWatch Events ルールを設定します。Trusted Advisor イベントとターゲット Lambda 関数に一致するイベント パターンを使用して、別の CloudWatch イベント ルールを作成します。対象の Lambda
関数で、シニアマネージャーに通知します。
C. AWS Personal Health Dashboard チェックを更新する AWS Lambda
関数をデプロイし、Lambda 関数を定期的に実行するように Amazon CloudWatch Events ルールを設定します。Personal Health Dashboard イベントとターゲット Lambda
関数に一致するイベント パターンを持つ別の CloudWatch Events
ルールを作成します。対象の Lambda 関数で、シニアマネージャーに通知します。
D. 定期的に実行され、AWS のサービス制限ステータスをチェックし、Amazon SNS トピックに通知をストリーミングする AWS Config
カスタムルールを追加します。シニアマネージャーに通知する AWS Lambda
関数をデプロイし、Lambda 関数を SNS トピックにサブスクライブします。
Answer: B
Explanation:
To meet the requirements, the company needs to create a solution that alerts the Senior Manager when the creation of resources approaches the service limits for the account with the least amount of development effort. The company can use AWS Trusted Advisor, which is a service that provides best practice recommendations for cost optimization, performance, security, and service limits. The company can deploy an AWS Lambda function that refreshes Trusted Advisor checks, and configure an Amazon CloudWatch Events rule to run the Lambda function periodically. This will ensure that Trusted Advisor checks are up to date and reflect the current state of the account. The company can then create another

CloudWatch Events rule with an event pattern matching Trusted Advisor events and a target Lambda function. The event pattern can filter for events related to service limit checks and their status. The target Lambda function can notify the Senior Manager via a third-party API call if the event indicates that the account is approaching or exceeding a service limit.

QUESTION NO: 26
ある企業は、AWS Organizations で組織のすべての機能を有効にしました。組織には 10 個の AWS アカウントが含まれています。同社はすべてのアカウントで AWS CloudTrail を有効にしました。同社は、組織内の AWS アカウントの数が来年中に 500 に増加すると予想しています。同社はこれらのアカウントに複数の OU を使用する予定です。
同社は、組織内の既存の各 AWS アカウントで AWS Config を有効にしました。DevOps エンジニアは、組織内で今後作成されるすべての AWS アカウントに対して AWS Config を自動的に有効にするソリューションを実装する必要があります。
この要件を満たすソリューションはどれですか?
A. 組織の管理アカウントで、CreateAccount API 呼び出しに反応する Amazon EventBridge ルールを作成します。組織の AWS Config への信頼されたアクセスを可能にする AWS Lambda 関数を呼び出すルールを設定します。
B. 組織の管理アカウントで、AWS Config を有効にする AWS CloudFormation スタックセットを作成します。組織を通じてアカウントが作成されたときにスタックセットが自動的に展開されるように構成します。
C. 組織の管理アカウントで、AWS Config を有効にするための適切な AWS Config API 呼び出しを許可する SCP を作成します。SCP をルートレベルの OU に適用します。
D. 組織の管理アカウントで、CreateAccount API 呼び出しに反応する Amazon EventBridge ルールを作成します。AWS Systems Manager Automation Runbook を呼び出してアカウントの AWS Config を有効にするルールを設定します。
Answer: B
Explanation:
https://aws.amazon.com/about-aws/whats-new/2020/02/aws-cloudformation-stacksets-introduces-automatic- deployments-across-accounts-and-regions-through-aws-organizations/

QUESTION NO: 27
ある企業は、Amazon EC2 起動タイプで Amazon Elastic Container Service (Amazon ECS) を使用しています。この企業では、すべてのログデータを Amazon CloudWatch で一元管理する必要があります。この企業の ECS タスクには、ログドライバー名に awslogs を指定する LogConfiguration オブジェクトが含まれています。
同社のECSタスクのデプロイに失敗しました。エラーメッセージには、権限不足が失敗の原因であることを示しています。同社は、コンテナインスタンスの起動に使用したIAMロールに以下のログが含まれていることを確認しました。
CreateLogGroup、logs:CreateLogStream、および logs:PutLogEvents 権限。
どの解決策が問題を解決しますか?
A. Amazon ECS を信頼できるサービスとして確立する IAM ロールに IAM 信頼ポリシーを追加します。
B. IAM ロールに適用されたポリシーに logs:PutDestination 権限を追加します。
C. IAM ロールに適用されたポリシーから logs:CreateLogStream 権限を削除します。

