# AWS 認定 DevOps エンジニア – プロフェッショナル
# AWS Certified DevOps Engineer – Professional
# (DOP-001) 認定試験の質問例

**(1) ある企業が AWS CodeCommit を使用して、アプリケーションのソースコードを管理しています。この企業は AWS CodePipeline を使用して、このアプリケーション用の CI/CD パイプラインを作成しようとしています。CodeCommit リポジトリのマスターブランチに変更が加えられた場合に、このパイプラインを自動開始する必要があります。このアプリケーションには毎日変更が加えられるので、変更後できるだけ速やかにパイプラインを開始する必要があります。**

**これらの要件を満たすには、どうすればよいですか。**

A) リポジトリを定期的に検査するよう、パイプラインを構成する。変更が検出された場合、パイプラインを開始する。
B) 変更時に Amazon CloudWatch Events イベントを生成するよう、リポジトリを構成する。イベント生成時に開始するよう、パイプラインを構成する。
C) AWS Lambda 関数を定期的に実行するよう、リポジトリを構成する。この関数は、リポジトリを検査し、変更が検出された場合にパイプラインを開始するものである。
D) 変更時に SNS 通知を発行するよう、リポジトリを構成する。パイプラインを Amazon SNS トピックにサブスクライブする。

**(2) 開発チームが、AWS CodeCommit リポジトリをセットアップしたいと考えています。開発者は、変更内容を自分用のブランチにプッシュすることを許可する必要があります。一方、開発者がコミットをマスターブランチにプッシュすることやプル要求をマスターブランチにマージすることは禁止する必要があります。また、マスターブランチに対するコミットまたはマージが発生した場合、プロジェクトマネージャーは通知を受信する必要があります。**

**マスターブランチを保護すると同時に、できるだけ速やかにアラートを送信するには、どうすればよいですか (2 つ選択してください)。**

A) AWS IAM ポリシーを開発者の IAM グループにアタッチする。このポリシー内で、コミットをマスターブランチにプッシュするアクション、プル要求をマスターブランチにマージするアクション、およびファイルをマスターブランチに追加するアクションを拒否する。
B) リソースポリシーを CodeCommit リポジトリにアタッチする。このポリシー内で、開発者の IAM グループのメンバーに対して、コミットをマスターブランチにプッシュするアクション、プル要求をマスターブランチにマージするアクション、およびファイルをマスターブランチに追加するアクションを拒否する。
C) AWS Lambda 関数をセットアップする。この関数は、15 分ごとに動作してリポジトリに対する変更を検査し、通知を Amazon SNS トピックに発行するものである。
D) マスターブランチに対する `CodeCommit Repository State Change` イベントによってトリガされる、Amazon CloudWatch Events ルールをセットアップする。Amazon SNS トピックをターゲットとして追加する。
E) ログイベントを Amazon CloudWatch Logs に送信するよう、AWS CloudTrail を構成する。リポジトリイベントを識別するためのメトリクスフィルタを定義する。CloudWatch アラームを作成する。Amazon SNS トピックをターゲットとして追加する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(3) ある企業が AWS CodeBuild を使用してアプリケーションをビルドしています。企業ポリシーに基づき、すべてのビルドアーティファクトを暗号化して格納する必要があります。また、運用ロール権限を持つ IAM ユーザーにのみ、アーティファクトへのアクセスを許可する必要があります。**

**これらの要件を満たすには、どうすればよいですか。**

A) ビルド後処理コマンドを CodeBuild ビルド仕様に追加する。このコマンドは、ビルドオブジェクトを Amazon S3 バケットに格納するものである。バケットポリシーを作成する。その中で、要求ヘッダー内で `x-amz-server-side-encryption` が指定されていない場合、バケットにアップロードしないことを指定する。また、すべてのアクションに対して `Deny` ステートメントを指定し、`NotPrincipal` セクション内で運用 IAM グループを参照する。

