

# ExamPrepAway

ExamPrepAway

> Contact Us   Login / Register   Search...

- HOME
- ALL VENDORS
- ★ GUARANTEE
- ? FAQ
- TESTIMONIALS
- CART (0)



Try **Online Engine** before you buy

We're not the only ones **happy** about ExamPrepAway Practice Material ...

56295+ customers in 100+ countries use ExamPrepAway Test Engine. Meet our customers.



<http://www.examprepaway.com/>

Latest Exam Guide & Learning Materials

**Exam** : **AWS-DevOps-Engineer-Professional-JPN**

**Title** : **AWS Certified DevOps Engineer - Professional (DOP-C01 日本語版)**

**Vendor** : **Amazon**

**Version** : **DEMO**

**QUESTION NO: 1**

DevOps エンジニアは、AWS CodePipeline と AWS CodeBuild を使用して、サーバーレスアプリケーションの継続的なデプロイパイプラインを構築しています。ソース、ビルド、テストの各ステージは作成済みで、デプロイステージが残っています。同社は、指定されたサブセットの顧客にデプロイし、すべての顧客に完全にリリースする前に監視することで、デプロイが失敗するリスクを軽減したいと考えています。

これらの要件を満たすには、デプロイ ステージをどのように構成する必要がありますか？

**A. AWS CloudFormation**

を使用して、スタックの更新ごとに新しいバージョンを公開します。次に、開発者が新しいバージョンをテストして承認できるように、CodePipeline 承認アクションを設定します。最後に、CodePipeline 呼び出しアクションを使用して、AWS Lambda 関数を更新し、本番環境エイリアスを使用します。

**B. CodeBuild を使用して AWS CLI で AWS Lambda**

関数コードを更新し、関数の新しいバージョンを公開して、プロダクションエイリアスを更新して関数の新しいバージョンを指すようにします。

**C. AWS CloudFormation を使用してサーバーレスアプリケーションを定義し、AWS CodeDeploy を使用して DeploymentPreference: Canary10Percent15Minutes を使用して AWS Lambda 関数をデプロイします。**

**D. AWS CloudFormation**

を使用して、スタックの更新ごとに新しいバージョンを公開します。AWS::Lambda::Alias リソースの RoutingConfig プロパティを使用して、スタックの更新中にトラフィックルーティングを更新します。

**Answer: C**

Explanation:

<https://docs.aws.amazon.com/codedeploy/latest/userguide/deployment-configurations.html>

**QUESTION NO: 2**

ある企業が、世界中の顧客向けに AWS

上で新しいモバイルゲームを展開しています。開発チームは AWS Code

サービスを使用しており、次の要件を満たす必要があります。

- クライアントはバックエンドからリアルタイムの再生データを送受信する必要がある頻繁に、最小限の遅延で

- ゲームデータはデータ保存要件を満たす必要があります

DevOps エンジニアは、ニーズを満たすためにどの戦略を実装できますか？

**A. バックエンドアプリケーションを複数のリージョンにデプロイします。コード**

リポジトリが更新されると、2 段階のビルドおよびデプロイ

パイプラインがトリガーされます。1 つのリージョンでデプロイが成功すると、AWS

Lambda 関数が呼び出され、ビルド成果物が別のリージョンの Amazon S3

バケットにコピーされます。成果物がコピーされると、新しいリージョンでデプロイ

パイプラインがトリガーされます。

**B. バックエンド**

アプリケーションを単一のリージョン内の複数のアベイラビリティゾーンにデプロイしま

す。Amazon CloudFront

ディストリビューションを作成して、アプリケーションのバックエンドを世界中の顧客に提供します。コード リポジトリが更新されると、2 段階のビルドおよびデプロイパイプラインがトリガーされます。パイプラインは、バックエンドアプリケーションをすべてのアベイラビリティゾーンにデプロイします。

**C.** バックエンド アプリケーションを複数のリージョンにデプロイします。AWS Direct Connect を使用して、アプリケーション バックエンドを世界中の顧客に提供します。コード リポジトリが更新されると、リージョンで 2 段階のビルドおよびデプロイパイプラインがトリガーされます。リージョンでのデプロイが成功すると、パイプラインはアーティファクトを別のリージョンにデプロイし続けます。

**D.** バックエンド アプリケーションを複数のリージョンにデプロイします。コード リポジトリが更新されると、リージョンで 2 段階のビルドおよびデプロイパイプラインがトリガーされます。リージョンでのデプロイが成功すると、パイプラインは別のリージョンのパイプラインを呼び出して、ビルド成果物の場所を渡します。パイプラインは成果物の場所を使用して、新しいリージョンにアプリケーションをデプロイします。

**Answer: A**

Explanation:

<https://docs.aws.amazon.com/codepipeline/latest/userguide/integrations-action-type.html#integrations-invoke>

### QUESTION NO: 3

あなたの会社では Web アプリケーションを開発し、静的 Web サイトホスティング用に構成された Amazon S3 バケットでそれをホストしています。アプリケーションはブラウザで AWS SDK for JavaScript を使用して、Amazon DynamoDB テーブルに保存されているデータにアクセスします。

DynamoDB のデータにアクセスするための API

キーが安全に保たれていることをどのように確認できますか？

**A.** 特定の DynamoDB テーブルへのアクセス権を持つ Amazon S3 ロールを IAM に作成し、Web サイトをホストしているバケットに割り当てます。

**B.** アプリケーションがアクセスを照会できるように、Web サイトをホストするバケットの AWS アクセスキーを使用して S3 バケット タグを設定します。

**C.** IAM 内で Web ID フェデレーション ロールを設定して、正しい DynamoDB リソースへのアクセスを有効にし、一時的な認証情報を取得します。

**D.** アプリケーション内のグローバル変数に AWS

キーを保存し、リクエストを行うときにこれらの認証情報を使用するようにアプリケーションを構成します。

**Answer: C**

### QUESTION NO: 4

RDS の展開を拡張する必要があります。ログによると、書き込みが 10%、読み取りが 90% で動作しています。これを簡単な方法で拡張するにはどうすればよいでしょうか。

**A.** 2 番目のマスター RDS インスタンスを作成し、RDS グループをピアリングします。

**B.** CloudFront を使用して、読み取り側のすべてのデータベース応答をキャッシュします。

C. 負荷は主に読み取りであるため、RDS の読み取りレプリカを作成します。

D. マルチ AZ RDS

インストールを作成し、読み取りトラフィックをスタンバイにルーティングします。

**Answer: C**

Explanation:

The high-availability feature is not a scaling solution for read-only scenarios; you cannot use a standby replica to serve read traffic. To service read-only traffic, you should use a Read Replica.