D. CloudWatch を信頼できるサービスとして確立する IAM ロールに IAM 信頼ポリシーを追加します。

Answer: A

Explanation:

When using the awslogs log driver with ECS on EC2, the ECS agent running on the container instance uses the instance's IAM role (container instance role or task execution role, depending on configuration) to write logs to CloudWatch Logs. The policy already grants logs:CreateLogGroup , logs:CreateLogStream , and logs:

PutLogEvents , which are the required CloudWatch Logs actions. However, for the role to be usable by ECS, the role's trust policy must allow the appropriate service principal to assume it.

In this question, the error message indicates "missing permission" during ECS task deployment. If the IAM role is not trusted by the ECS service (for example, ecs-tasks.amazonaws.com for a task execution role or the proper principal for container instances), ECS cannot assume that role and therefore cannot use the granted CloudWatch permissions, causing deployment failures.

Option A addresses this by adding a trust relationship so that Amazon ECS can assume the IAM role. Options B and C mutate the permissions but do not fix the underlying problem: the missing trust. Option D incorrectly attempts to trust CloudWatch, which does not assume roles in this context.

Thus, adding a trust policy that establishes ECS as a trusted service is the correct fix.

QUESTION NO: 28

ある企業は、トランクベースの開発ブランチ戦略を採用しています。この企業には、Gitプロバイダーと統合された2つのAWS CodePipelineパイプラインがあります。pull_requestパイプラインには、機能ブランチに一致するブランチフィルターが設定されています。main_branchパイプラインには、メインブランチに一致するブランチフィルターが設定されています。

プルリクエストがメインブランチにマージされると、main_branchパイプラインを使用してプルリクエストがデプロイされます。会社の開発者は、pull_requestパイプラインから送信されたすべてのプルリクエストのテスト結果をできるだけ早く取得する必要があります。会社は、次のパイプライン実行前にmain_branchパイプラインのテスト結果が完了し、各デプロイが完了していることを確認したいと考えています。

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.

pull_requestパイプラインをSUPERSEDEDモードに設定します。main_branchパイプラインをQUEUEDモードに設定します。

B.

pull_requestパイプラインをPARALLELモードに設定します。main_branchパイプラインをQUEUEDモードに設定します。

C.

pull_requestパイプラインをPARALLELモードに設定します。main_branchパイプラインをSUPERSEDEDモードに設定します。

D.

pull_requestパイプラインをQUEUEDモードに設定します。main_branchパイプラインをSU

PERSEDEDモードに設定します。

Answer: B

Explanation:

In CodePipeline's execution mode,

* PARALLEL mode for pull_request pipelines ensures that multiple feature branches can be tested simultaneously for quick feedback.

* QUEUED mode for main_branch ensures deployments run sequentially - each must finish before the next begins, preventing overlap. This configuration aligns with AWS CodePipeline best practices for trunk-based development and concurrent test pipelines.

QUESTION NO: 29

電子医療記録を使用する企業は、Amazon Linux オペレーティング
システムを搭載した一連の Amazon EC2
インスタンスを実行しています。患者のプライバシー要件の一環として、企業は、EC2
インスタンス上で実行されるオペレーティング
システムとアプリケーションのパッチに対する継続的なコンプライアンスを確保する必要があります。
デフォルトおよびカスタム リポジトリを使用して、オペレーティング
システムとアプリケーション パッチの展開を自動化するにはどうすればよいですか?

A. AWS Systems Manager を使用して、カスタム リポジトリを含む新しいパッチ
ベースラインを作成します。run コマンドを使用して AWS-RunPatchBaseline
ドキュメントを実行し、パッチを確認してインストールします。

B. AWS Direct Connect を使用して企業リポジトリを統合し、Amazon CloudWatch
のスケジュールされたイベントを使用してパッチをデプロイし、CloudWatch
ダッシュボードを使用してレポートを作成します。

C. yum-config-manager を使用して /etc/yum.repos.d にカスタム リポジトリを追加し、yum-config-manager-enable を実行してリポジトリをアクティブ化します。

D. AWS Systems Manager を使用して、企業リポジトリを含む新しいパッチ
ベースラインを作成します。run コマンドを使用して AWS-
AmazonLinuxDefaultPatchBaseline
ドキュメントを実行し、パッチを確認してインストールします。