B) ビルド後処理コマンドを CodeBuild ビルド仕様に追加する。このコマンドは、ビルドオブジェクトを Amazon S3 バケットに格納するものである。AWS Lambda 関数をトリガするよう、S3 イベント通知を構成する。この Lambda 関数は、オブジェクトを取得して暗号化し、タグ（キー: `encrypted`、値: `true`）を付加して S3 バケットに再格納するものである。S3 バケットポリシーを作成する。その中で、すべてのアクションに対して `Deny` ステートメントを指定し、`NotPrincipal` セクション内で運用 IAM グループを参照し、`Condition` セクション内で `Encrypted` タグを参照する。

C) ビルド後処理コマンドを CodeBuild ビルド仕様に追加する。このコマンドは、ビルドオブジェクトを、S3 デフォルト暗号化が有効化された Amazon S3 バケットに格納するものである。S3 バケットポリシーを作成する。その中で、すべてのアクションに対して `Deny` ステートメントを指定し、`NotPrincipal` セクション内で運用 IAM ロールを参照する。

D) ビルド後処理コマンドを CodeBuild ビルド仕様に追加する。このコマンドは、AWS KMS の `Encrypt` API を呼び出してアーティファクトを AWS KMS に渡し、指定顧客マスターキー (CMK) を使用して暗号化するものである。暗号化されたアーティファクトを Amazon S3 バケットに格納する。AWS KMS で、運用 IAM グループをこの CMK に対する唯一のユーザーとして設定する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(4) DevOps エンジニアが、アプリケーションを AWS に展開するための、ブルー/グリーンデプロイメントプロセスを実装したいと考えています。また、トラフィックのルーティング先環境を徐々に切り替えられるようにしたいと考えています。このアプリケーションは、Application Load Balancer の内側にある Amazon EC2 インスタンス上で動作します。このインスタンスは、EC2 Auto Scaling グループ内で動作します。データは、Amazon RDS マルチ AZ データベースインスタンスに格納されます。外部 DNS レコードは、Amazon Route 53 から提供されます。**

**このブルー/グリーンデプロイメントプロセスを実装するには、どうすればよいですか (3 つ選択してください)。**

A) 2 つ目の Auto Scaling グループを同じ Application Load Balancer の内側に作成する。
B) 2 つ目の Application Load Balancer と Auto Scaling グループを作成する。
C) Route 53 を使用して、新しい環境をルーティング先とする 2 つ目のエイリアスレコードを作成する。2 つのレコード間でフェイルオーバールーティングポリシーを使用する。
D) Route 53 を使用して、新しい環境をルーティング先とする 2 つ目のエイリアスレコードを作成する。2 つのレコード間で加重ルーティングポリシーを使用する。
E) 同じ RDS データベースインスタンスを使用するよう、新しい EC2 インスタンスを構成する。
F) RDS データベースインスタンスのフェイルオーバーノードを使用するよう、新しい EC2 インスタンスを構成する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(5) DevOps エンジニアが、AWS Lambda 関数を記述し、この Lambda 関数を AWS CloudFormation テンプレートスニペット内で指定し（下記参照）、このテンプレートを Amazon S3 バケットに格納しました。**

```
MyLambdaFunctionV1:
  Type:"AWS::Lambda::Function"
  Properties:
    Handler:"index.handler"
    Role:"arn:aws:iam::515290864834:role/AccountScanner"
    Code:
      S3Bucket:"johndoe-com-lambda-source"
      S3Key:"AccountScanner.zip"
    Runtime:"dotnetcore2.1"
    Timeout:60
```

**CloudFormation スタックが作成され、Lambda 関数が想定どおりに動作しています。DevOps エンジニアは、関数コードの新バージョンを入手したので、スタック更新後すぐに新バージョンが実行されるようにしたいと考えています。**

**この要件を満たすには、どのように展開すればよいですか（3 つ選択してください）。**

A) CloudFormation テンプレート内の Lambda 関数の論理名を `MyLambdaFunctionV1` から `MyLambdaFunctionV2` に変更し、CloudFormation スタックを更新する。