For more information, see Working with PostgreSQL, MySQL, and MariaDB Read Replicas.  
<http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

### QUESTION NO: 5

開発チームは、AWS CodeCommit

を使用してアプリケーションコードのバージョン管理を行い、AWS CodePipeline

を使用してソフトウェアのデプロイメントを調整しています。チームは、パイプラインがコード変更を統合するためのトリガーとして、リモートマスターブランチを使用することを決定しました。開発者はコード変更を CodeCommit リポジトリにプッシュしましたが、10分経ってもパイプラインが反応しないことに気付きました。

この問題をトラブルシューティングするには、次のどのアクションを実行する必要がありますか？

A. マスターブランチがパイプラインをトリガーするための Amazon CloudWatch Events ルールが作成されていることを確認します。

B. CodePipeline サービスロールに CodeCommit リポジトリにアクセスする権限があることを確認します。

C. 開発者の IAM ロールに CodeCommit リポジトリにプッシュする権限があることを確認します。

D. Amazon CloudWatch Logs の CodeCommit エラーが原因でパイプラインの起動に失敗したかどうかを確認します。

**Answer: C**

### QUESTION NO: 6

ある会社では、Amazon Route 53、AWS Elastic Beanstalk、Amazon RDS

を使用する単一の AWS

アカウントで本番環境のアプリケーションワークロードを実行しています。セキュリティインシデントが発生した場合、セキュリティチームはアプリケーションワークロードを新しい AWS

アカウントにフェイルオーバーしたいと考えています。また、セキュリティチームは、フォレンジック分析中は元のアカウントへのすべてのアクセスを直ちにブロックし、元の AWS アカウントの AWS リソースにアクセスできないようにしたいと考えています。

セキュリティ インシデントが発生する前に 2

番目のアカウントにフェイルオーバーする準備をする最もコスト効率の高い方法は何ですか？

A. Amazon Route 53 設定を専用の AWS アカウントに移行します。Elastic Beanstalk 設定を別のアカウントにミラーリングします。別のアカウントで RDS データベースのリードレプリカを有効にします。

**B.** Amazon Route 53 設定を専用の AWS アカウントに移行します。Elastic Beanstalk 設定ファイルを別の AWS アカウントに保存/コピーします。RDS データベースのスナップショットを別のアカウントにコピーします。

**C.** インシデント発生後に別の AWS アカウントで使用するために、Amazon Route 53 設定を保存/コピーします。

Elastic Beanstalk 設定ファイルを別のアカウントに保存/コピーします。別のアカウントで RDS データベースの読み取りレプリカを有効にします。

**D.** インシデント発生後に別の AWS アカウントで使用するために、Amazon Route 53 設定を保存/コピーします。

Elastic Beanstalk の設定を別のアカウントにミラーリングします。RDS データベースのスナップショットを別のアカウントにコピーします。

**Answer: B**

Explanation:

<https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/environment-configuration-savedconfig.html>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/hosted-zones-migrating.html>

### QUESTION NO: 7

DevOps エンジニアは、Amazon Route 53、Application Load Balancer、Auto Scaling、Amazon DynamoDB

を使用してデプロイされる新しいアプリケーションを立ち上げています。この立ち上げの重要な要件の 1

つは、突然の負荷増加に対応できるようにアプリケーションを拡張できることです。使用量が少ない期間中は、インフラストラクチャ

コンポーネントをスケールダウンしてコストを最適化する必要があります。

DevOps エンジニアは要件を満たすためにどのような手順を踏めますか? (2 つ選択してください。)

**A.** AWS Trusted Advisor を使用して、インフラストラクチャで使用される Amazon EC2 インスタンスの制限増加リクエストを送信します。

**B.** AWS Trusted Advisor を活用して、どの Amazon EC2 インスタンスの制限を引き上げる必要があるかを判断し、それらの制限を引き上げるリクエストを AWS サポートに送信します。

**C.** アプリケーションで使用される DynamoDB テーブルの Auto Scaling を有効にします。

**D.** 現在の負荷に基づいてターゲット グループを自動的に調整するように Application Load Balancer を設定します。

**E.** Auto Scaling グループの現在の使用状況を追跡するために 5 分ごとに実行される Amazon CloudWatch Events

のスケジュールされたルールを作成します。使用量が変わった場合は、スケールアップイベントをトリガーして容量を調整します。

DynamoDB の読み取りおよび書き込み容量についても同様の操作を行います。

**Answer: BC**

Explanation:

D is wrong because Auto Scaling can terminate and replace any instances that are reported as unhealthy not ALB.

<https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/>

### QUESTION NO: 8

ある会社では、Amazon ECS を使用してアプリケーションをデプロイしており、データは Amazon DynamoDB

テーブルに保存されています。この会社では、災害復旧シナリオでアプリケーションを別のリージョンにフェイルオーバーしたいと考えています。また、アプリケーションは、偶発的なデータ損失イベントから効率的に回復する必要があります。アプリケーションの RPO は 1 時間、RTO は 2 時間です。

DevOps エンジニアはどの高可用性ソリューションを推奨すべきでしょうか？

**A.** 既存の DynamoDB テーブルの設定を変更します。これをグローバルテーブルとして有効にし、使用する 2 番目のリージョンを指定します。

DynamoDB ポイントインタイムリカバリを有効にします。

**B.** テーブルの DynamoDB ストリームを有効にし、2 番目のリージョンの S3 バケットにストリームデータを書き込む AWS Lambda 関数を作成します。

AWS Data Pipeline

を使用してデータベースをフェイルオーバーリージョンに復元するジョブを 2 時間ごとにスケジュールします。

**C.** AWS Data Pipeline を使用して、2 時間ごとに DynamoDB テーブルを 2 番目のリージョンの Amazon S3 バケットにエクスポートします。

2 番目のリージョンで Data Pipeline を使用して、S3 からのエクスポートを 2 番目の DynamoDB テーブルに復元します。

**D.** AWS DMS を使用して、1 時間ごとにデータを複製します。元の DynamoDB テーブルをソースとして設定し、新しい DynamoDB テーブルをターゲットとして設定します。

**Answer: B**

### QUESTION NO: 9

ある会社では、Amazon EC2

インスタンスを使用して新しいアプリケーションを導入しています。この会社では、1 つのツールまたはサービスを使用してクエリできる集中型のアプリケーションと Amazon API ログを維持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.** Amazon CloudWatch エージェントを使用して、Amazon EC2 インスタンスから CloudWatch にログを送信します。

AWS CloudTrail を構成して API ログを CloudWatch に配信し、Amazon Athena を使用して CloudWatch の両方のログセットをクエリします。

**B.** Amazon CloudWatch エージェントを使用して、Amazon EC2 インスタンスから CloudWatch にログを送信します。

これらのログを Amazon S3 に送信するには、Amazon Kinesis Data Firehouse ロググループサブスクリプションを設定します。

AWS CloudTrail を使用して API ログを Amazon S3 に配信します。

Amazon Athena を使用して、Amazon S3 の両方のログセットをクエリします。

**C.** Amazon CloudWatch エージェントを使用して、Amazon EC2 インスタンスから Amazon

Kinesis にログを送信します。

API ログを Kinesis に配信するように AWS CloudTrail を設定します。

Amazon を使用してデータを Amazon Redshift にロードし、Amazon Redshift を使用して両方のログセットをクエリします。

**D.** Amazon CloudWatch エージェントを使用して、Amazon EC2 インスタンスから Amazon S3 にログを送信します。

Amazon CloudTrail を使用して API ログを Amazon S3 に配信し、Amazon Redshift を使用して Amazon S3 の両方のログセットをクエリします。