Answer: A

Explanation:

https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-manager-how-it-works-alt-source- repository.html

QUESTION NO: 30

ある企業はS3を使用して画像を保存しており、双方向レプリケーションと15分以内の遅延を備えたマルチリージョンDRを必要としている。
要件を満たす手順はどれですか？(3つ選択してください。)

A. 各レプリケーションルールに対してS3レプリケーション時間制御(RTC)を有効にします。

B. S3マルチリージョンアクセスポイント(アクティブ/パッシブ)を作成します。

C. フェイルオーバー中に SubmitMultiRegionAccessPointRoutes を呼び出します。

D. S3転送アクセラレーションを有効にします。

E. Route 53 ARC ルーティング制御を使用します。
F. フェイルオーバー時にトラフィックを切り替えるために Route 53 ARC を使用します。
Answer: A B C
* S3 RTC ensures #15-min replication SLA.
* Multi-Region Access Point (MRAP) simplifies active-passive replication access.
* SubmitMultiRegionAccessPointRoutes API manages routing changes during failover. This configuration follows AWS DR best practices for S3 active/passive architectures.

QUESTION NO: 31
AWS Organizations 内の会社の組織には、単一の OU があります。会社は、OU アカウントで Amazon EC2 インスタンスを実行します。会社は、各 EC2 インスタンスの認証情報の使用を、認証情報が割り当てられている特定の EC2 インスタンスに制限する必要があります。DevOps エンジニアは、EC2 インスタンスのセキュリティを構成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. VPC CIDR ブロックを指定する SCP を作成します。 aws:VpcSourceIp 条件キーの値が指定されたブロック内にあるかどうかを確認するように SCP を設定します。同じ SCP チェックで、aws:EC2InstanceSourcePrivateIPv4 と aws:SourceVpc 条件キーの値が同じかどうかを確認します。
いずれかの条件が偽の場合、アクセスを拒否します。OU に SCP を適用します。
B. aws:EC2InstanceSourceVPC と aws:SourceVpc 条件キーの値が同じかどうかをチェックする SCP を作成します。値が同じでない場合はアクセスを拒否します。同じ SCP チェックで、aws:EC2InstanceSourcePrivateIPv4 と awsVpcSourceIp 条件キーの値が同じかどうかを確認します。値が同じでない場合はアクセスを拒否します。SCP を OU に適用します。
C. 許容される VPC 値のリストを含む SCP を作成し、aws の値が次の値であるかどうかを確認します。
SourceVpc 条件キーがリスト内にあります。同じ SCP チェックで、許容される IP アドレス値のリストを定義し、aws:VpcSourceIp 条件キーの値がリスト内に含まれているかどうかを確認します。いずれかの条件が false の場合、アクセスを拒否します。組織内の各アカウントに SCP を適用します。
D. aws:EC2InstanceSourceVPC と aws:VpcSourceIp 条件キーの値が同じかどうかをチェックする SCP を作成します。値が同じでない場合はアクセスを拒否します。同じ SCP チェックで、aws:EC2InstanceSourcePrivatoIPv4 と aws:SourceVpc 条件キーの値が同じかどうかをチェックします。値が同じでない場合はアクセスを拒否します。組織内の各アカウントに SCP を適用します。
Answer: B
Explanation:
Step 1: Using Service Control Policies (SCPs) for EC2 SecurityTo limit the use of EC2 instance credentials to the specific EC2 instance they are assigned to, you can create a Service Control Policy (SCP) that verifies specific conditions, such as whether the EC2 instance ' s source VPC and private IP match expected values.
Action: Create an SCP that checks whether the values of the aws:EC2InstanceSourceVPC

and aws:
SourceVpc condition keys are the same. Deny access if they are not.
Why: This ensures that credentials cannot be used outside the designated EC2 instance or VPC.
Step 2: Further Validation with Private IPsThe SCP should also verify that the EC2 instance ' s private IP matches the IP range specified for the VPC. If the instance ' s private IP does not match, access should be denied.
Action: In the same SCP, check whether the values of the aws:EC2InstanceSourcePrivateIP and aws:
VpcSourceIP condition keys are the same. Deny access if they are not.
Why: This ensures that the credentials are only used within the specific EC2 instance and its associated VPC.
Reference: AWS documentation on Service Control Policies (SCPs).
This corresponds to Option B: Create an SCP that checks whether the values of the aws: EC2InstanceSourceVPC and aws:SourceVpc condition keys are the same. Deny access if the values are not the same. In the same SCP check, check whether the values of the aws:EC2InstanceSourcePrivateIP and aws:
VpcSourceIP condition keys are the same. Deny access if the values are not the same. Apply the SCP to the OU.