B) 既存の S3 バケットにおいてバージョニングを有効化する。新しいコードを既存の S3 バケットにアップロードする。CloudFormation テンプレート内の Lambda 関数の `S3ObjectVersion` プロパティで、S3 オブジェクトのバージョン ID を指定し、その後 CloudFormation スタックを更新する。

C) AWS SAM を使用して、`sam deploy` コマンドを CloudFormation テンプレートに対して発行し、Lambda 関数のバージョンを更新する。

D) CloudFormation テンプレート内の Lambda 関数の S3 バケットプロパティ値を、別のバケット場所を指すよう変更する。新しいコードを新しい S3 バケット場所にアップロードする。CloudFormation スタックを更新する。

E) CloudFormation テンプレート内の Lambda 関数の `S3Key` プロパティ値を、別の場所および名前の .zip ファイルを指すよう変更する。新しいコードを新しい S3 バケット場所にアップロードする。その際、.zip ファイルの場所と名前を変更したことに留意する。その後、CloudFormation スタックを更新する。

F) サーバーレスフレームワークを使用して、`serverless deploy function -f MyLambdaFunctionV1` コマンドを発行し、既存の Lambda 関数を更新する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(6) DevOps エンジニアが、ある企業におけるセキュリティ適合作業を自動化するように要求されています。この企業は、カスタム AWS Config ルールを作成し、不適合セキュリティ構成を検出しています。また、不適合問題が検出された場合、問題を自動修復し、社内セキュリティメッセージチャネルを使用してセキュリティチームに通知したいと考えています。メッセージボードには REST インターフェイスが備わっており、HTTPS POST 要求のボディがこのチャネル経由で発行されます。**

**最も費用対効果の高い方法でこれらの要件を満たすには、どうすればよいですか (3 つ選択してください)。**

A) 構成項目変更通知を Amazon SNS トピックに発行するための Amazon CloudWatch Events ルールを作成する。
B) 適合状況変更通知を Amazon SNS トピックに発行するための Amazon CloudWatch Events ルールを作成する。
C) 構成項目変更通知を Amazon SNS トピックに発行するよう、AWS Config を構成する。
D) Amazon API Gateway を使用して RESTful API を作成し、AWS を AWS Config に統合する。
この API を Amazon SNS トピックにサブスクライブする。
E) メッセージチャネルの HTTPS エンドポイントを Amazon SNS トピックにサブスクライブする。
F) 不適合セキュリティ構成を処理するための AWS Lambda 関数を記述する。この関数を Amazon SNS トピックにサブスクライブする。

**(7) ある企業が、Amazon Linux AMI の最新バージョンが動作している Amazon EC2 インスタンス上で、アプリケーションを実行しています。サーバー管理者は、新しいセキュリティ修正プログラムを適用する際、適用先インスタンスをサービスから手動で削除し、修正プログラムを適用し、その後、それらのインスタンスをサービスに戻します。新しい企業セキュリティポリシーに基づき、セキュリティ修正プログラムがリリースされた場合、7 日以内に適用する必要があります。セキュリティチームは、すべてのインスタンスが適合状態であることを検証する必要があります。修正プログラム適用作業は、ユーザーへの影響が最も小さい時間帯に実行する必要があります。**