**Answer: D**

### QUESTION NO: 10

ある会社では、Apache ウェブサーバーを使用した Java-Apache Tomcat アプリケーションのアプリケーション デプロイメントを自動化するために、AWS CodeDeploy

を導入しています。開発チームは概念実証から開始し、開発者環境用のデプロイメントグループを作成し、アプリケーション内で機能テストを実行しました。完了後、チームはステージングと本番環境用の追加のデプロイメントグループを作成します。現在のログレベルは Apache

設定内で設定されていますが、チームはデプロイメントの発生時にこの設定を動的に変更して、グループごとに異なるアプリケーション リビジョンを持たずに、デプロイメントグループに応じて異なるログレベル設定を設定できるようにしたいと考えています。管理オーバーヘッドを最小限に抑え、展開グループごとに異なるスクリプトバージョンを必要とせずに、これらの要件を満たすにはどうすればよいでしょうか。

**A.** デプロイメントグループに応じて Amazon EC2 インスタンスにタグを付けます。次に、メタデータ サービスと EC2 API を呼び出してインスタンスが属するデプロイメントグループを識別するスクリプトをアプリケーションリビジョンに配置します。この情報を使用して、ログレベル設定を構成します。appspec.yml ファイルの Afterinstall ライフサイクルフックの一部としてスクリプトを参照します。

**B.** CodeDeploy 環境変数 DEPLOYMENT\_GROUP\_NAME を使用して、インスタンスが属するデプロイメントグループを識別するスクリプトを作成します。この情報を使用して、ログレベル設定を構成します。このスクリプトを appspec.yml ファイルの BeforeInstall ライフサイクルフックの一部として参照します。

**C.** 各環境の CodeDeploy カスタム環境変数を作成します。次に、この環境変数をチェックしてインスタンスが属するデプロイメントグループを識別するスクリプトをアプリケーションリビジョンに配置します。この情報を使用して、ログレベル設定を構成します。このスクリプトを appspec.yml ファイルの ValidateService ライフサイクルフックの一部として参照します。

**D.** CodeDeploy 環境変数 DEPLOYMENT\_GROUP\_ID を使用して、インスタンスが属するデプロイメントグループを識別し、ログレベル設定を構成するスクリプトを作成します。このスクリプトを、appspec.yml ファイルのインストール ライフサイクルフックの一部として参照します。

**Answer: B**

**QUESTION NO: 11**

DevOps

エンジニアが会社用のアプリケーションを開発しています。アプリケーションは、Amazon S3

にファイルを保持する必要があります。アプリケーションは、会社が定義するさまざまなセキュリティ分類のファイルをアップロードする必要があります。これらの分類には、機密、非公開、公開が含まれます。機密分類のファイルは、アップロードしたユーザー以外には表示できません。アプリケーションは、ユーザーの IAM ロールを使用して、S3 API 操作を呼び出します。

DevOps エンジニアは、Amazon S3

にアップロードされる各機密オブジェクトに、confidential の値を持つ DataClassification タグと、アップロードするユーザーの ID を持つ Owner タグを追加するようにアプリケーションを変更しました。

会社の要件を満たすために、DevOps

エンジニアが実行する必要がある追加の手順はどれですか？

**A.**

S3 バケットの ACL を変更して、アップロードするユーザーの IAM

ロールにバケット所有者の読み取りアクセスを許可します。aws:ResourceTag/DataClassification が confidential で、s3:ExistingObjectTag/Ownerequals \${aws:userid} の場合に、S3 バケットで s3:GetObject 操作を許可する IAM ポリシーを作成します。S3

バケットへのアクセスを必要とするユーザーの IAM ロールにポリシーをアタッチします。

**B.**

S3バケットポリシーを変更して、s3:GetObjectアクションを許可するようにする

aws:ResourceTag/DataClassificationはconfidentialであり、

s3:ExistingObjectTag/Ownerequals \${aws:userid}。S3 バケットで s3:GetObject

操作を許可する IAM ポリシーを作成します。S3

バケットへのアクセスを必要とするユーザーの IAM ロールにポリシーをアタッチします。

**C.**

S3バケットポリシーを変更して、s3:GetObjectアクションを許可するようにする

aws:ResourceTag/DataClassificationはconfidentialであり、

aws:RequesttTag/Ownerequals \${aws:userid}。S3 バケットで s3:GetObject 操作を許可する

IAM ポリシーを作成します。S3 バケットへのアクセスを必要とするユーザーの IAM

ロールにポリシーをアタッチします。

**D.**

S3バケットのACLを変更して、認証された読み取りアクセスを許可する

aws:ResourceTag/DataClassificationはconfidentialであり、

s3:ExistingObjectTag/Ownerequals \${aws:userid}。S3 バケットで s3:GetObject

操作を許可する IAM ポリシーを作成します。S3

バケットへのアクセスを必要とするユーザーの IAM ロールにポリシーをアタッチします。

**Answer: B**

**QUESTION NO: 12**

大規模で可用性の高い多層 Web アプリケーション インフラストラクチャを担当します。

このアーキテクチャは、ロードバランサーと複数の Amazon EC2 インスタンスを備えた Amazon Route53 で構成されています。

ブルー/グリーン

スタイルのデプロイメントを提供するためのプロセスを作成することが課題となっています。

この新しい要件を満たすにはどの手法を使用すればよいでしょうか？

**A. Elastic Beanstalk** を使用してアプリケーションを再デプロイし、Elastic Beanstalk デプロイメントタイプを設定してから、Amazon Route53 のエイリアスリソースレコードセットを使用して Elastic Beanstalk デプロイメントタイプを交換します。

**B. AWS CloudFormation**

テンプレートを使用してロードバランサーの背後にアプリケーションを再デプロイし、各デプロイ中に新しい AWS CloudFormation スタックを起動し、Amazon Route53 エイリアスリソースレコードセットを更新して新しいロードバランサーを指すようにし、最後に古い AWS CloudFormation スタックを終了します。

**C. Auto Scaling**

グループを使用してロードバランサーの背後にアプリケーションを再デプロイし、新しい同一の Auto Scaling グループを作成し、それをロードバランサーに関連付けます。

デプロイ中に、新しい Amazon Route53

ホストゾーンを作成し、この新しいロードバランサーをエイリアスリソースレコードセットのゾーンに追加して、古い Auto Scaling グループを削除します。

**D. OpsWorks**

スタックを使用してロードバランサーの背後にアプリケーションを再デプロイし、AWS OpsWorks スタックのバージョン管理を使用します。

デプロイ中に、アプリケーションの新しいバージョンを作成し、OpsWorks

にロードバランサーの背後で新しいバージョンを起動するように指示し、新しいバージョンが起動したら、Amazon Route53

エイリアスリソースレコードセットを更新して新しいロードバランサーを指すようにします。

**Answer: B**

### QUESTION NO: 13

ソース管理に AWS CodeCommit を使用している企業は、開発環境における AWS での継続的インテグレーションと継続的デプロイメントのパイプラインを自動化したいと考えています。この企業には 3 つの要件があります。