QUESTION NO: 32
ある企業がAWS上でウェブ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はAWS CodeConnectionsを使用してGitリポジトリにアクセスします。この企業はAWS CodePipelineにパイプラインを設定し、コードをメインブランチにプッシュすると、アプリケーションが自動的にビルドされ、ステージング環境にデプロイされます。しかし、パイプラインに自動テスト機能が統合されていないため、メインブランチでバグや統合の問題が発生することがあります。
同社は、Gitリポジトリでコードがマージされた際に自動的にテストを実行し、テストが失敗した場合はステージング環境へのデプロイメントを阻止したいと考えています。テストは最大20分間実行できます。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. パイプラインにAWS CodeBuildアクションを追加します。テストを実行するコマンドを定義するbuildspec.ymlファイルをGitリポジトリに追加します。テストが失敗した場合にデプロイを停止するようにパイプラインを設定します。
B. Git Webhookを設定して、コードのマージごとにAWS Lambda関数を起動します。Lambda関数はプログラムでテストを実行し、テストが失敗した場合はパイプラインを停止するように設定します。
C. テスト環境のDockerイメージを使用するようにAWS Batchを設定します。AWS Batchをパイプラインに統合します。バッチジョブを送信し、テストが失敗した場合にコードマージを元に戻すAWS Lambda関数をパイプラインに追加します。
D. Gitリポジトリを設定し、コードのマージごとにAmazon S3バケットにコードをプッシュします。S3イベント通知を使用してテストを開始し、テストが失敗した場合はコードマージを元に戻します。
Answer: A
Explanation:

AWS CodePipeline supports multiple stages including source, build, test, and deploy. The most efficient way to integrate automated testing is by adding an AWS CodeBuild action to the pipeline that runs the tests using a buildspec.yml file. CodeBuild can be configured to fail the pipeline automatically if tests fail, ensuring that deployments do not proceed to the staging environment. This pattern is directly supported and documented in AWS CodePipeline + CodeBuild CI/CD architecture guidance.

QUESTION NO: 33
ある企業が午前1時のポリシーを見直しています。DevOpsエンジニアが作成したポリシーの1つが、制限が緩すぎるとして(遅延)されました。このポリシーは、週末にEnvironment: NonProduccionタグが付けられたAmazon EC2インスタンスに停止コマンドを発行するAWS Lambda関数で使用されています。現在のポリシーは次のとおりです。
少なくとも許可のポリシーを実現するために、エンジニアはどのような変更を加える必要がありますか? (3 つ選択してください)。
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
E. オプションE
F. オプションF
Answer: A B D
Explanation:
The engineer should make the following changes to achieve a policy of least permission:
A: Add a condition to ensure that the principal making the request is an AWS Lambda function. This ensures that only Lambda functions can execute this policy.
B: Narrow down the resources by specifying the ARN of EC2 instances instead of allowing all resources. This ensures that the policy only affects EC2 instances.
D: Add a condition to ensure that this policy only applies to EC2 instances tagged with "Environment:
NonProduction". This ensures that production environments are not affected by this policy.
AWS Identity and Access Management (IAM) - AWS Documentation
Certified DevOps Engineer - Professional (DOP-C02) Study Guide (page 179)

QUESTION NO: 34
ある企業は、AWS CloudFormation
スタックを使用してアプリケーションに更新をデプロイします。スタックはさまざまなリソースで構成されます。リソースには、AWS Auto Scaling グループ、Amazon EC2
インスタンス、Application Load Balancer
(ALB)、および独立したスタックの起動と維持に必要なその他のリソースが含まれます。CloudFormation スタックの更新以外のアプリケーション リソースへの変更は許可されません。
同社は最近、AWS CLI を使用してアプリケーション
スタックを更新しようとしました。スタックは更新に失敗し、次のエラー
メッセージが生成されました:「エラー: デプロイメントと CloudFormation
スタックのロールバックの両方に失敗しました。次のリソースの更新に失敗したため、デプロイメントに失敗しました: [AutoScalingGroup]」。スタックは

UPDATE_ROLLBACK_FAILED のステータスのままになります。*
この問題を解決できるのはどのソリューションですか?
A. ALB に構成されているサブネット マッピングを更新します。aws Cloudformation update-stack-set AWS CLI コマンドを実行します。
B. スタックを更新するために必要な権限を付与して、1AM ロールを更新します。aws Cloudformation continue-update-rollback AWS CLI コマンドを実行します。
C. アカウントの EC2 インスタンス数のクォータ増加リクエストを送信します。aws Cloudformation cancel-update-stack AWS CLI コマンドを実行します。
D. Auto Scaling グループリソースを削除します。aws Cloudformation rollback-stack AWS CLI コマンドを実行します。
Answer: B
Explanation:
https://repost.aws/knowledge-center/cloudformation-update-rollback-failed If your stack is stuck in the UPDATE_ROLLBACK_FAILED state after a failed update, then the only actions that you can perform on the stack are the ContinueUpdateRollback or DeleteStack operations.

QUESTION NO: 35
会社の DevOps チームは、AWS Organizations 内の組織内にある一連の AWS アカウントを管理しています。会社には、すべての Amazon EC2 インスタンスが DevOps チームが管理する承認済みの AMI
を使用するようにするソリューションが必要です。また、このソリューションでは、承認されていない AMI
の使用を修正する必要もあります。個々のアカウント管理者は、承認済みの AMI
を使用するための制限を削除できないように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. AWS CloudFormation StackSets を使用して、各アカウントに Amazon EventBridge ルールをデプロイします。Amazon EC2 の AWS CloudTrail イベントに反応し、Amazon Simple Notification Service (Amazon SNS)
トピックに通知を送信するようにルールを設定します。DevOps チームを SNS
トピックにサブスクライブします。
B. AWS CloudFormation StackSets を使用して、approved-amis-by-id AWS Config マネージドルールを各アカウントにデプロイします。承認された AMI
のリストを使用してルールを設定します。非準拠の EC2 インスタンスに対して AWS-StopEC2Instance AWS Systems Manager Automation
ランブックを実行するようにルールを設定します。
C. Amazon EC2 の AWS CloudTrail イベントを処理する AWS Lambda 関数を作成します。Amazon Simple Notification Service (Amazon SNS)
トピックに通知を送信するように Lambda 関数を設定します。DevOps チームを SNS
トピックにサブスクライブします。組織内の各アカウントに Lambda
関数をデプロイします。各アカウントに Amazon EventBridge
ルールを作成します。Amazon EC2 の AWS CloudTrail イベントに反応し、Lambda
関数を呼び出すように EventBridge ルールを設定します。
D. 組織全体で AWS Config を有効にします。承認された AMI のリストを含む、承認された -

amis-by-id AWS Config
マネージドルールを使用する適合パックを作成します。適合パックを組織全体にデプロイします。準拠していない EC2 インスタンスに対して AWS-StopEC2Instance AWS Systems Manager Automation ランブックを実行するようにルールを設定します。

Answer: D

Explanation:
Enable AWS Config Across the Organization:
AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. It can be used to assess, audit, and evaluate the configurations of your resources.
Enabling AWS Config across the organization ensures that all accounts are monitored for compliance.
Create a Conformance Pack Using the approved-amis-by-id AWS Config Managed Rule:
A conformance pack is a collection of AWS Config rules and remediation actions that can be easily deployed across an organization.
The approved-amis-by-id managed rule checks whether running instances are using approved AMIs.
Deploy the Conformance Pack Across the Organization:
Deploying the conformance pack across the organization ensures that all accounts adhere to the policy of using only approved AMIs.
The conformance pack can be deployed via the AWS Management Console, CLI, or SDKs.
Configure the Rule to Run the AWS-StopEC2Instance AWS Systems Manager Automation Runbook for Non- Compliant EC2 Instances:
The AWS-StopEC2Instance runbook can be configured to automatically stop any EC2 instances that are found to be non-compliant (i.e., not using approved AMIs).
This remediation action ensures that any unauthorized instances are promptly stopped, enforcing the policy without manual intervention.
By following these steps, the solution ensures that all EC2 instances across the organization use approved AMIs, and any non-compliant instances are remediated automatically.
References:
AWS Config Conformance Packs
AWS Config Managed Rules
AWS Systems Manager Automation Runbooks