**セキュリティポリシー適合作業を自動化するには、どうすればよいですか。**

A) SSH を使用して修正プログラムをすべてのマシンにダウンロードして適用するよう、AWS CodeBuild プロジェクトを構成する。Amazon CloudWatch Events スケジュール化イベントを使用して、メンテナンスウィンドウ中に CodeBuild プロジェクトを実行する。
B) AWS Systems Manager Patch Manager を使用して、修正プログラムベースラインを作成する。EC2 インスタンス上でスクリプトを作成する。このスクリプトは、CLI を使用して、Patch Manager から最新の修正プログラムを取得するものである。クローンジョブを作成し、メンテナンスウィンドウ中にスクリプトを実行するようスケジューリングする。
C) スクリプトを作成する。このスクリプトは、未適用のセキュリティ修正プログラムがある場合に適用するものである。クローンジョブを作成し、メンテナンスウィンドウ中にスクリプトを実行するようスケジューリングする。スクリプトとクローンジョブをアプリケーション AMI にインストールし、アプリケーションを再展開する。
D) すべてのアプリケーション EC2 インスタンスを修正プログラムグループに登録する。AWS Systems Manager Patch Manager を使用して、修正プログラムベースラインを作成する。修正プログラムベースラインを適用するよう、メンテナンスウィンドウを構成する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(8) 運用担当者が、AWS 上でレガシー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はモノリシック Microsoft Windows プログラムであり、1 個の Amazon EC2 インスタンス上で動作しています。このアプリケーションのソースコードは入手できないので、このアプリケーションを修正することはできません。インスタンスのメモリ使用率が 90% を上回ると、このアプリケーションにメモリリークと障害が発生します。運用担当者は、EC2 インスタンス上で統合 Amazon CloudWatch エージェントを構成し、メモリ使用率に関するパフォーマンスモニタカウンタを収集しています。**

**アプリケーション障害を回避するには、どうすればよいですか (2 つ選択してください)。**

A) Amazon CloudWatch Events イベントを作成し、メモリ使用率が 80% を上回ったときに Amazon SNS トピックに発行する。
B) Amazon CloudWatch Logs で、メモリ使用率に関するメトリクスフィルタを作成する。メモリ使用率フィルタに対する CloudWatch アラームを作成し、メモリ使用率が 80% を上回ったときに Amazon SNS トピックに発行する。
C) メモリ使用率メトリクスに対する CloudWatch アラームを作成し、メモリ使用率が 80% を上回ったときに Amazon SNS トピックに発行する。
D) AWS Lambda 関数を Amazon SNS トピックにサブスクライブする。この関数は、AWS Systems Manager の Run コマンドを使用してアプリケーションを再起動するものである。
E) EC2 インスタンスを Amazon SNS トピックにサブスクライブする。アプリケーションを再起動するスクリプトを実行する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com 

**(9) ある企業が、100 個以上の内部アプリケーションを AWS に移行しようとしています。これらのアプリケーションの間に依存関係はありませんが、すべてのアプリケーションにおいて、似たような社内標準アーキテクチャが使用されています。これらのアーキテクチャの主な要素のうち、違いがあるものは次のとおりです。**

- **Web 層とアプリケーション層の両方を備えたアプリケーションもあれば、Web 層しかないアプリケーションもある。**
- **データベースが存在する場合、そのデータベースは MySQL、SQL Server、または PostgreSQL である。(この企業は、すべてのデータベースを Amazon RDS で管理する予定である。)**
- **LAMP スタックをベースにしたアプリケーションもあれば、.NET スタックをベースにしたアプリケーションもある。**

**DevOps チームは、各アプリケーションチームがインフラストラクチャを起動して各々のアプリケーションを展開できるようにしたいと考えています。同時に、社内標準に準拠していないインフラストラクチャの起動を制限したいとも考えています。**

**各アプリケーションチームが必要最小限の権限を使用して各々のアプリケーション用のインフラストラクチャを起動できるようにするには、どうすればよいですか。**

A) AWS Service Catalog プロダクトを 2 個作成する。一つは 2 層アーキテクチャを作成するもの、もう一つは 3 層アーキテクチャを作成するものである。技術スタックとデータベース技術をパラメータとして渡す。プロダクトを起動するために必要な権限を、アプリケーションチームに付与する。

B) AWS CloudFormation テンプレートを 2 個作成する。一つは 2 層アーキテクチャを作成するもの、もう一つは 3 層アーキテクチャを作成するものである。技術スタックとデータベース技術をパラメータとして渡す。CloudFormation スタックを作成するために必要な権限を、アプリケーションチームに付与する。

C) AWS CloudFormation テンプレートを作成する。このテンプレートは、AWS Elastic Beanstalk Web サーバー環境のアプリケーションを起動するものである。層数、技術スタック、およびデータベース技術をパラメータとして渡す。CloudFormation スタックを作成するために必要な権限を、アプリケーションチームに付与する。