1. 機密情報がソース

コードを通じて漏洩しないように、コード変更には法的およびセキュリティ上のレビューが必要です。

2. すべての変更はユニットテストを経る必要があります。

3. すべての変更は、機能性を保証するために一連の機能テストを受ける必要があります。

さらに、同社では自動化に関して次のような要件を掲げています。

1. コードの変更により、CI/CD パイプラインが自動的にトリガーされます。

2. パイプラインで障害が発生した場合は、devops-admin@xyz.com に通知する必要があります。

3. テストの実行後、アセットを Amazon S3 にステージングするための承認が必要です。

CI/CD のベスト プラクティスに従いながら、これらすべての要件を満たすために DevOps

エンジニアは何をすべきでしょうか？

- A.** 開発ブランチにコミットし、開発ブランチから AWS CodePipeline をトリガーします。セキュリティレビュー、ユニットテスト、機能テスト、手動承認のために、CodePipeline で個別のステージを作成します。パイプラインステージの変更を検出するには Amazon CloudWatch メトリクスを使用し、devops-admin@xyz.com にメールを送信するには Amazon SES を使用します。
- B.** メインラインにコミットし、メインラインから AWS CodePipeline をトリガーします。セキュリティレビュー、ユニットテスト、機能テスト、手動承認のために、CodePipeline に個別のステージを作成します。AWS CloudTrail ログを使用してパイプラインステージの変更を検出し、Amazon SNS を使用して devops-admin@xyz.com にメールを送信します。
- C.** 開発ブランチにコミットし、開発ブランチから AWS CodePipeline をトリガーします。CodePipeline でセキュリティレビュー、ユニットテスト、機能テスト、手動承認のための個別のステージを作成します。パイプラインステージの変更を検出するには Amazon CloudWatch Events を使用し、devops-admin@xyz.com にメールを送信するには Amazon SNS を使用します。
- D.** メインラインにコミットし、メインラインから AWS CodePipeline をトリガーします。セキュリティレビュー、ユニットテスト、機能テスト、手動承認のために、CodePipeline に個別のステージを作成します。Amazon CloudWatch Events を使用してパイプラインステージの変更を検出し、Amazon SES を使用して devops-admin@xyz.com にメールを送信します。

**Answer: C**

#### QUESTION NO: 14

ある会社では、AWS Lambda と Amazon API Gateway を活用したサーバーレスアーキテクチャを使用する Web およびモバイルアプリケーションを構築しています。この会社は、AWS CodeCommit リポジトリの適切な環境ブランチにプッシュされるコードに基づいて、バックエンドの Lambda デプロイメントを完全に自動化したいと考えています。展開には次のものがが必要です。

- テスト用と本番用の環境パイプラインを分離します。
- テスト環境のみで実行される自動展開。

これらの要件を満たすにはどのような手順を踏む必要がありますか？

- A.** 新しい AWS CodePipeline サービスを設定します。環境ごとに CodeCommit リポジトリを作成します。適切なリポジトリからソースコードを取得するために CodePipeline を設定します。AWS CloudFormation を使用して Lambda 関数をデプロイするためのデプロイ手順を設定します。
- B.** テスト環境と本番環境用に 2 つの AWS CodePipeline 設定を作成します。本番パイプラインに手動承認ステップを設定します。環境ごとに CodeCommit リポジトリを作成します。各 CodePipeline を設定して、適切なリポジトリからソースコードを取得します。AWS CloudFormation を使用して Lambda 関数をデプロイするためのデプロイ手順を設定します。
- C.** テスト環境と本番環境用に 2 つの AWS CodePipeline 設定を作成します。本番パイプラインに手動承認ステップを設定します。環境ごとにブランチ

チを持つ 1 つの CodeCommit

リポジトリを作成します。リポジトリ内の適切なブランチからソースコードを取得するように各 CodePipeline を設定します。AWS CloudFormation を使用して Lambda 関数をデプロイするためのデプロイ手順を設定します。

**D. テスト環境と本番環境用の AWS CodeBuild**

設定を作成します。本番パイプラインに手動承認ステップを設定します。環境ごとにブランチを持つ 1 つの CodeCommit リポジトリを作成します。Lambda 関数コードを Amazon S3 バケットにプッシュします。S3 バケットから Lambda 関数をデプロイするためのデプロイ手順を設定します。

**Answer: C**

Explanation:

First, A&B both are in-correct: As a basic policy - do not create a repo for the same code for multiple environments. Always create a branch from the same repo. The strategy is wrong for A&B. Now C&D: D uses Lambda function with s3, whereas C uses code pipeline to store and build. Using code pipeline is a smart choice rather than using S3 as a code pipeline that offers better branching strategy and controls.

### QUESTION NO: 15

ある企業は、品質保証 (QA) パイプラインの一環として、AWS CloudFormation を使用してインフラストラクチャを自動的に再作成したいと考えています。QA を実行するたびに、1 つのアカウントに新しい VPC を作成し、その VPC にリソースをデプロイし、この新しいインフラストラクチャに対してテストを実行する必要があります。会社のポリシーでは、集中ログ記録を可能にするために、すべての VPC を中央管理 VPC とピアリングする必要があると規定されています。この会社には、VPC と関連リソースをデプロイするための既存の CloudFormation テンプレートがあります。どのステップの組み合わせが、自動化され、繰り返し可能な方法で目標を達成するでしょうか？

( 2 つ選択してください。 )

**A. CreateVpcPeeringConnection API 呼び出しが行われたときに Amazon CloudWatch Events ルールによって呼び出される AWS Lambda 関数を作成します。Lambda 関数は、ピアリング要求のソースを確認し、要求を受け入れ、管理 VPC のルートテーブルを更新して、トラフィックがピアリング接続を通過できるようにする必要があります。**

**B. CloudFormation テンプレートの場合:**

- カスタム リソースを呼び出して、VPC とサブネットの一意の VPC CIDR 範囲を生成します。
- 管理 VPC へのピアリング接続を作成します。
- 管理 VPC へのトラフィックを許可するようにルート テーブルを更新します。

**C. CloudFormation テンプレートの場合:**

- Fn::Cidr 関数を使用して、VPC とサブネットに未使用の CIDR 範囲を割り当てます。
- 管理 VPC へのピアリング接続を作成します。
- 管理 VPC へのトラフィックを許可するようにルート テーブルを更新します。

**D. CloudFormation テンプレートを変更して、スタックがデプロイされる各アカウントの /16 CIDR 範囲のリストを含むマッピング オブジェクトを追加します。**

**E. CloudFormation StackSets を使用して、カスタムリソースを使用して一意の CIDR**

範囲を割り当て、VPC と関連リソースを複数の AWS アカウントにデプロイします。  
各 VPC から中央管理 VPC へのピアリング接続を作成し、管理 VPC  
でそれらの接続を受け入れます。

**Answer:** AB

Explanation:

<https://blog.irdeto.com/2017/10/11/how-to-implement-vpc-peering-between-2-vpcs-in-the-same-aws-account-using-cloudformation/>

### QUESTION NO: 16

あなたの会社では AWS 上で複数のアプリケーションが実行されています。  
あなたの会社では、環境内でアラームがトリガーされたときに、オンコール  
チームに電子メールですぐに通知するツールを開発したいと考えています。  
異なるシフトで働く複数のオンコール  
チームがあり、ツールは適切な時間に適切なチームに通知を処理する必要があります。  
このソリューションをどのように実装すればよいでしょうか？

**A.** Amazon SNS トピックと Amazon SQS キューを作成します。

Amazon SQS キューを Amazon SNS トピックのサブスクライバーとして設定します。  
アラームがトリガーされたときにこのトピックに通知するように CloudWatch  
アラームを設定します。

最小インスタンス数と必要インスタンス数を両方とも 0 に設定した Amazon EC2 Auto  
Scaling グループを作成します。

このグループ内のワーカー ノードは、メッセージがキューに追加されると生成されます。  
その後、作業者は Amazon Simple Email Service  
を使用して、オンコールチームにメッセージを送信します。

**B.** Amazon SNS

トピックを作成し、オンコールチームのメールアドレスをサブスクライバーとして設定しま  
す。

AWS SDK ツールを使用して、アプリケーションを Amazon SNS  
と統合し、この新しいトピックにメッセージを送信します。

CloudWatch アラームがトリガーされると、オンコール ユーザーに通知が送信されます。

**C.** Amazon SNS

トピックを作成し、オンコールチームのメールアドレスをサブスクライバーとして設定しま  
す。

アラーム用のセカンダリ Amazon SNS

トピックを作成し、トリガーされたときにこのトピックに通知するように CloudWatch  
アラームを設定します。

アラームがトリガーされたときに HTTP POST

経由でアプリケーションに通知する、このトピックへの HTTP  
サブスクライバーを作成します。

AWS SDK ツールを使用してアプリケーションを Amazon SNS

と統合し、最初のトピックにメッセージを送信して、オンコールエンジニアがアラートを受  
信できるようにします。

**D.** オンコールグループごとに Amazon SNS

トピックを作成し、チームメンバーのメールをサブスクライバーとして設定します。  
別の Amazon SNS

トピックを作成し、トリガーされたときにこのトピックに通知するように CloudWatch アラームを設定します。

アラームがトリガーされたときに HTTP POST

経由でアプリケーションに通知する、このトピックへの HTTP サブスクライバーを作成します。

AWS SDK ツールを使用して、アプリケーションを Amazon SNS と統合し、シフト中に適切なチームトピックにメッセージを送信します。

**Answer: D**

### QUESTION NO: 17

ある会社では、特定の並列処理メカニズムを実行するソフトウェアソリューションを作成しています。このソフトウェアは、特殊なシナリオでは数十台のサーバーに拡張できます。このソリューションでは、ライセンスベースの独自のライブラリが使用されるため、個々のサーバーに 1 つの専用ライセンスがインストールされている必要があります。この会社には 200 のライセンスがあり、最大 200 のサーバー ノードを同時に実行する予定です。同社が要求した機能は、以下の通りです。

- ライセンスの使用を大規模に自動化するメカニズム。

将来、どのライセンスがいつでも利用可能かを確認するために使用するダッシュボードの作成。

これらの要件を達成するための最も効果的な方法は何ですか？

**A.** ライセンスをプライベート Amazon S3

バケットにアップロードします。ライセンスのマッピングセクションを含む AWS CloudFormation テンプレートを作成します。テンプレートで、サーバーを起動するための Auto Scaling グループを作成します。

ユーザー データ スクリプトで、マッピング

セクションから使用可能なライセンスを取得します。Auto Scaling ライフサイクルフックを作成し、インスタンスが終了した後にマッピングを更新するために使用します。

**B.** ライセンスを Amazon DynamoDB テーブルにアップロードします。Auto Scaling グループを使用してサーバーを起動する AWS CloudFormation

テンプレートを作成します。ユーザーデータ スクリプトで、DynamoDB

テーブルから使用可能なライセンスを取得します。Auto Scaling lifecycle

フックを作成し、インスタンスの終了後にマッピングを更新するために使用します。

**C.** ライセンスをプライベート Amazon S3 バケットにアップロードします。Amazon SQS キューに、S3 に保存されているライセンスのリストを入力します。Auto Scaling

グループを使用してサーバーを起動する AWS CloudFormation

テンプレートを作成します。ユーザーデータ スクリプトで、SQS

から使用可能なライセンスを取得します。Auto Scaling ライフサイクル

フックを作成し、インスタンスが終了した後にそれを使用してライセンスを SQS に戻します。

**D.** ライセンスを Amazon DynamoDB テーブルにアップロードします。AWS CLI

スクリプトを作成し、起動するインスタンスの最小数と最大数を指定した --count

パラメータを使用してサーバーを起動します。ユーザーデータ スクリプトで、DynamoDB テーブルから使用可能なライセンスを取得します。各インスタンスを監視し、障害が発生し

た場合はインスタンスを交換してから、DynamoDB テーブルを手動で更新します。

**Answer: B**

Explanation:

A dashboard to verify which licenses are available." is a key requirement. S3 and SQS won't provide you such a feature. "D" includes CLI scripts. CLI works in most cases, but it is never a right answer if an AWS native solution is available.

#### QUESTION NO: 18

EC2 EIP の範囲は何ですか？

- A. 配置グループ
- B. 可用性ゾーン
- C. 地域
- D. VPC

**Answer: C**

Explanation:

An Elastic IP address is tied to a region and can be associated only with an instance in the same region.

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/resources.html>

#### QUESTION NO: 19

ある教育会社では、Amazon ECS クラスター内の複数の Amazon EC2 インスタンスで Docker

ベースのアプリケーションを実行しています。アプリケーションの新しいバージョンをデプロイする際、開発者は新しいイメージをプライベート Docker コンテナレジストリにプッシュし、すべてのタスクを停止して開始し、すべてのタスクにアプリケーションの最新バージョンがあることを確認します。開発者は、新しいタスクが時々古いイメージで実行されていることを発見しました。

この問題を防ぐにはどうすればよいですか？

- A. 新しいイメージをプッシュした後、ECS エージェントを再起動してタスクを開始します。
- B. タスク定義の Docker イメージ タグに「latest」を使用します。
- C. 新しいイメージをプッシュするときに、タスク定義のダイジェストを更新します。
- D. Docker コンテナレジストリに Amazon ECR を使用します。

**Answer: C**

Explanation:

[https://docs.aws.amazon.com/en\\_us/AmazonECS/latest/developerguide/task\\_definition\\_parameters.html](https://docs.aws.amazon.com/en_us/AmazonECS/latest/developerguide/task_definition_parameters.html) When a new task starts, the Amazon ECS container agent pulls the latest version of the specified image and tag for the container to use. However, subsequent updates to a repository image are not propagated to already running tasks.