D) AWS Service Catalog プロダクトを作成する。このプロダクトは、AWS Elastic Beanstalk Web サーバー環境のアプリケーションを起動するものである。層数、技術スタック、およびデータベース技術をパラメータとして渡す。プロダクトを起動するために必要な権限を、アプリケーションチームに付与する。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**(10) ある企業が、Amazon RDS for PostgreSQL マルチ AZ データベースインスタンスに対するリージョン間ディザスタリカバリソリューションを設計しています。このディザスタリカバリソリューションにおける要件は、RPO が 4 時間、RTO が 2 時間です。**

**最も費用対効果の高い方法でこれらの要件を満たすには、どうすればよいですか。**

A) AWS Lambda 関数を作成する。この関数は、RDS のスナップショットを作成し、別のリージョンにコピーするものである。Amazon CloudWatch Events スケジュール化イベントを作成する。このイベントは、Lambda 関数を 4 時間ごとにトリガするものである。RDS 通知イベントを作成する。このイベントは、データベース可用性イベントに関する Amazon SNS メッセージを発行するものである。Lambda 関数を SNS トピックにサブスクライブする。この関数は、スナップショットをディザスタリカバリリージョン内の新しいインスタンスに復元し、アプリケーションに対する接続文字列を更新するものである。

B) AWS Lambda 関数を作成する。この関数は、SQL ダンプファイルを生成し、別のリージョン内の Amazon S3 バケットに格納するものである。Amazon CloudWatch Events スケジュール化イベントを作成する。このイベントは、Lambda 関数を 4 時間ごとにトリガするものである。RDS 通知イベントを作成する。このイベントは、データベース可用性イベントに関する Amazon SNS メッセージを発行するものである。Lambda 関数を SNS トピックにサブスクライブする。この関数は、新しいデータベースインスタンスを起動し、SQL ダンプファイルを実行して、アプリケーションに対する接続文字列を更新するものである。

C) AWS Lambda 関数を作成する。この関数は、自動作成された最新のスナップショットを別のリージョンにコピーするものである。Amazon CloudWatch Events スケジュール化イベントを作成する。このイベントは、Lambda 関数を 4 時間ごとにトリガするものである。RDS 通知イベントを作成する。このイベントは、データベース可用性イベントに関する Amazon SNS メッセージを発行するものである。Lambda 関数を SNS トピックにサブスクライブする。この関数は、スナップショットをディザスタリカバリリージョン内の新しいインスタンスに復元し、アプリケーションに対する接続文字列を更新するものである。

D) データベースインスタンスに対するリードレプリカを別のリージョン内で構成する。RDS 通知イベントを作成する。このイベントは、データベース可用性イベントに関する Amazon SNS メッセージを発行するものである。AWS Lambda 関数を作成する。この関数は、リードレプリカを昇格させ、アプリケーションに対する接続文字列を更新するものである。この関数を SNS トピックにサブスクライブする。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

**回答**

(1) B – これは決定論的な方法で、最も速やかにパイプラインを開始できます。変更が加えられると、イベントが直接生成され、このイベントによってパイプラインが直接トリガされます。A で述べられている定期検査でも問題はありませんが、決定論的ではありません。次回の定期検査が実行されるまで、パイプラインが開始されないからです。B は推奨されるソリューションでもあります。C は、CodeCommit でサポートされていません。D は、パイプラインを開始するための有効な方法ではありません。

(2) A、D – CodeCommit では、IAM ポリシーを使用して、リポジトリに対するアクセス権限を許可および拒否します。また、CloudWatch Events を使用することにより、CodeCommit イベント (例: リポジトリ状態の変更) のストリームをほぼリアルタイムで取得できます。特定のパターンに合致するイベントが発生したときに、CloudWatch Events ルールをトリガし、通知を SNS トピックに送信することができます。B は不正解です。CodeCommit では IAM ポリシーだけがサポートされており、リソースポリシーはサポートされていないからです。C は不正解です。Lambda 関数によってイベントが検出されるまでに最長で 15 分かかるからです。E は不正解です。CloudTrail ログにイベントが記録されるまでに最長で 15 分かかるからです。