#### QUESTION NO: 20

ある会社では、AWS CloudFormation を使用して Web アプリケーションのインフラストラクチャを開発しています。データベースエンジニアリング チームは、CloudFormation テンプレートでデータベースリソースを管理し、ソフトウェア開発チームは、別の CloudFormation テンプレートで Web

## アプリケーション

リソースを管理しています。アプリケーションの範囲が拡大するにつれて、ソフトウェア開発チームは、データベース エンジニアリング

チームが管理するリソースを使用する必要があります。ただし、両方のチームには、維持したい独自のレビューおよびライフサイクル管理プロセスがあります。また、両方のチームで、リソースレベルの変更セットのレビューも必要です。ソフトウェア開発チームは、CI/CD パイプラインを使用して、このテンプレートへの変更をデプロイ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. データベース CloudFormation

テンプレートからスタックのエクスポートを作成し、それらの参照を Web アプリケーション CloudFormation テンプレートにインポートします。

B. CloudFormation ネストされたスタックを作成して、クロススタック リソース参照とパラメータを両方のスタックで使用できるようにします。

### C. クロススタック

リソース参照とパラメータを両方のスタックで使用できるように、CloudFormation スタック セットを作成します。

### D. Web アプリケーションの CloudFormation

テンプレートに入力パラメータを作成し、データベース スタックからリソース名と ID を渡します。

**Answer: A**

## QUESTION NO: 21

SQS にメッセージを保存できる最大時間はどれくらいですか？

A. 14日間

B. 1か月

C. 4日間

D. 7日間

**Answer: A**

Explanation:

A message can be stored in the Simple Queue Service (SQS) from 1 minute up to a maximum of 14 days.

## QUESTION NO: 22

現在のログ分析アプリケーションでは、Web アプリケーションの上位 10 人のユーザーのレポートを生成するのに 4 時間以上かかります。

この情報をリアルタイムでレポートし、レポートが常に最新であることを保証し、Web アプリケーションへのリクエスト数の増加に対応できるシステムを実装するように求められています。コスト効率が良く、要件を満たすオプションを選択してください。

### A. データを CloudWatch Logs

に公開し、オンデマンドで負荷を処理できるようにアプリケーションを自動スケーリングするように設定します。

### B. ログデータを Amazon S3 バケットに公開します。

AWS CloudFormation を使用して Auto Scaling グループを作成し、Amazon S3

に保存されているログファイルを取得するように構成された後処理アプリケーションをスケールリングします。

**C. ログデータを Amazon Kinesis**

データストリームに投稿し、ログデータを処理するように構成されたログ処理アプリケーションをサブスクライブします。

**D. Auto Scaling グループを設定して、Amazon EMR バケットのサイズを増やします。**

**E. マルチ AZ Amazon RDS MySQL クラスターを作成し、ログデータを MySQL に投稿し、マップ削減ジョブを実行して、ユーザー数に関する必要な情報を取得します。**

**Answer: C**

### QUESTION NO: 23

DevOpsエンジニアは、12のAmazon

EC2インスタンスにデプロイされたアプリケーションを操作しています。

3つのアベイラビリティゾーン。AMI

イメージから新しいインスタンスを起動できます。通常の日では、各 EC2

インスタンスの使用率は、営業時間中に 30%、営業時間外に 10% です。

CPU 使用率は、営業時間の最初の数分で急上昇します。CPU

使用率は、その後徐々に増加します。

エンジニアは、同等以上の信頼性を維持しながらコストを削減するよう求められています。

これらの要件を満たすソリューションはどれですか？

**A. 営業時間の開始前と終了前、および営業時間終了後のスケジュールを含む 2 つの**

Amazon CloudWatch Events ルールを作成します。各ルールによって 1 つずつ呼び出される

2 つの AWS Lambda 関数を作成します。最初の関数は営業時間終了後に 9

つのインスタンスを停止し、2 番目の関数は営業日が始まる前に 9

つのインスタンスを再起動する必要があります。

**B. AMI イメージを使用して Amazon EC2 Auto Scaling グループを作成し、Auto Scaling**

グループの CPU 使用率の平均に基づいて 75%

を目標とするスケールリングアクションを設定します。営業時間終了後にインスタンスの最小

数を 3 に調整し、営業時間開始前に 6

にリセットするスケジュールされたアクションをグループに作成します。

**C. 営業時間の開始前と終了前、および営業時間の終了前にスケジュールを設定する 2 つの**

Amazon CloudWatch Events

ルールを作成します。インスタンス数のパラメータを使用して、EC2 Auto Scaling

グループを作成する AWS CloudFormation

スタックを作成します。各ルールからスタックを呼び出し、午前中に 3、夕方に 6

というパラメータ値を渡します。

**D. AMI イメージを使用して EC2 Auto Scaling グループを作成し、Auto Scaling グループの**

CPU 使用率の平均に基づいて 75% を目標とするスケールリング

アクションを設定します。営業終了後の毎晩に 9 つのインスタンスを終了するスケジュール

アクションを作成します。

**Answer: B**

### QUESTION NO: 24

サーバーの不変性とは何ですか？

- A. 作成後にサーバーを更新しません。
- B. サーバー数を変更する機能。
- C. 作成後にサーバーを更新します。
- D. サーバー数を変更できない。

**Answer: A**

Explanation:

... disposable upgrades offer a simpler way to know if your application has unknown dependencies. The underlying EC2 instance usage is considered temporary or ephemeral in nature for the period of deployment until the current release is active. During the new release, a new set of EC2 instances are rolled out by terminating older instances. This type of upgrade technique is more common in an immutable infrastructure.

<https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

### QUESTION NO: 25

DevOps

エンジニアは、新しいアプリケーション用のマルチリージョン災害復旧ソリューションを支援しています。アプリケーションは、Auto Scaling グループで実行されている Amazon EC2 インスタンスと Amazon Aurora MySQL DB クラスターで構成されています。アプリケーションは、RTO 120 分、RPO 60 分で利用可能である必要があります。

これらの要件を満たす最も費用対効果の高い方法は何ですか？

**A.** 別のリージョンで Aurora レプリカとして Aurora DB クラスターを起動します。すべてのコンピューティングリソース用の AWS CloudFormation テンプレートを作成し、2 つのリージョンにスタックを作成します。

障害が発生した場合に Aurora レプリカをプライマリインスタンスに昇格するスクリプトを作成します。

**B.** 別のリージョンで Aurora レプリカとして Aurora DB クラスターを起動し、自動クロスリージョンフェイルオーバーを構成します。Auto Scaling グループを含む AWS CloudFormation テンプレートを作成し、2 つのリージョンにスタックを作成します。

災害復旧リージョンの CloudFormation スタックを更新してインスタンスの数を増やすスクリプトを作成します。

**C.** AWS Lambda を使用して、Aurora DB クラスターのスナップショットを作成し、1 時間ごとに宛先リージョンにコピーします。

Auto Scaling グループを含む AWS CloudFormation テンプレートを作成し、2 つのリージョンにスタックを作成します。

スナップショットから Aurora DB クラスターを復元し、Auto Scaling グループを更新してインスタンスの起動を開始します。

**D.** Amazon DynamoDB のクロスリージョンレプリケーションを設定します。Auto Scaling グループを含む AWS CloudFormation テンプレートを作成し、2 つのリージョンにスタックを作成します。

障害発生時に災害復旧リージョンの CloudFormation スタックを更新し、DynamoDB レプリカをプライマリインスタンスに昇格させるスクリプトを作成します。

**Answer: C**

Explanation:

[https://d1.awsstatic.com/training-and-certification/docs-devops-pro/AWS-Certified-DevOps-Engineer-Professional\\_Sample-Questions.pdf](https://d1.awsstatic.com/training-and-certification/docs-devops-pro/AWS-Certified-DevOps-Engineer-Professional_Sample-Questions.pdf)

#### QUESTION NO: 26

サービス用のパッケージをインストールし、そのパッケージの構成ファイルをシステムに配置して、サービスを再起動するタスクを含むプレイブックがあります。その後、プレイブックは 2 回続けて実行されます。

2 回目の実行で Ansible が何を実行すると予想しますか？

**A.**

古いパッケージと構成ファイルを削除し、再インストールしてからサービスを再起動します。

**B.** ターゲットホストに対してアクションを実行しません。

**C.** パッケージがインストールされているかどうかを確認し、ファイルがソースファイルと一致しているかどうかを確認します。一致していない場合は再インストールし、サービスを再起動します。

**D.** パッケージを再インストールし、ファイルをコピーしてサービスを再起動します。

**Answer: C**

Explanation:

Ansible follows an idempotence model and will not touch or change the system unless a change is warranted.

Reference: <http://docs.ansible.com/ansible/glossary.html>

#### QUESTION NO: 27

リモートターゲットホストでプレイブックを実行すると、「[Errno 13] アクセスが拒否されました:

`/home/nick/.ansible/tmp'。この問題の最も可能性の高い原因は何でしょうか？

**A.** Ansible システム上のユーザーのホームまたは `.ansible'

ディレクトリは、プレイを実行しているユーザーによって書き込み可能ではありません。

**B.** 指定されたユーザーはリモートシステムに存在しません。

**C.** `ansible-playbook' を実行するユーザーは、自分のホームディレクトリから実行する必要があります。

**D.** Ansible リモートホスト上のユーザーのホームまたは `.ansible'

ディレクトリは、プレイを実行しているユーザーによって書き込み可能ではありません。

**Answer: D**

Explanation:

Each task that Ansible runs calls a module. When Ansible uses modules it copies the module to the remote target system. In the error above it attempted to copy it to the remote user's home directory and found that either the home directory or the `.ansible' directory were not writeable and thus could not continue.

Reference: [http://docs.ansible.com/ansible/modules\\_intro.html](http://docs.ansible.com/ansible/modules_intro.html)

#### QUESTION NO: 28

AWS EBS を使用して、AWS EC2 でクラスター化された NoSQL

データベースを実行しています。データベースの応答時間のレイテンシーを削減する必要が

あります。最も重要な懸念事項は、可用性ではなくパフォーマンスです。初期設定はあなたが実行したのではなく、AWSの知識があまりない誰かが実行したため、すべてが最適に構成されたかどうかはわかりません。次のどれが、レイテンシーの増加につながる問題ではないでしょうか。

- A. EC2 インスタンスは EBS 最適化されていません。
- B. データベースと要求システムの両方が間違ったアベイラビリティゾーンにあります。
- C. EBS ボリュームは PIOPS を使用していません。
- D. データベースは配置グループ内で実行されていません。

**Answer: B**

Explanation:

For the highest possible performance, all instances in a clustered database like this one should be in a single Availability Zone in a placement group, using EBS optimized instances, and using PIOPS SSD EBS Volumes. The particular Availability Zone the system is running in should not be important, as long as it is the same as the requesting resources.

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html>

### QUESTION NO: 29

ある企業には、単一の AWS リージョンで Amazon DynamoDB

テーブルを使用してユーザー情報を保存する Web

アプリケーションがあります。ますますグローバル化するユーザーベースをサポートするには、アプリケーションをセカンダリリージョンで実行し、ユーザーが最も近いリージョンに接続してセカンダリリージョンにフェイルオーバーできるようにする必要があります。

展開がこれらの要件を満たしていることを確認するには、どのアプローチを使用する必要がありますか？

- A. リージョン間でデータをコピーするように DynamoDB ストリームを設定し、両方のリージョンに Web スタックをデプロイし、ヘルスチェック付きの地理的近接性ルーティングポリシーを使用するように Amazon Route 53 を設定します。
- B. DynamoDB テーブルをグローバルテーブルに変換し、両方のリージョンに Web スタックをデプロイし、ヘルスチェック付きの地理的近接性ルーティングポリシーを使用するように Amazon Route 53 を設定します。
- C. DynamoDB クロスリージョンバックアップを定義してデータをセカンダリリージョンにコピーし、両方のリージョンに Web スタックをデプロイし、ヘルスチェック付きのレイテンシーベースのルーティングポリシーを使用するように Amazon Route 53 を設定します。
- D. DynamoDB Accelerator を使用してデータをセカンダリリージョンにコピーし、両方のリージョンに Web スタックをデプロイし、フェイルオーバールーティングポリシーを使用するように Amazon Route 53 を設定します。

**Answer: B**

Explanation:

<https://aws.amazon.com/blogs/database/how-to-use-amazon-dynamodb-global-tables-to-power-multiregion-architectures/>

**QUESTION NO: 30**

AWS Auto Scaling グループと Auto Scaling

起動設定を使用する場合、個々のサーバーの有効期間を最短にできるデプロイメント方法はどれですか？

- A. デプロイ時にすべてのコードと構成を含む AMI を事前にバイクします。
- B. インスタンスの起動時に Dockerfile ブートストラップを使用します。
- C. UserData ブートストラップ スクリプトを使用します。
- D. AWS EC2 Run Commands を使用してフリートに動的に SSH 接続します。

**Answer: A**

Explanation:

Note that the bootstrapping process can be slower if you have a complex application or multiple applications to install. Managing a fleet of applications with several build tools and dependencies can be a challenging task during rollouts. Furthermore, your deployment service should be designed to do faster rollouts to take advantage of Auto Scaling. Prebaking is a process of embedding a significant portion of your application artifacts within your base AMI. During the deployment process you can customize application installations by using EC2 instance artifacts such as instance tags, instance metadata, and Auto Scaling groups. <https://d0.awsstatic.com/whitepapers/overview-of-deployment-options-on-aws.pdf>

**QUESTION NO: 31**

Amazon SQS を使用すると、空の受信リクエストが大量に発生します。

これにより、インスタンスに不要なネットワーク負荷が大量に発生します。

この負荷を軽減するにはどうすればよいでしょうか？

- A. 代わりにキューを SNS トピックにサブスクライブします。
- B. 短いアンケートではなく、できるだけ長いアンケートを使用してください。
- C. 可視性のタイムアウトを短く変更します。
- D. EC2 インスタンスで `sqsd` を使用します。

**Answer: B**

Explanation:

One benefit of long polling with Amazon SQS is the reduction of the number of empty responses, when there are no messages available to return, in reply to a ReceiveMessage request sent to an Amazon SQS queue. Long polling allows the Amazon SQS service to wait until a message is available in the queue before sending a response.

<http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-long-polling.html>

**QUESTION NO: 32**

以下の CloudTrail ログ

ファイルの部分を検討してください。どのタイプのイベントがキャプチャされていますか？

```
"イベント時間":"2016-07-16T17:35:32Z",  
"イベントソース":"signin.amazonaws.com",  
"イベント名":"コンソールログイン",  
"aws リージョン":"us-west-1",  
"ソースIPアドレス":"192.1.2.10",
```

...

- A. AWS コンソールのサインイン
- B. AWS ログオフ
- C. AWS エラー
- D. AWS デプロイメント

**Answer: A**

Explanation:

CloudTrail records attempts to sign into the AWS Management Console, the AWS Discussion Forums and the AWS Support Center. Note, however, that CloudTrail does not record root sign-in failures.

Reference:

<http://docs.aws.amazon.com/awsccloudtrail/latest/userguide/cloudtrail-event-reference-aws-console-sign-in-events.html>

### QUESTION NO: 33

CTO から、AWS

アカウントのすべてのユーザーがリソースを変更するために何をしているかを常に把握するように依頼されました。彼女は、できるだけ幅広いリソース タイプグループについて、誰が何を行っているかを週 1 回報告してほしいと考えています。これはどのように行うべきでしょうか。

- A. グローバル AWS CloudTrail トレイルを作成します。ログデータをレポートに集約するスクリプトを設定し、週に 1 回 S3 に公開して CTO に配信します。
- B. すべての AWS API 呼び出しにサブスクライブされた SNS トピックで CloudWatch イベント ルールを使用します。この SNS トピックの電子メール タイプの配信に CTO をサブスクライブします。
- C. AWS IAM 認証情報レポートを使用して、一定期間にわたる IAM ユーザートークンのすべての使用の CSV を CTO に提供します。
- D. Lambda の SNS サブスクリプションで AWS Config を使用し、時間の経過に伴うこれらの変更を DynamoDB テーブルに挿入します。このテーブルの内容に基づいてレポートを生成します。

**Answer: A**

Explanation:

This is the ideal use case for AWS CloudTrail.

CloudTrail provides visibility into user activity by recording API calls made on your account. CloudTrail records important information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service. This information helps you to track changes made to your AWS resources and to troubleshoot operational issues. CloudTrail makes it easier to ensure compliance with internal policies and regulatory standards.

Reference: <https://aws.amazon.com/cloudtrail/faqs/>

### QUESTION NO: 34

ある会社では、AWS CodeDeploy

を使用してアプリケーションのデプロイメントを管理しています。最近、開発チームはバー

ジョン管理に GitHub を使用することを決定し、GitHub リポジトリを CodeDeploy と統合する方法を模索しています。また、そのリポジトリに新しいコミットがあるたびにデプロイメントを自動化する方法も開発する必要があります。現在、チームは Amazon S3 の場所を手動で指定して、新しいアプリケーション リビジョンをデプロイしています。最も効率的な方法で統合を実現するにはどうすればよいでしょうか？

- A.** GitHub Webhook を作成して、リポジトリを AWS CodeCommit に複製します。CodeCommit をソースプロバイダーとして使用し、AWS CodeDeploy をデプロイメントプロバイダーとして使用する AWS CodePipeline パイプラインを作成します。設定が完了したら、GitHub リポジトリに変更をコミットして、最初のデプロイメントを開始します。
- B.** GitHub をソースプロバイダーとして使用し、AWS CodeDeploy をデプロイメントプロバイダーとして使用する AWS CodePipeline パイプラインを作成します。この新しいパイプラインを GitHub アカウントに接続し、CodePipeline に GitHub の Webhook を使用して変更が発生したときにパイプラインを自動的に開始するように指示します。
- C.** GitHub リポジトリ内に新しいコミットがあったかどうかを定期的にチェックする AWS Lambda 関数を作成します。新しいコミットが見つかった場合は、AWS CodeDeploy への CreateDeployment API 呼び出しをトリガーして、デプロイメントグループ内の最後のコミット ID に基づいて新しいデプロイメントを開始します。
- D.** AWS CodeDeploy カスタムデプロイ設定を作成し、GitHub リポジトリをデプロイグループに関連付けます。関連付けプロセス中に、デプロイグループを GitHub で認証して、GitHub セキュリティ認証トークンを取得します。新しいコミットが見つかった場合に自動的にデプロイするように、デプロイグループのオプションを設定します。最初のデプロイをトリガーするには、GitHub リポジトリに新しいコミットを実行します。

**Answer:** B

Explanation:

The team also needs to develop a way to automate deployment whenever there is a new commit on that repository.

<https://docs.aws.amazon.com/codedeploy/latest/userguide/integrations-partners-github.html#behaviors-deploy-automatically>

### QUESTION NO: 35

アプリケーションは、CloudFormation を使用してアプリケーションのリソースを調整します。アプリケーションが稼働する前のテストフェーズ中に、Amazon RDS インスタンスタイプが変更され、インスタンスが再作成され、テスト データが失われました。今後このような事態が起きないようにするにはどうすればよいでしょうか？

- A.** ユーザーが Amazon RDS インスタンスタイプを選択できる AWS CloudFormation パラメータ内で、AllowedValues に現在のインスタンスタイプのみが含まれるように設定します。
- B.** AWS CloudFormation スタックポリシーを使用して、インスタンスへの更新を拒否します。SetStackPolicy で拒否された IAM プリンシパルにのみ UpdateStack 権限を許可します。

C. AWS CloudFormation テンプレートで、AWS::RDS::DBInstance の DBInstanceClass プロパティを読み取り専用を設定します。

D. AWS CloudFormation

通知「BeforeResourceUpdate」をサブスクライブし、識別されたリソースが Amazon RDS インスタンスである場合は CancelStackUpdate を呼び出します。

E. AWS CloudFormation テンプレートで、AWS::RDS::DBInstance の DeletionPolicy プロパティの DeletionPolicy を「Retain」に設定します。

**Answer: E**