(3) C – S3 デフォルト暗号化を使用した場合、アーティファクトは暗号化された状態で格納されます。`Deny` ステートメントを指定し、`NotPrincipal` セクション内で運用ロールを参照した場合、このロールを使用した要求を除いて、バケットへのアクセスが拒否されます。問題文の中で、「運用ロールには、バケットへのアクセスを許可する権限ポリシーが割り当てられている」ということが示されています。A および B は不正解です。バケットポリシー内で、IAM ロールではなく IAM グループを参照しているからです。A は、別の理由でも不正解です。AWS では、バケットポリシーを使用するのではなく、デフォルト暗号化を使用して強制的に暗号化することが推奨されているからです。また、B の場合、アーティファクトを暗号化せずに格納できてしまいます。D は不正解です。AWS KMS では最大 4 KB のデータしか暗号化できないからです。

(4) B、D、E – ブルー/グリーンデプロイメントでは、2 つの環境が構築されます。ブルー環境では、Auto Scaling グループ内の EC2 インスタンス上で現行バージョンのアプリケーションが動作します。グリーン環境では、別の Auto Scaling グループ内の別の EC2 インスタンス上で新バージョンのアプリケーションが動作します。各 Auto Scaling グループは、専用の Application Load Balancer (ALB) の内側にあります。これにより、Route 53 で 2 つのエイリアスレコードをエンドポイントとして構成し、加重ルーティングポリシーを使用して、トラフィックのルーティング先をブルー環境用 ALB からグリーン環境用 ALB に徐々に切り替えることができます。新バージョンに合わせてスキーマを変更する必要がある場合を除き、両環境で同じデータベースを使用する方法が最善です。これにより、新バージョンへの切り替え時にデータの整合性が確保されます。A は不正解です。Route 53 を使用してトラフィックのルーティング先を徐々に切り替えるには、エンドポイントとしての ALB が 2 個必要であるからです。C は不正解です。フェイルオーバールーティングポリシーの場合、ヘルスチェックで障害が検出されない限り、すべてのトラフィックが一方のエンドポイントにのみルーティングされるからです。つまり、トラフィックのルーティング先を徐々に切り替えたい場合、この方法は使用できません。F は不正解です。マルチ AZ RDS の第 2 インスタンスはホットスタンバイ環境であり、読み取り処理および書き込み処理に使用できないからです。

(5) B、D、E – このアイテムにおける重要ポイントは、「S3 内のソースファイルが変更されたことを、テンプレート内で何らかの方法を使用して CloudFormation に示唆する必要がある」ということです。CloudFormation では、ソースファイルのタイムスタンプもチェックサムも保持されないからです。正解の選択肢では、バージョン (B)、コード場所 (D)、またはオブジェクト名 (E) を使用して、テンプレートに変更を加えています。C および F は、テンプレートを大幅に書き換えて SAM テンプレート (SAM は標準 CloudFormation テンプレートを拡張したもの) または serverless.yml ファイルにしない限り、想定どおりに動作しません。A の場合、新しいコードはアップロードされますが、新しい ARN と新しい関数名を持つまったく新しい関数になります。そのため、テンプレート内の他の部分に修正を加える必要があります。また、この関数に依存している、テンプレート以外のリソースがある場合、それらのリソースを停止する必要があります。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved | aws.amazon.com

(6) B、E、F – これらの要件を満たす方法には、2 つの要素があります。一つは、不適合セキュリティ構成を通知すること、もう一つは、すべての要件を満たすよう、SNS ファンアウトを構成することです。B は、不適合セキュリティ構成に関する正確な通知を送信する方法です。A および C では、適合状況にかかわらず、構成変更が発生した場合に通知が送信されます。この場合、通知受信者は、「各メッセージが重要であるかどうかを判断する」という余計な作業を行う必要があります。要件を満たすには、SNS エンドポイントを複数個使用します。E では、SNS の HTTPS エンドポイントを使用して、POST 要求のボディ内のメッセージを配信します。F では、SNS の Lambda エンドポイントを使用して、SNS メッセージから AWS Lambda 関数をトリガします。D では要件は満たされません。メッセージを AWS Config に戻すだけであるからです。

(7) D – Patch Manager を使用した場合、修正プログラムベースライン内で指定した承認済み修正プログラムのリストに基づいて、メンテナンスウィンドウの間にセキュリティ修正プログラムを自動適用できます。セキュリティチームは、Systems Manager コンソールで、各インスタンスの修正プログラム適用状況を確認できます。また、CLI を使用して、概要情報を取得することができます。A は不正解です。CodeBuild はソースコードからアーティファクトをビルドするサービスであるからです。修正プログラムをインスタンスに展開するサービスではありません。B は不正解です。修正プログラムを取得する際に、AWS Systems Manager エージェントをスケジューリングする必要がないからです。修正プログラム適用構成を Systems Manager のメンテナンスウィンドウに関連付けるだけでかまいません。C は不正解です。セキュリティチームが修正プログラム適用状況を検証する手段が含まれていないからです。また、cron ジョブ内に単一障害点があります。

(8) C、D – この質問には 2 つの要素があります。1 つ目は、統合 CloudWatch エージェントがシステムレベルメトリクスを発行する方法です。システムレベルメトリクスは、CloudWatch メトリクスとして発行されます。CloudWatch メトリクスは、他のメトリクスと同様、アラームに対して直接使用できます。したがって、C は正解です。CloudWatch エージェントは、EC2 インスタンス上のログファイルを CloudWatch Logs に発行します。したがって、B は不正解です。CloudWatch Events の機能は、CloudWatch とは異なります。CloudWatch Events は、システムイベント発生時に、またはスケジュールに基づいて、イベントを生成します。したがって、A は不正解です。この質問の 2 つ目の要素は、SNS メッセージへの対処方法です。EC2 インスタンスを SNS メッセージにサブスクライブすることはできません。したがって、E は不正解です。Lambda 関数を SNS メッセージにサブスクライブできます。したがって、D は正解です。

(9) A – 管理者は AWS Service Catalog を使用することにより、プロダクトを発行し、また、プロダクト起動権限を IAM ユーザーに付与することができます。その際、下位サービスを起動する権限をこれらのユーザーに付与することはありません。ユーザーが CloudFormation スタックを起動するには、スタック内のすべての下位インフラストラクチャを起動する権限が必要です。IAM サービスロールを使用して権限を CloudFormation に直接付与する機能もあります。しかし、これらの選択肢では、「権限をアプリケーションチームに直接付与する」と明記されています。Elastic Beanstalk の Web サーバー環境では、Web 層だけを使用できます。Web 層とアプリケーション層の両方を使用することはできません。

(10) A – この方法は、RPO 要件を満たしています。具体的には、スタンバイインスタンスのスナップショットを手動で作成し、別のリージョンにコピーしています。RDS では、SNS トピックに発行可能な通知イベントがサポートされています。この Lambda 関数は、スナップショットを新しいデータベースインスタンスに復元するものです。したがって、アプリケーションに対する接続文字列内の DNS 名を更新する必要があります。B は動作します。ただし、pg_dump プロセスを実行すると、プライマリインスタンス上で大量の I/O が発生します。一方、セカンダリインスタンス上で RDS のスナップショットが作成されます。また、サイズの大きいデータベースの場合、SQL ダンプファイルのサイズが非常に大きくなります。したがって、新しいインスタンスを作成し、ダンプファイル内の SQL コマンドを実行すると、RTO が 2 時間を超えてしまうおそれがあります。C は不正解です。スナップショットは 1 日 1 回しか自動作成されないからです。つまり、RPO 要件を満たしていません。D は動作します。ただし、RPO 要件がわずか 4 時間なので、リージョン間レプリケーションのコストがかかりすぎます。

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved 