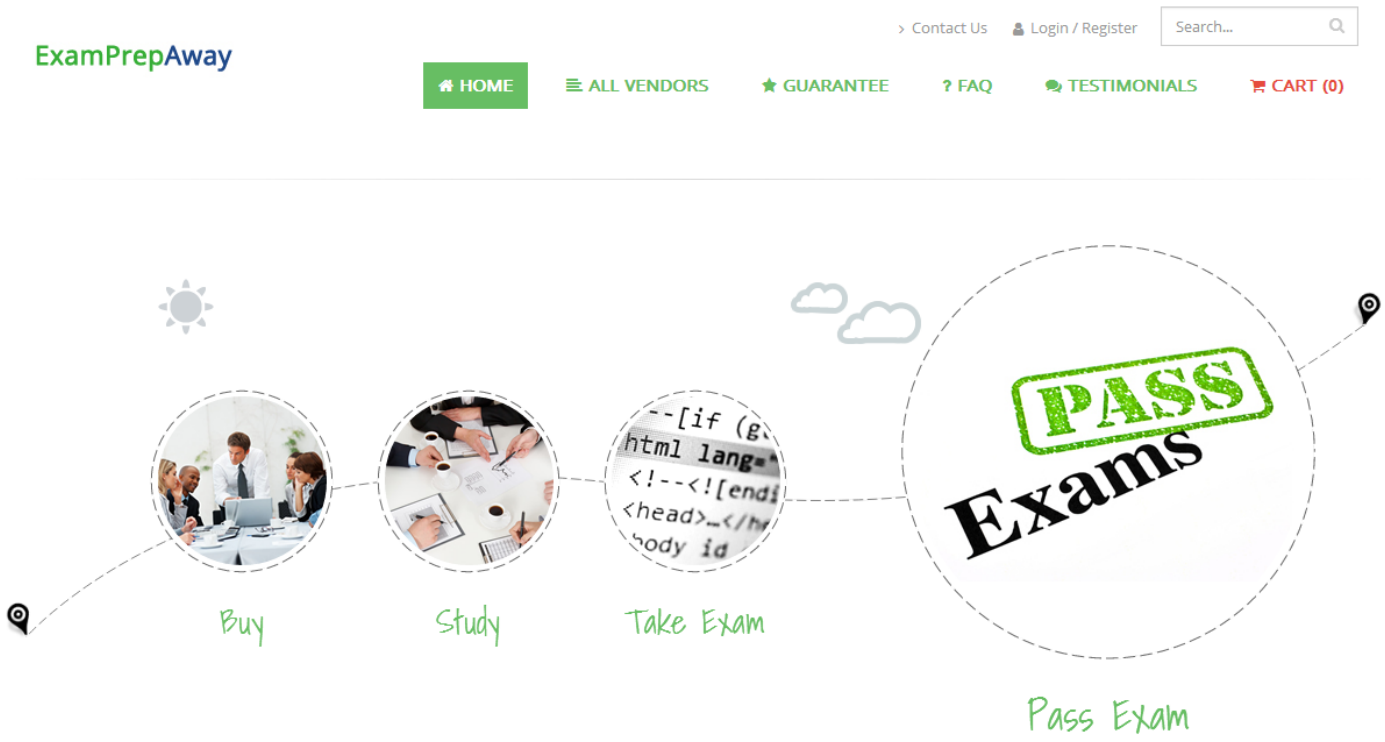
ExamPrepAway



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.

GetCustom JET ORANGE iCompany Paradoxx iMessenger VOREED

http://www.examprepaway.com/

Latest Exam Guide & Learning Materials

Exam : **SAP-C02-JPN**

Title : AWS Certified Solutions Architect - Professional (SAP-C02日本語版)

Vendor : Amazon

Version : DEMO

QUESTION NO: 1
ある企業が、静的コンテンツをホストする新しい Web サイトを設計しています。この Web サイトにより、ユーザーは大きなファイルをアップロードおよびダウンロードできます。会社の要件に従って、すべてのデータは転送中および保管中に暗号化する必要があります。ソリューション アーキテクトは、Amazon S3 と Amazon CloudFront を使用してソリューションを構築しています。
暗号化の要件を満たすのは、どの手順の組み合わせですか? (3 つ選択してください。)
A. Web アプリケーションが使用する S3 バケットの S3 サーバー側暗号化をオンにします。
B. S3 ACL の読み取りおよび書き込み操作用に、"aws:SecureTransport": "true" のポリシー属性を追加します。
C. Web アプリケーションが使用する S3 バケットで暗号化されていない操作を拒否するバケット ポリシーを作成します。
D. AWS KMS キー (SSE-KMS) によるサーバー側の暗号化を使用して、CloudFront で保存時の暗号化を構成します。
E. CloudFront で HTTP リクエストの HTTPS リクエストへのリダイレクトを設定します。
F. Web アプリケーションが使用する S3 バケットの署名付き URL の作成で RequireSSL オプションを使用します。
Answer: A C E
Explanation:
Turning on S3 server-side encryption for the S3 bucket that the web application uses will enable encrypting the data at rest using Amazon S3 managed keys (SSE-S3)1. Creating a bucket policy that denies any unencrypted operations in the S3 bucket that the web application uses will enable enforcing encryption for all requests to the bucket2. Configuring redirection of HTTP requests to HTTPS requests in CloudFront will enable encrypting the data in transit using SSL/TLS3.

QUESTION NO: 2
電力会社は、スマートメーターから5分ごとに使用量データを収集しています。データはAPI Gatewayに送信され、Lambdaで処理された後、DynamoDBに保存されます。使用量が増加するにつれて、Lambdaの実行時間も長くなり、DynamoDBへのPUTリクエストがProvisionedThroughputExceededException エラーで失敗しました。また、LambdaではTooManyRequestsException エラーも発生しています。
これらの問題を解決するには、どの変更の組み合わせが適切でしょうか？(2つ選択してください。)
A. スマートメーターからのペイロードサイズを増やす。
B. Amazon SQS FIFO キューにデータを収集し、各メッセージを処理するための Lambda 関数をトリガーします。
C. API Gateway から Amazon Kinesis データ ストリームにデータをストリーミングし、データをバッチ処理します。
D. DynamoDB テーブルへの書き込み容量ユニットを増やします。
E. ラムダ関数に使用可能なメモリを増やします。
Answer: C,D

QUESTION NO: 3

あるアプリケーションが、us-east-1リージョンにあるAmazon RDS for MySQLマルチAZ DBインスタンスを使用しています。フェイルオーバーテスト後、アプリケーションはデータベースへの接続を失い、接続を再確立できませんでした。
アプリケーションを再起動すると、接続が再確立されました。
ソリューションアーキテクトは、アプリケーションが再起動を必要とせずにデータベースへの接続を再確立できるよう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. 2ノード構成のAmazon Aurora MySQL DBクラスタを作成します。RDS DBインスタンスをAurora
DBクラスタに移行します。RDSプロキシを作成します。既存のRDSエンドポイントをターゲットとして設定します。アプリケーションの接続設定を更新し、RDSプロキシエンドポイントを指すようにします。
B. Amazon S3 バケットを作成します。AWS Database Migration Service (AWS DMS) を使用してデータベースを Amazon S3 にエクスポートします。Amazon Athena が S3 バケットをデータ ストアとして使用するように設定します。アプリケーションに最新の Open Database Connectivity (ODBC)
ドライバーをインストールします。アプリケーションの接続設定を更新して、Athena エンドポイントを指すようにします。
C. Amazon Aurora MySQL Serverless v1 DBインスタンスを作成します。RDS DBインスタンスをAurora Serverless v1
DBインスタンスに移行します。アプリケーションの接続設定を更新して、Auroraリーダーエンドポイントを指すようにします。
D.
RDSプロキシを作成します。既存のRDSエンドポイントをターゲットとして構成します。アプリケーションの接続設定を更新して、RDSプロキシエンドポイントを指すようにします。
Answer: D

QUESTION NO: 4
ある会社では、AWS
上でデータ集約型アプリケーションを実行しています。このアプリケーションは、数百の Amazon EC2 インスタンスのクラスター上で実行されます。共有ファイルシステムも、200 TB のデータを保存する複数の EC2 インスタンス上で実行されます。
アプリケーションは共有ファイル
システム上のデータを読み取って変更し、レポートを生成します。ジョブは毎月 1 回実行され、共有ファイル
システムからファイルのサブセットを読み取り、完了するまでに約 72 時間かかります。
コンピューティング インスタンスは Auto Scaling
グループ内でスケーリングされますが、共有ファイルシステムをホストするインスタンスは継続的に実行されます。コンピューティング インスタンスとストレージ
インスタンスはすべて同じ AWS リージョンにあります。
ソリューション アーキテクトは、共有ファイル システム
インスタンスを置き換えることでコストを削減する必要があります。ファイル
システムは、72 時間の実行中、必要なデータへの高パフォーマンス
アクセスを提供する必要があります。
これらの要件を満たしながら、全体的なコストを最も削減できるソリューションはどれでし

ょうか?
A. 既存の共有ファイルシステムから、S3 Intelligent-Tiering ストレージクラスを使用する Amazon S3 バケットにデータを移行します。毎月ジョブを実行する前に、Amazon FSx for Lustre を使用して、遅延読み込みにより Amazon S3
のデータで新しいファイルシステムを作成します。ジョブの期間中、新しいファイルシステムを共有ストレージとして使用します。ジョブが完了したら、ファイルシステムを削除します。
B. 既存の共有ファイルシステムから、マルチアタッチが有効になっている大規模な Amazon Elastic Block Store (Amazon EBS) ボリュームにデータを移行します。Auto Scaling
グループの起動テンプレートのユーザーデータスクリプトを使用して、各インスタンスに EBS ボリュームをアタッチします。ジョブの期間中、EBS
ボリュームを共有ストレージとして使用します。ジョブが完了したら、EBS
ボリュームをデタッチします。
C. 既存の共有ファイルシステムから、S3 標準ストレージクラスを使用する Amazon S3 バケットにデータを移行します。毎月ジョブを実行する前に、Amazon FSx for Lustre
を使用して、バッチロードにより Amazon S3
のデータで新しいファイルシステムを作成します。ジョブの期間中、新しいファイルシステムを共有ストレージとして使用します。ジョブが完了したら、ファイルシステムを削除します。
D. 既存の共有ファイルシステムから Amazon S3
バケットにデータを移行します。毎月ジョブを実行する前に、AWS Storage Gateway
を使用して Amazon S3
のデータを含むファイルゲートウェイを作成します。ファイルゲートウェイをジョブの共有ストレージとして使用します。ジョブが完了したら、ファイルゲートウェイを削除します。
Answer: A
Explanation:
https://aws.amazon.com/blogs/storage/new-enhancements-for-moving-data-between-amazon-fsx-for-lustre- and-amazon-s3/

QUESTION NO: 5
ソリューションアーキテクトは、Auto Scaling グループ内の Amazon EC2
インスタンスに運用ワークロードを展開しています。VPC アーキテクチャは、Auto Scaling グループがターゲットとするサブネットを持つ 2 つのアベイラビリティーゾーン (AZ)
にまたがっています。VPC
はオンプレミス環境に接続されており、接続が中断されることはありません。Auto Scaling グループの最大サイズは、サービス中のインスタンス 20 個です。VPC IPv4
アドレス指定は次のとおりです。
VPCIDR 10 0 0 0/23
AZ1 サブネット CIDR: 10 0 0 0724
AZ2 サブネット CIDR: 10.0.1 0724
導入後、リージョンで 3 番目の AZ が利用可能になりました。ソリューション
アーキテクトは、追加の IPv4
アドレス空間を追加することなく、またサービスのダウンタイムなしで新しい AZ
を導入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. Auto Scaling グループを更新して、AZ2

サブネットのみを使用します。以前のアドレス空間の半分を使用して AZ1
サブネットを削除して再作成します。Auto Scaling グループを調整して、新しい AZI
サブネットも使用します。インスタンスが正常な場合は、Auto Scaling
グループを調整して、AZ1 サブネットのみを使用します。現在の AZ2
サブネットを削除します。元の AZ1
サブネットのアドレス空間の後半部分を使用して、新しい AZ2
サブネットを作成します。元の AZ2 サブネットのアドレス空間の半分を使用して、新しい AZ3 サブネットを作成し、Auto Scaling グループを更新して、3
つの新しいサブネットすべてをターゲットにします。

B. AZ1 サブネット内の EC2 インスタンスを終了します。アドレス空間全体を使用して AZ1 サブネットを削除し、再作成します。この新しいサブネットを使用するように Auto Scaling グループを更新します。2 番目の AZ に対してこれを繰り返します。
AZ3 に新しいサブネットを定義します。次に、Auto Scaling グループを更新して、3
つの新しいサブネットすべてをターゲットにします。

C. 同じ IPv4 アドレス空間を持つ新しい VPC を作成し、各 AZ に 1 つずつ、3
つのサブネットを定義します。新しい VPC
の新しいサブネットをターゲットにするように既存の Auto Scaling グループを更新します。

D. Auto Scaling グループを更新して、AZ2 サブネットのみを使用するようにします。AZ1 サブネットを更新して、以前のアドレス空間を停止します。Auto Scaling
グループを調整して、AZ1
サブネットも再び使用するようにします。インスタンスが正常なら、Auto Seating
グループを調整して、AZ1 サブネットのみを使用します。現在の AZ2
サブネットを更新し、元の AZ1
サブネットからアドレス空間の後半部分を割り当てます。元の AZ2 サブネット
アドレス空間の半分を使用して新しい AZ3 サブネットを作成し、Auto Scaling
グループを更新して、3 つの新しいサブネットすべてをターゲットにします。

Answer: A
Explanation:
https://repost.aws/knowledge-center/vpc-ip-address-range

QUESTION NO: 6
ある医療関連企業が、Amazon
Bedrock上にユーザーサポート用のチャットアシスタントを構築している。ユーザーは健康に関する質問をするが、その質問には個人情報が含まれる可能性がある。
ソリューションアーキテクトは、以下の機能を備えたソリューションを構成する必要があります。
* アシスタントが医療診断に関する助言を提供しないようにする。
* ユーザー入力とモデル応答の両方から個人識別情報(PII)を削除します。
* 会社が後から基盤モデル(FM)を変更した場合でも、同じ管理策を適用する。
*
不要な推論コストを回避するため、リスクの高いユーザープロンプトをモデルに送信する前に評価してください。
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. 承認済みの健康サポートガイドラインをAmazon
Bedrockナレッジベースに保存します。モデルが診断アドバイスを提供しないように指示す

るシステムプロンプトを設定します。推論後にAWS
Lambda関数を使用して、モデル応答から個人情報(PII)を削除してから、応答をユーザーに返します。
B.
承認済みのサポート会話に関するFM(ファンクションマネージャー)を微調整します。診断アドバイスを禁止するプロンプトテンプレートを追加します。会話終了後、禁止されているトピックや機密情報がないかトランスクリプトをスキャンする別のレビュープロセスを実行します。
C.
アプリケーション内にカスタムモデレーションレイヤーを構築し、プロンプトに禁止されているトピックが含まれていないか検査し、ユーザー入力から機密情報を削除します。Converse API
を介してモデルを呼び出します。ユーザーに応答を表示する前に、応答から機密情報を削除するために、別の後処理ロジックを使用します。
D. Amazon Bedrock
のガードレールを作成します。医療診断アドバイスに関する拒否トピックを設定します。個人情報をマスクするための機密情報フィルターを設定します。コンテンツフィルターを設定します。推論の前に、ユーザープロンプトに対して ApplyGuardrail API
を呼び出します。モデルの応答を評価するために、同じガードレールを Converse API
に含めます。
Answer: D
Explanation:
Amazon Bedrock Guardrails are the correct managed control plane for this scenario. A guardrail can combine denied topics, content filters, and sensitive information filters, and it can be applied to prompts and responses across supported foundation models. The ApplyGuardrail API can evaluate text independently before invoking a foundation model, which helps reject or mask risky prompts before incurring model inference cost.
The Converse API also supports guardrail configuration so the same policy can evaluate conversational model responses. Option A relies on prompts and post-processing only, so it does not evaluate risky prompts before inference and does not protect inputs. Option B is retrospective and model-specific. Option C could work technically but creates custom moderation and response-filtering code, which is higher operational overhead than Bedrock Guardrails.

QUESTION NO: 7
ある会社には、Amazon CloudFront、Amazon API Gateway、および AWS Lambda
関数で構成されるサーバーレス アプリケーションがあります。アプリケーション
コードの現在のデプロイ プロセスは、Lambda
関数の新しいバージョン番号を作成し、AWS CLI
スクリプトを実行して更新することです。新しい関数バージョンにエラーがある場合は、別の CLI
スクリプトが関数の以前の作業バージョンをデプロイすることによって元に戻します。同社は、Lambda 関数によって提供されるアプリケーション
ロジックの新しいバージョンをデプロイする時間を短縮し、エラーが特定されたときにそれを検出して元に戻す時間を短縮したいと考えています。

これはどのように達成できますか？
A. AWS CloudFront ディストリビューションと API Gateway
で構成される親スタックと、Lambda 関数を含む子スタックを使用して、ネストされた
AWS CloudFormation スタックを作成してデプロイします。Lambda
への変更については、AWS CloudFormation
変更セットを作成してデプロイします。エラーが発生した場合は、AWS CloudFormation
の変更セットを以前のバージョンに戻します。
B. AWS SAM と組み込みの AWS CodeDeploy を使用して新しい Lambda
バージョンをデプロイし、トラフィックを徐々に新しいバージョンに移行し、トラフィック
前およびトラフィック後のテスト関数を使用してコードを検証します。Amazon
CloudWatch アラームがトリガーされた場合はロールバックします。
C. AWS CLI スクリプトを、新しい Lambda
バージョンをデプロイする単一のスクリプトにリファクタリングします。デプロイが完了す
ると、スクリプト テストが実行されます。エラーが検出された場合は、以前の Lambda
バージョンに戻します。
D. 新しい Lambda バージョンを参照する新しい API Gateway エンドポイントで構成される
AWS CloudFormation スタックを作成してデプロイします。CloudFront オリジンを新しい
API Gateway
エンドポイントに変更し、エラーを監視して、エラーが検出された場合は、AWS
CloudFront オリジンを以前の API Gateway エンドポイントに変更します。
Answer: B
Explanation:
https://aws.amazon.com/about-aws/whats-new/2017/11/aws-lambda-supports-traffic-shifting-and-phased- deployments-with-aws-codedeploy/

QUESTION NO: 8
ある企業がアプリケーションを AWS に移行しています。移行中はフルマネージド
サービスを可能な限り利用したいと考えています。企業は、次の要件を満たすアプリケーシ
ョン内に、大容量の重要なドキュメントを保存する必要があります。
1 データは高い耐久性と可用性が求められる
2. データは保存時も転送時も常に暗号化される必要があります。
3
暗号化キーは会社によって管理され、定期的にローテーションされる必要があります。ソリ
ューション アーキテクトが推奨すべきソリューションは次のどれですか。
A. ファイルゲートウェイモードでストレージゲートウェイを AWS にデプロイし、AWS
KMS キーを使用して Amazon EBS
ボリューム暗号化を使用してストレージゲートウェイボリュームを暗号化します。
B. バケットポリシーで Amazon S3 を使用して、バケットへの接続に HTTPS
を適用し、オブジェクトの暗号化にサーバー側の暗号化と AWS KMS を適用します。
C. SSL 付きの Amazon DynamoDB を使用して DynamoDB に接続します。AWS KMS
キーを使用して、保存中の DynamoDB オブジェクトを暗号化します。
D. このデータを保存するには、Amazon EBS
ボリュームが接続されたインスタンスをデプロイします。AWS KMS キーを使用して EBS
ボリューム暗号化を行い、データを暗号化します。

Answer: B

QUESTION NO: 9
ある会社は、単一の Amazon EC2
インスタンスで重要なアプリケーションをホストしています。このアプリケーションは、インメモリ データ ストアに Amazon ElastiCache for Redis 単一ノード
クラスターを使用します。アプリケーションは、リレーショナル データベースに Amazon RDS for MariaDB DB
インスタンスを使用します。アプリケーションが機能するには、インフラストラクチャの各部分が正常で、アクティブな状態である必要があります。
ソリューション
アーキテクトは、最小限のダウンタイムでインフラストラクチャが障害から自動的に回復できるように、アプリケーションのアーキテクチャを改善する必要があります。
これらの要件を満たすのは、どのステップの組み合わせですか? (3 つ選択してください。)
A. Elastic Load Balancer を使用して、複数の EC2
インスタンスにトラフィックを分散します。EC2 インスタンスが、最小容量が 2
つのインスタンスを持つ Auto Scaling グループの一部であることを確認します。
B. Elastic Load Balancer を使用して、複数の EC2
インスタンスにトラフィックを分散させます。EC2
インスタンスが無制限モードで構成されていることを確認してください。
C. DB
インスタンスを変更して、同じアベイラビリティーゾーンにリードレプリカを作成します。障害シナリオでは、リードレプリカをプライマリ DB インスタンスに昇格させます。
D. DB インスタンスを変更して、2 つのアベイラビリティーゾーンにまたがるマルチ AZ 配置を作成します。
E. ElastiCache for Redis クラスターのレプリケーション
グループを作成します。最小容量が 2 つのインスタンスである Auto Scaling
グループを使用するようにクラスターを構成します。
F. ElastiCache for Redis クラスターのレプリケーション
グループを作成します。クラスターでマルチ AZ を有効にします。
Answer: A D F
Explanation:
Option A is correct because using an Elastic Load Balancer and an Auto Scaling group with a minimum capacity of two instances can improve the availability and scalability of the EC2 instances that host the application. The load balancer can distribute traffic across multiple instances and the Auto Scaling group can replace any unhealthy instances automatically1 Option D is correct because modifying the DB instance to create a Multi-AZ deployment that extends across two Availability Zones can improve the availability and durability of the RDS for MariaDB database. Multi- AZ deployments provide enhanceddata protection and minimize downtime by automatically failing over to astandby replica in another Availability Zone in case of a planned or unplanned outage4 Option F is correct because creating a replication group for the ElastiCache for Redis cluster and enabling Multi-AZ on the cluster can improve the availability and fault tolerance of the in-memory data store. A replication group consists of a primary node and up to five read-only replica nodes that are synchronized with the primary node using asynchronous replication. Multi-AZ allows automatic failover to one of the replicas

if the primary node fails or becomes unreachable6 References: 1: https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing- works.html 2: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-performance-instances- unlimited-mode.html 3: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl. html 4: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html 5: https://docs. aws.amazon.com/AmazonElastiCache/latest/red-ug/AutoScaling.html 6: https://docs.aws.amazon.com /AmazonElastiCache/latest/red-ug/Replication.Redis.Groups.html

QUESTION NO: 10
ある企業は、オンプレミスのインフラストラクチャと AWS の間に専用接続を確立したいと考えています。この企業は、アカウント VPC への 1 Gbps AWS Direct Connect 接続を設定しています。アーキテクチャには、複数の VPC とオンプレミスのインフラストラクチャを接続するためのトランジットゲートウェイと Direct Connect ゲートウェイが含まれています。
企業は、Direct Connect 接続を使用して、トランジット VIF 経由で VPC リソースに接続する必要があります。
どの手順の組み合わせがこれらの要件を満たしますか? (2 つ選択してください。)
A. 1 Gbps の Direct Connect 接続を 10 Gbps に更新します。
B. オンプレミスのネットワーク プレフィックスをトランジット VIF 経由でアドバタイズします。
C. Direct Connect ゲートウェイからオンプレミス ネットワークへの VPC プレフィックスをトランジット VIF 経由で転送します。
D. Direct Connect 接続の MACsec 暗号化モード属性を暗号化必須に更新します。
E. MACsec 接続キー名と接続関連付けキー (CKN/CAK) のペアを Direct Connect 接続に関連付けます。
Answer: B C
Explanation:
To connect VPC resources over a transit Virtual Interface (VIF) using a Direct Connect connection, the company should advertise the on-premises network prefixes over the transit VIF and advertise the VPC prefixes from the Direct Connect gateway to the on-premises network over the same VIF. This configuration ensures seamless connectivity between the on-premises infrastructure and the AWS VPCs through the transit gateway, facilitating efficient and secure communication across the network.
AWS Documentation on AWS Direct Connect and transit gateways provides detailed instructions on configuring transit VIFs and routing for Direct Connect connections. This setup is recommended in AWS best practices for establishing dedicated network connections between on-premises environments and AWS to achieve low-latency, high-throughput, and secure connectivity.

QUESTION NO: 11
ある会社が、VPC 内の Amazon EC2 インスタンスに新しいプライベート イントラネット サービスを展開することを計画しています。AWS サイト間 VPN は、VPC

を会社のオンプレミス ネットワークに接続します。新しいサービスは、既存のオンプレミス サービスと通信する必要があります。オンプレミス サービスには、会社のサンプル DNS ゾーンにあるホスト名を使用してアクセスできます。この DNS ゾーンは完全にオンプレミスでホストされており、会社のプライベート ネットワークでのみ使用できます。
ソリューション アーキテクトは、新しいサービスが会社のサンプル ドメイン上のホスト名を解決して既存のサービスと統合でき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すか?

A. Amazon Route 53 に会社例の空のプライベートゾーンを作成します。会社のオンプレミスの会社例のゾーンに、Route 53 の新しいプライベートゾーンの権威ネームサーバーを指す追加の NS レコードを追加します。

B. VPC の DNS ホスト名をオンにします。Amazon Route 53 Resolver を使用して新しいアウトバウンドエンドポイントを設定します。会社例のリクエストをオンプレミスのネームサーバーに転送する Resolver ルールを作成します。

C. VPCのDNSホスト名をオンにするAmazon Routeで新しいインバウンドリゾルバエンドポイントを設定する 53 リゾルバ。オンプレミスの DNS サーバーを構成して、会社例の要求を新しいリゾルバに転送します。

D. AWS Systems Manager を使用して、必要なホスト名を含むホストファイルをインストールする実行ドキュメントを設定します。インスタンスが実行状態に入るときにドキュメントを実行するには、Amazon EventBndge ルールを使用します。

Answer: B
Explanation:
https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resolver.html

QUESTION NO: 12
ある企業がAWSクラウドでワークロードを実行しています。この企業は、アプリケーションのデータをAmazon DocumentDBの旧バージョンに保存しています。複数のバックエンドサービスが、一日中データベースへのデータの読み書きを継続的に行っています。すべてのサービスは、Amazon Route 53にDNSレコードとして登録されているAmazon DocumentDBクラスターエンドポイントを使用してデータベースに接続します。
同社は、データを失うことなくデータベースをAmazon DocumentDBの最新バージョンにアップグレードする必要があります。バックエンドサービスにアップグレード版の使用を許可する前に、アップグレードのテストと検証を行う必要があります。同社は既に変更ストリームを有効化し、24時間の保持期間を設定しています。
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. 最新バージョンを実行する新しい Amazon DocumentDB クラスターを作成します。Amazon DocumentDB インデックスツールを使用して既存のインデックスをエクスポートし、新しいクラスターにインポートします。新しい AWS DMS インスタンスと、ソースエンドポイントおよびターゲットエンドポイントを作成します。移

行タイプ「移行とレプリケーション」を使用してデータを移行する DMS
タスクを作成します。新しいクラスターをテストして検証します。Route 53
レコードを更新して、新しいクラスターを参照するようにします。
B. 最新バージョンを実行する新しい Amazon DocumentDB
クラスターを作成します。Amazon EC2 インスタンスに MongoDB
コマンドラインインターフェイス (CLI)
データベースツールをインストールします。MongoDB CLI
を使用してバイナリエクスポートを作成し、新しい Amazon DocumentDB
クラスターにデータをインポートします。新しいクラスターをテストして検証します。Rout e 53 レコードを更新して、新しいクラスターを参照するようにします。
C. 既存の Amazon DocumentDB
クラスターのスナップショットを作成します。インプレースメジャーバージョンアップグレードを実行します。既存のクラスターを最新バージョンと最新のクラスターパラメータグループに変更します。変更はすぐに適用します。アップグレードをテストして検証します。
D. 最新バージョンを実行する新しい Amazon DocumentDB クラスターを作成します。AWS DataSync エージェントを Amazon EC2
インスタンスにデプロイし、エージェントを有効化します。拡張モードで新しい AWS DataSync
タスクを作成します。転送タスクを開始し、新しいクラスターにデータをコピーします。新しいクラスターをテストして検証します。Route 53
レコードを更新して、新しいクラスターを参照するようにします。
Answer: A
Explanation:
The company needs to upgrade DocumentDB to the latest version with no data loss while allowing continuous reads and writes. The company also must be able to test and verify the upgrade before switching production traffic. This is a classic requirement for performing an upgrade using a blue/green approach: build a new target environment on the new version, keep it in sync with the source, validate it, and then cut over by changing the endpoint (here, Route 53 DNS).
Option A implements this pattern using a new DocumentDB cluster running the latest version and AWS DMS to continuously migrate and replicate changes from the old cluster to the new cluster. Because the workload is continuously changing, a one-time export/import is insufficient; continuous replication is needed to keep the target cluster current during the test period. AWS DMS supports a "migrate and replicate" style of task that performs a full load and then applies ongoing changes (CDC) so the target stays synchronized. The question also states that change streams are enabled with a 24-hour retention period, which supports capturing and applying changes during migration/validation and helps ensure the replication stream can be maintained while testing.
Option A also addresses indexes by using the DocumentDB Index Tool to export and import indexes, which is important because indexes can affect query performance and behavior. After the company validates the new cluster, the cutover is done by updating the Route 53 record to point to the new cluster endpoint, switching all backend services without changing application configuration beyond DNS resolution.
Option B uses MongoDB CLI tools to export/import. This is not suitable for continuous write workloads because export/import is a point-in-time operation and would require downtime or

risk data divergence during the test period. It also adds more operational overhead and does not provide continuous replication for the duration of validation.
Option C performs an in-place major version upgrade. That does not satisfy the requirement to test and verify the upgrade before backend services use the upgraded version because the upgrade happens directly on the production cluster. Even though a snapshot exists for rollback, production is still exposed to the upgrade immediately, which violates the requirement for pre-cutover verification.
Option D is incorrect because AWS DataSync transfers files between storage systems such as NFS/SMB and AWS storage services. It is not a database migration or replication service and cannot copy a DocumentDB database in a way that preserves database semantics and supports continuous replication.
Therefore, creating a new DocumentDB cluster, keeping it synchronized using AWS DMS (supported by change stream retention), validating it, and then cutting over via Route 53 DNS update (option A) meets all requirements.
References:
AWS documentation on blue/green style database upgrades by migrating to a new cluster and cutting over via DNS.
AWS documentation on AWS DMS full load plus ongoing replication (CDC) patterns for minimizing downtime and maintaining target synchronization during validation.
AWS documentation on Amazon DocumentDB change streams and retention considerations for capturing ongoing changes during migration windows.

QUESTION NO: 13
ある企業が AWS への移行を希望しています。同社は VMware ESXi 環境で数千の VM を実行しています。同社には構成管理データベースがなく、VMware ポートフォリオの利用に関する知識もほとんどありません。
ソリューション
アーキテクトは、企業が費用対効果の高い移行を計画できるように、正確なインベントリを企業に提供する必要があります。
運用オーバーヘッドを最小限に抑えながら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 Systems Manager Patch Manager を使用して、Migration Evaluator を各 VM にデプロイします。Amazon QuickSight で収集されたデータを確認します。使用率の高いサーバーを特定します。使用率の高いサーバーを移行リストから削除します。データを AWS Migration Hub にインポートします。
B. VMware ポートフォリオを csv ファイルにエクスポートします。各サーバーのディスク使用率を確認します。使用率の高いサーバーを削除します。データを AWS Application Migration Service にエクスポートします。AWS Server Migration Service (AWS SMS) を使用して、残りのサーバーを移行します。
C. Migration Evaluator エージェントレス コレクターを ESXi ハイパーバイザーにデプロイします。Migration Evaluator で収集されたデータを確認します。非アクティブなサーバーを特定します。非アクティブなサーバーを移行リストから削除します。データを AWS Migration Hub にインポートします。

D. AWS Application Migration Service Agent を各 VM にデプロイします。データが収集されたら、Amazon Redshift を使用してデータをインポートし、分析します。データの視覚化には Amazon QuickSight を使用します。

Answer: C

Explanation:

https://aws.amazon.com/migration-evaluator/features/

QUESTION NO: 14

ソリューションアーキテクトは、手動で作成された既存の非本番 AWS 環境から AWS CloudFormation テンプレートを作成しています。 CloudFormation テンプレートは、必要に応じて破棄して再作成できます。 環境には Amazon EC2 インスタンスが含まれています。 EC2 インスタンスには、EC2 インスタンスが親アカウントでロールを引き受けるために使用するインスタンスプロファイルがあります。 ソリューションアーキテクトは、CloudFormation テンプレートでロールを再作成し、同じロール名を使用します。 CloudFormation テンプレートが子アカウントで起動されると、権限が不十分なため、EC2 インスタンスは親アカウントでロールを引き受けることができなくなります。

ソリューションアーキテクトはこの問題を解決するために何をすべきでしょうか?

A.

親アカウントで、EC2インスタンスが引き受ける必要のあるロールの信頼ポリシーを編集します。sts AssumeRoleアクションを許可する既存のステートメントのターゲットロールARNが正しいことを確認します。信頼ポリシーを保存します。

B.

親アカウントで、EC2インスタンスが引き受ける必要のあるロールの信頼ポリシーを編集します。子アカウントのルートプリンシパルに対してsts AssumeRoleアクションを許可するステートメントを追加します。信頼ポリシーを保存します。

C.

CloudFormationスタックを再度更新し、CAPABILITY_NAMED_IAM機能のみを指定します。

D.

CloudFormationスタックを再度更新し、CAPABIUTYJAM機能とCAPABILITY_NAMEDJAM機能を指定します。

Answer: A

Explanation:

Edit the Trust Policy:

Go to the IAM console in the parent account and locate the role that the EC2 instance needs to assume.

Edit the trust policy of the role to ensure that it correctly allows the sts action for the role ARN in the child account.

Update the Role ARN:

Verify that the target role ARN specified in the trust policy matches the role ARN created by the CloudFormation stack in the child account.

If necessary, update the ARN to reflect the correct role in the child account.
Save and Test:
Save the updated trust policy and ensure there are no syntax errors.
Test the setup by attempting to assume the role from the EC2 instance in the child account.
Verify that the instance can successfully assume the role and perform the required actions.
This ensures that the EC2 instance in the child account can assume the role in the parent account, resolving the permission issue.
References
AWS IAM Documentation on Trust Policies#51#.

QUESTION NO: 15
ある企業には、大都市全体の交通パターンを監視する IoT
センサーがあります。この企業は、センサーからデータを読み取って収集し、そのデータを集計したいと考えています。
ソリューションアーキテクトは、IoT デバイスが Amazon Kinesis Data Streams
にストリーミングするソリューションを設計しています。複数のアプリケーションがストリームから読み取りを行っています。ただし、複数のコンシューマーでスロットリングが発生しており、定期的に RealProvisioned Throughput Exceeded エラーが発生しています。
この問題を解決するためにソリューション
アーキテクトが実行すべきアクションはどれですか? (3 つ選択してください。)
A. ストリームを再シャーディングして、ストリーム内のシャードの数を増やします。
B. Kinesis Producer Library KPL を使用します。ポーリング頻度を調整します。
C. 拡張ファンアウト機能を備えたコンシューマーを使用します。
D. ストリームを再シャーディングして、ストリーム内のシャードの数を減らします。
E. コンシューマー ロジックでエラー再試行と指数バックオフ メカニズムを使用します。
F. 動的パーティション分割を使用するようにストリームを構成します。
Answer: A C E
Explanation:
https://repost.aws/knowledge-center/kinesis-readprovisionedthroughputexceeded Follow Data Streams best practices To mitigate ReadProvisionedThroughputExceeded exceptions, apply these best practices:
* Reshard your stream to increase the number of shards in the stream.
* Use consumers with enhanced fan-out. For more information about enhanced fan-out, see Developing custom consumers with dedicated throughput (enhanced fan-out).
* Use an error retry and exponential backoff mechanism in the consumer logic if ReadProvisionedThroughputExceeded exceptions are encountered. For consumer applications that use an AWS SDK, the requests are retried by default.

QUESTION NO: 16
質問：
ある企業がAmazon EC2とAWS
Lambda上でアプリケーションを実行しています。アプリケーションはAmazon
S3に一時データを保存します。S3オブジェクトは24時間後に削除されます。
同社はAWS
CloudFormationスタックを起動することで、アプリケーションの新バージョンをデプロイし

ています。スタックは必要なリソースを作成します。新バージョンの検証後、古いスタックを削除します。最近、古い開発スタックの削除に失敗しました。
ソリューション
アーキテクトは、アーキテクチャを大幅に変更せずにこの問題を解決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バケットからオブジェクトを削除するLambda関数を作成します。Lambda関数をCloudFormationスタックにカスタムリソースとして追加し、S3バケットリソースを指すDependsOn属性を設定します。

B. CloudFormation スタックを変更して、Delete の値を持つ DeletionPolicy 属性を S3 バケットにアタッチします。

C. CloudFormation スタックを更新して、S3 バケット リソースの Snapshot 値を持つ DeletionPolicy 属性を追加します。

D. CloudFormation テンプレートを更新し、Amazon S3 の代わりに Amazon EFS ファイルシステムを作成して一時ファイルを保存できるようにします。Lambda 関数を EFS ファイルシステムと同じ VPC で実行するように設定します。

Answer: A
Explanation:
CloudFormation cannot delete non-empty S3 buckets. OptionAallows you to create acustom Lambda resourcethat deletes all objects in the S3 bucket before the stack deletes it. The DependsOn ensures the bucket deletion occurs only after the Lambda has completed.
B: Adding DeletionPolicy: Delete does not resolve the issue if the bucket still contains objects.
C: Snapshot doesn ' t apply to S3 and won ' t help here.
D: Changing to Amazon EFS would require architectural changes, which are not allowed per requirements.
Reference:https://aws.amazon.com/blogs/devops/safely-delete-s3-buckets-using-aws-cloudformation/https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-attribute- deletionpolicy.html

QUESTION NO: 17
企業には複数の AWS アカウントがあります。開発チームは、クラウドガバナンスと修復プロセスの自動化フレームワークを構築しています。自動化フレームワークは、一元化されたアカウントで AWS Lambda 関数を使用します。ソリューションアーキテクトは、会社の各 AWS アカウントで Lambda 関数を実行できるようにする最小権限のアクセス許可ポリシーを実装する必要があります。これらの要件を満たすのは、どのステップの組み合わせですか?（2つ選んでください。）

A. 一元化されたアカウントで、信頼できるエンティティとして Lambda サービスを持つ IAM ロールを作成します。インライン ポリシーを追加して、他の AWS アカウントのロールを引き受けます。

B. 他の AWS アカウントで、最小限の権限を持つ IAM ロールを作成します。一元化されたアカウントの Lambda IAM ロールを信頼できるエンティティとして追加します。

C.
一元化されたアカウントで、信頼されたエンティティとして他のアカウントのロールを持つ

IAM ロールを作成します。最小限のアクセス許可を提供します。
D. 他の AWS アカウントで、集中管理されたアカウントのロールを引き受ける権限を持つ IAM ロールを作成します。信頼できるエンティティとして Lambda サービスを追加します。
E. 他の AWS アカウントで、最小限の権限を持つ IAM ロールを作成します。信頼できるエンティティとして Lambda サービスを追加します。
Answer: A B
Explanation:
https://medium.com/@it.melnichenko/invoke-a-lambda-across-multiple-aws-accounts-8c094b2e70be

QUESTION NO: 18
質問：
ある企業では、ユーザーがアップロードした動画をAmazon S3標準ストレージを使用してS3バケットに保存するアプリケーションを運用しています。ユーザーは最初の180日間は頻繁に動画にアクセスしますが、それ以降はほとんどアクセスしません。ほとんどの動画は100MBを超えています。ユーザーのインターネット接続が不安定な場合が多いため、この企業はマルチパートアップロードを使用しています。
ソリューション アーキテクトは S3 ストレージ コストを最適化する必要があります。
これらの要件を満たすアクションの組み合わせはどれですか? (2 つ選択してください)。
A. S3 バケットを Requester Pays バケットとして設定します。
B. S3 Transfer Acceleration を使用してビデオをアップロードします。
C. 7 日後に不完全なマルチパートアップロードを期限切れにするライフサイクル ルールを作成します。
D. 1 日後にオブジェクトを S3 Glacier Instant Retrieval に移行するライフサイクル ルールを作成します。
E. 180 日後にオブジェクトを S3 Standard-IA に移行するライフサイクル ルールを作成します。
Answer: C E
Explanation:
C: Multipart uploads can leave incomplete parts behind, which incur storage costs. Expiring them after 7 days minimizes waste and saves cost.
E: Since objects are infrequently accessed after 180 days, transitioning toS3 Standard-IAis cost-effective, especially for large files > 128 KB (your 100 MB+ files qualify).
Ais for shifting download cost, not reducing your S3 storage expenses.
Bhelps with upload speed butincreases cost.
Dis too aggressive; Glacier is not suited for access patterns within the first few days.
Reference:https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-configuration-examples.
htmlhttps://docs.aws.amazon.com/AmazonS3/latest/userguide/mpuoverview.html

QUESTION NO: 19
ある企業は、オンプレミスのデータセンターを AWS に移行することを計画しています。同社は現在、Linux ベースの VMware VM でデータセンターをホストしています。ソリューション アーキテクトは、VM 間のネットワーク依存関係に関する情報を収集する必要があります。情報は、ホストの IP

アドレス、ホスト名、およびネットワーク接続情報の詳細を示す図の形式である必要があります。
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 Application Discovery Service を使用します。 AWS Migration Hub のホーム AWS リージョンを選択します。データ収集のためにオンプレミスサーバーに AWS Application Discovery Agent をインストールします。 Migration Hub ネットワーク ダイアグラムを使用するためのアクセス許可を Application Discovery Service に付与します。
B. サーバー データの収集には、AWS Application Discovery Service Agentless Collector を使用します。 AWS Migration Hub からネットワーク図を .png 形式でエクスポートします。
C. データ収集のためにオンプレミス サーバーに AWS Application Migration Service エージェントをインストールします。 AWS の Workload Discovery で AWS Migration Hub データを使用して、ネットワーク図を生成します。
D. データ収集のためにオンプレミスサーバーに AWS Application Migration Service エージェントをインストールします。 AWS Migration Hub から .csv 形式でデータを Amazon CloudWatch ダッシュボードにエクスポートして、ネットワーク図を生成します。
Answer: A
Explanation:
To effectively gather information about network dependencies between VMs in an on-premises data center for migration to AWS, it ' s crucial to use tools that can capture detailed application and server dependencies. The AWS Application Discovery Service is designed for this purpose, particularly when migrating from environments like Linux-based VMware VMs. By installing the AWS Application Discovery Agent on the on- premises servers, the service can collect necessary data such as host IP addresses, hostnames, and network connection information. This data is crucial for creating a comprehensive network diagram that outlines the interactions and dependencies between various components of the on-premises infrastructure. The integration with AWS Migration Hub enhances this process by allowing the visualization of these dependencies in a network diagram format, aiding in the planning and execution of the migration process. This approach ensures a thorough understanding of the on-premises environment, which is essential for a successful migration to AWS.
References:
AWS Documentation on Application Discovery Service: This provides detailed guidance on how to use the Application Discovery Service, including the installation and configuration of the Discovery Agent.
AWS Migration Hub User Guide: Offers insights on how to integrate Application Discovery Service data with Migration Hub for comprehensive migration planning and tracking.
AWS Solutions Architect Professional Learning Path: Contains advanced topics and best practices for migrating complex on-premises environments to AWS, emphasizing the use of AWS services and tools for effective migration planning and execution.

QUESTION NO: 20
ある企業が、今後3年間実行されるプロジェクトのために、AWS上でアプリケーションをホスティングしています。このアプリケーションは、ネットワークロードバランサー（NLB）

のターゲットグループに登録された20個のAmazon
EC2オンデマンドインスタンスで構成されています。インスタンスは2つのアベイラビリティゾーンに分散されています。アプリケーションはステートレスで、
1日24時間、週7日。
同社は、アプリケーションからの応答が遅いと感じているユーザーからの報告を受けています。
パフォーマンス メトリックは、通常のアプリケーション使用時にインスタンスの CPU 使用率が 10% であることを示しています。
ただし、通常数時間続くビジー時には CPU 使用率が 100% まで増加します。
同社では、アプリケーションからの応答が遅いという問題を解決するために、新しいアーキテクチャを必要としています。
これらの要件を最もコスト効率よく満たすソリューションはどれでしょうか?
A. Auto Scaling グループを作成します。Auto Scaling グループを NLB のターゲットグループにアタッチします。最小容量を 20、希望容量を 28 に設定します。20 インスタンス分のリザーブドインスタンスを購入します。
B. リクエストタイプが request であるスポットフリートを作成します。TotalTargetCapacity パラメータを 20 に設定します。DefaultTargetCapacityType パラメータを On-Demand に設定します。スポットフリートの作成時に NLB を指定します。
C.
リクエストタイプがmaintainのスポットフリートを作成します。TotalTargetCapacityパラメータを20に設定します。DefaultTargetCapacityTypeパラメータをSpotに設定します。NLBをApplication Load Balancerに置き換えます。
D. Auto Scaling グループを作成します。Auto Scaling グループを NLB のターゲットグループにアタッチします。最小容量を 4、最大容量を 28 に設定します。4 つのインスタンス分のリザーブドインスタンスを購入します。
Answer: D

QUESTION NO: 21
ある企業が自社のデータセンターとAWSを組み合わせたハイブリッドソリューションを開発しました。Amazon VPCとAmazon
EC2インスタンスを使用し、アプリケーションログをAmazon
CloudWatchに送信しています。EC2インスタンスは、オンプレミスでホストされている複数のリレーショナルデータベースからデータを読み取ります。
同社は、どのEC2インスタンスがデータベースに接続しているかをほぼリアルタイムで監視したいと考えています。同社は既にオンプレミスでSplunkを使用した監視ソリューションを導入しています。ソリューションアーキテクトは、ネットワークトラフィックをSplunkに送信する方法を決定する必要があります。
ソリューション アーキテクトはこれらの要件をどのように満たすべきでしょうか?
A.
VPCフローログを有効にし、CloudWatchに送信します。定義済みのエクスポート関数を使用して、CloudWatchログをAmazon S3バケットに定期的にエクスポートするAWS Lambda関数を作成します。AWS認証情報（ACCESS_KEYとSECRET_KEY）を生成します。これらの認証情報を使用してS3バケットからログを取得するようにSplunkを設定します。
B. Splunk を宛先とする Amazon Data Firehose 配信ストリームを作成します。CloudWatch

Logs
サブスクリプションフィルターによって送信されたレコードから個々のログイベントを抽出する Firehose ストリームプロセッサを使用して、前処理 AWS Lambda 関数を設定します。VPC フローログを有効にし、CloudWatch に送信します。ログイベントを Firehose 配信ストリームに送信する CloudWatch Logs サブスクリプションを作成します。

C.
データベースへのすべてのリクエストとEC2インスタンスのIPアドレスをログに記録するよう依頼します。CloudWatchログをAmazon S3バケットにエクスポートします。Amazon Athenaを使用して、データベース名でグループ化されたログをクエリします。Athenaの結果を別のS3バケットにエクスポートします。AWS Lambda関数を呼び出して、S3バケットに追加された新しいファイルを自動的にSplunkに送信します。

D. Amazon Managed Service for Apache Flink（旧称Amazon Kinesis Data Analytics）を使用して、CloudWatch ログを Amazon Kinesis データストリームに送信します。イベントを収集するための 1 分間のスライディングウィンドウを設定します。異常検出テンプレートを使用して、ネットワークトラフィックの異常をほぼリアルタイムで監視する SQL クエリを作成します。結果を、Splunk を宛先として Amazon Data Firehose 配信ストリームに送信します。

Answer: B

Explanation:
The company needs near-real-time visibility into which EC2 instances are connecting to on-premises databases. The correct telemetry source for network connection metadata at the VPC level is VPC Flow Logs.
VPC Flow Logs capture information about IP traffic going to and from network interfaces in a VPC, including source/destination IPs, ports, protocol, and accept/reject decisions. This data can be used to infer which EC2 instance IPs are connecting to database IPs.
The company already uses Splunk on premises, so the solution should deliver these logs to Splunk with minimal delay and operational overhead. Amazon Data Firehose provides a fully managed way to deliver streaming data to supported destinations, including Splunk, with buffering and retry handling. CloudWatch Logs subscription filters can stream log events in near real time from CloudWatch Logs to destinations such as Firehose.
Option B uses the standard pattern: enable VPC Flow Logs to CloudWatch Logs, then create a CloudWatch Logs subscription filter that streams the flow logs to a Firehose delivery stream configured with Splunk as the destination. Because CloudWatch Logs subscription deliveries can batch log events, using a Firehose preprocessing Lambda to extract individual log events is a common approach to format records in a way that Splunk ingests cleanly. This yields near-real-time delivery with low operational overhead.
Option A introduces delay because it exports CloudWatch logs periodically to S3 and requires Splunk to poll S3. It also requires long-lived access keys and periodic batch exports, which is not near real time.
Option C relies on application-level logging changes and batch analytics with Athena, which is not near real time and requires substantial changes and additional pipelines.
Option D is over-engineered for the stated requirement. Using Flink and anomaly detection

focuses on anomalies rather than simply identifying connections, and it adds significant operational complexity compared to direct delivery of flow logs to Splunk via Firehose. Therefore, streaming VPC Flow Logs from CloudWatch Logs to Splunk using a Firehose delivery stream and a subscription filter is the best approach.
References:AWS documentation on VPC Flow Logs and the metadata they provide for network connection visibility.AWS documentation on CloudWatch Logs subscription filters for near-real-time streaming of log events.AWS documentation on Amazon Data Firehose delivery to Splunk and optional Lambda transformations for record formatting.

QUESTION NO: 22
ある企業が Amazon Workspaces の概念実証を成功させました。現在、同社は 2 つの AWS リージョンにわたって Workspaces の可用性を高めたいと考えています。Workspaces はフェイルオーバーリージョンにデプロイされています。ホストゾーンは Amazon Route 53 で利用できます。
ソリューション アーキテクトは何をすべきでしょうか?
A. プライマリ リージョンとフェイルオーバー
リージョンに接続エイリアスを作成します。それぞれをそのリージョン内のディレクトリに関連付けます。Evaluate Target Health = Yes で Route 53 フェイルオーバー ルーティング ポリシーを作成します。
B.
両方のリージョンで接続エイリアスを作成します。両方をプライマリリージョンのディレクトリに関連付けます。Route 53 の複数値回答ルーティングポリシーを使用します。
C.
プライマリリージョンに接続エイリアスを作成します。プライマリリージョンのディレクトリに関連付けます。
Route 53 の重み付けルーティングを使用します。
D. プライマリ リージョンに接続エイリアスを作成します。これをフェイルオーバー リージョンのディレクトリに関連付けます。Evaluate Target Health = Yes で Route 53 フェイルオーバー ルーティングを使用します。
Answer: A
Explanation:
A is correct because AWS recommends using oneconnection alias per Region, associated witheach directory.
Then, configure a Route 53 failover policy so that if the primary Region becomes unhealthy, users are directed to the failover Region automatically. "Evaluate Target Health" ensures automatic detection and failover.
References:
Amazon Workspaces Cross-Region Resilience
Route 53 Failover Routing

QUESTION NO: 23
企業は AWS Organizations に組織を持っています。同社は AWS Control Tower を使用して、組織のランディング
ゾーンをデプロイしています。この会社は、ガバナンスとポリシーの施行を実装したいと考えています。会社は、会社の実稼働 OU で保存時に暗号化されていない Amazon RDS DB

インスタンスを検出するポリシー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. AWS Control Tower で必須のガードレールをオンにします。必須のガードレールを運用 OU に適用します。
B. AWS Control Tower で強く推奨されるガードレールのリストから適切なガードレールを有効にします。ガードレールを運用 OU に適用します。
C. AWS Config を使用して、新しい必須のガードレールを作成します。運用 OU のすべてのアカウントにルールを適用します。
D. AWS Control Tower でカスタム SCP を作成します。SCP を運用 OU に適用します。
Answer: B
Explanation:
AWS Control Tower provides a set of " strongly recommended guardrails " that can be enabled to implement governance and policy enforcement. One of these guardrails is " Encrypt Amazon RDS instances " which will detect RDS DB instances that are not encrypted at rest. By enabling this guardrail and applying it to the production OU, the company will be able to enforce encryption for RDS instances in the production environment.

QUESTION NO: 24
ある冒険会社が、モバイル アプリに新しい機能を導入しました。ユーザーはこの機能を使用して、ハイキングやラッティングの写真やビデオをいつでもアップロードできます。写真とビデオは、S3 バケットの Amazon S3 標準ストレージに保存され、Amazon CloudFront を通じて提供されます。
この会社は、ストレージのコストを最適化する必要があります。ソリューション アーキテクトは、アップロードされた写真とビデオのほとんどが 30 日後にほとんどアクセスされていないことに気付きました。ただし、アップロードされた写真やビデオの一部は、30 日後に頻繁にアクセスされます。ソリューション アーキテクトは、可能な限り低いコストで写真やビデオをミリ秒単位で取得でき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. S3 バケットで S3 Intelligent-Tiering を構成します。
B. S3 ライフサイクル ポリシーを構成して、30 日後にイメージ オブジェクトとビデオ オブジェクトを S3 標準から S3 Glacier Deep Archive に移行します。
C. Amazon S3 を、Amazon EC2 インスタンスにマウントされた Amazon Elastic File System (Amazon EFS) ファイル システムに置き換えます。
D. Cache-Control: max-age ヘッダーを S3 イメージ オブジェクトと S3 ビデオ オブジェクトに追加します。ヘッダーを 30 日に設定します。
Answer: A
Explanation:
Amazon S3 Intelligent-Tiering is a storage class that automatically moves objects between two access tiers based on changing access patterns. Objects that are accessed frequently are stored in the frequent access tier and objects that are accessed infrequently are stored in the infrequent access tier. This allows for cost optimization without requiring manual intervention. This makes it an ideal solution for the scenario described, as it can automatically

move objects that are infrequently accessed after 30 days to a lower-cost storage tier while still maintaining millisecond retrieval availability.

QUESTION NO: 25
ある会社が人気のビデオ
ゲームの新リリースを開発し、一般公開でダウンロードできるようにしたいと考えています。新リリース パッケージのサイズは約 5 GB です。同社は、オンプレミス データ
センターでホストされている Linux ベースの一般公開 FTP
サイトから、既存のリリースのダウンロードを提供しています。同社は、世界中のユーザーが新リリースをダウンロードすると予想しています。同社は、ユーザーの場所に関係なく、ダウンロード
パフォーマンスが向上し、転送コストが低くなるソリューションを求めています。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. Auto Scaling グループ内の Amazon EC2 インスタンスにマウントされた Amazon EBS ボリュームにゲームファイルを保存します。EC2 インスタンスで FTP
サービスを設定します。Auto Scaling グループの前に Application Load Balancer
を使用します。ユーザーがパッケージをダウンロードできるようにゲームのダウンロード URL を公開します。
B. Auto Scaling グループ内の Amazon EC2 インスタンスにアタッチされた Amazon EFS ボリュームにゲームファイルを保存します。各 EC2 インスタンスで FTP
サービスを設定します。Auto Scaling グループの前に Application Load Balancer
を使用します。ユーザーがパッケージをダウンロードできるようにゲームのダウンロード URL を公開します。
C. ウェブサイトホスティング用に Amazon Route 53 と Amazon S3 バケットを設定する ゲームファイルを S3 バケットにアップロードする ウェブサイトに Amazon CloudFront を使用する ユーザーがパッケージをダウンロードできるようにゲームダウンロード URL を公開する
D. ウェブサイトホスティング用に Amazon Route 53 と Amazon S3 バケットを設定する ゲームファイルを S3 バケットにアップロードする S3
バケットのリクエスタ支払いを設定する
ユーザーがパッケージをダウンロードできるようにゲームのダウンロード URL を公開する
Answer: C
Explanation:
Create an S3 Bucket:
Navigate to Amazon S3 in the AWS Management Console and create a new S3 bucket to store the game files.
Enable static website hosting on this bucket.
Upload Game Files:
Upload the 5 GB game release package to the S3 bucket. Ensure that the files are publicly accessible if required for download.
Configure Amazon Route 53:
Set up a new domain or subdomain in Amazon Route 53 and point it to the S3 bucket. This allows users to access the game files using a custom URL.
Use Amazon CloudFront:
Create a CloudFront distribution with the S3 bucket as the origin. CloudFront is a content

delivery network (CDN) that caches content at edge locations worldwide, improving download performance and reducing latency for users regardless of their location.
Publish the Download URL:
Use the CloudFront distribution URL as the download link for users to access the game files. CloudFront will handle the efficient distribution and caching of the content.
This solution leverages the scalability of Amazon S3 and the performance benefits of CloudFront to provide an optimal download experience for users globally while minimizing costs.
References
Amazon CloudFront Documentation
Amazon S3 Static Website Hosting

QUESTION NO: 26
複数の AWS アカウントを持つ企業が AWS Organizations を使用しています。同社の AWS アカウントは、VPC、Amazon EC2 インスタンス、およびコンテナをホストしています。同社のコンプライアンス チームは、会社が展開している各 VPC にセキュリティ ツールを展開しています。セキュリティ ツールは EC2 インスタンスで実行され、コンプライアンス チーム専用の AWS アカウントに情報を送信します。この会社は、コンプライアンス関連のすべてのリソースに「costCenter」のキーと値または「compliance」のタグを付けています。
会社は、コンプライアンス チームの AWS アカウントに請求できるように、EC2 インスタンスで実行されているセキュリティ ツールのコストを特定したいと考えています。原価計算は、できるだけ正確にする必要があります。
これらの要件を満たすために、ソリューション アーキテクトは何をすべきでしょうか?
A. 組織の管理アカウントで、costCenter ユーザー定義タグを有効にします。毎月の AWS のコストと使用状況レポートを構成して、管理アカウントの Amazon S3 バケットに保存します。レポートのタグ内訳を使用して、costCenter タグ付きリソースの総コストを取得します。
B. 組織のメンバー アカウントで、costCenter ユーザー定義タグを有効にします。毎月の AWS のコストと使用状況レポートを構成して、管理アカウントの Amazon S3 バケットに保存します。毎月の AWS Lambda 関数をスケジュールして、レポートを取得し、costCenter タグ付きリソースの総コストを計算します。
C. 組織のメンバー アカウントで、costCenter ユーザー定義タグを有効にします。管理アカウントから、毎月の AWS のコストと使用状況レポートをスケジュールします。レポートのタグ内訳を使用して、cost Center タグ付きリソースの総コストを計算します。
D. AWS Trusted Advisor の組織ビューでカスタム レポートを作成します。コンプライアンス チームの AWS アカウントの costCenter タグ付きリソースの月次請求概要を生成するようにレポートを構成します。
Answer: A
Explanation:
https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/custom-tags.htmlhttps://docs.aws.amazon.com

/awsaccountbilling/latest/aboutv2/configurecostallocreport.html

QUESTION NO: 27
ある企業は、Amazon Managed Streaming for Apache Kafka (Amazon MSK)
クラスターからのメッセージを利用する、レイテンシーの影響を受けやすいアプリケーションを運用しています。MSK クラスターは 3
つのアベイラビリティゾーンにまたがって稼働しています。
現在の MSK クラスターは、各アベイラビリティーゾーンに 2
つの標準の大規模インスタンスを備えた標準ブローカーを使用します。
同社は、ブローカーと同じアベイラビリティゾーンにデプロイされているApache Kafkaクライアント間のレイテンシを最小限に抑えたいと考えています。また、利用可能な帯域幅を増やし、クラスターのスケーリング速度を向上させたいと考えています。現在、クライアントはデフォルト設定を使用しています。ソリューションの実装中は、ある程度のダウンタイムは許容されます。
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.
予測スケーリングポリシーを設定し、MSKクラスターをターゲットとして設定します。ターゲット値を80に設定し、スケジューリングバッファサイズを0に設定します。Kafkaクライアントの配置グループを設定し、MSKホストをその配置グループに関連付けます。
B. MSK クラスターで Cruise Control を構成し、帯域幅制御と再バランスを有効にします。
レイテンシーベースのルーティングを使用する Amazon MSK Connect
プロキシレイヤーをデプロイします。プロキシエンドポイントを使用するように Kafka
クライアントを再設定します。
C. 標準ブローカーを、Express Large Instancesを使用するExpressブローカーに置き換えます。Kafkaクライアントのclient.rackプロパティをaz_idに設定します。
D.
ブローカーのサイズを標準のxlargeインスタンスに変更します。各アベイラビリティゾーンにMSK
PrivateLinkエンドポイントを作成します。各Kafkaクライアントを、クライアントと同じアベイラビリティゾーンにあるエンドポイントを使用するように再設定します。
Answer: C
Explanation:
The company wants three things: minimize client-to-broker latency within the same Availability Zone, increase available bandwidth, and increase the scaling speed of the MSK cluster. The current brokers are Standard brokers (two per AZ). Clients use default settings, which means they are not explicitly configured for rack awareness or AZ affinity.
A common way to reduce latency in multi-AZ Kafka deployments is to enable rack awareness on clients and brokers so clients prefer brokers in the same "rack," which can map to an Availability Zone. In Kafka, the client.rack setting allows the client to include rack information so the broker can return metadata that helps the client select replicas that are closest, reducing cross-AZ traffic and improving latency.
To increase bandwidth and improve scaling speed, the most direct approach in the choices is to move from Standard brokers to Express brokers. Express brokers are designed to provide higher throughput and faster scaling characteristics compared to standard broker types.

Since the question explicitly calls out increasing available bandwidth and scaling speed, the broker type change is the key lever, and it can be combined with client.rack configuration to minimize cross-AZ latency.
Option C matches these requirements: it replaces Standard brokers with Express brokers (to improve throughput/bandwidth and scaling speed) and sets client.rack to the Availability Zone identifier (az_id) to improve locality and reduce latency between clients and brokers in the same AZ.
Option A is not appropriate because MSK does not use EC2 Auto Scaling predictive scaling in that manner, and Kafka clients/brokers are not "associated" with an EC2 placement group as a primary latency solution in MSK. Placement groups are for EC2 instance placement; MSK broker placement is managed by the service.
Option B introduces a proxy layer and MSK Connect in a way that increases complexity and does not directly guarantee lower latency or higher bandwidth. MSK Connect is for Kafka Connect workloads, not as a general- purpose low-latency routing proxy for Kafka clients. Cruise Control is used for partition rebalancing and cluster optimization, but it does not replace the benefits of higher-throughput broker types and client rack awareness for AZ locality.
Option D increases broker size and introduces PrivateLink endpoints. PrivateLink is about private connectivity from VPCs to services and does not inherently ensure AZ-local broker selection or reduce latency between clients and brokers in the same AZ. Also, resizing to xlarge increases capacity but does not address scaling speed and locality as directly as express brokers plus rack configuration.
Therefore, option C best meets all requirements.
References:AWS documentation on Amazon MSK broker types, including performance and scaling characteristics of Standard and Express brokers.Apache Kafka concepts and AWS guidance on rack awareness and using client.rack to reduce cross-AZ traffic and latency in multi-AZ Kafka deployments.

QUESTION NO: 28
ある会社が環境データを処理しています。市内のさまざまな工リアから継続的にデータストリームを提供するセンサーを設置しています。データは JSON 形式で利用できます。同社は、AWS
ソリューションを使用して、保存用に固定スキーマを必要としないデータベースにデータを送信したいと考えています。データはリアルタイムで送信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. Amazon Kinesis Data Firehouse を使用してデータを Amazon Redshift に送信します。

B. Amazon Kinesis Data ストリームを使用してデータを Amazon DynamoDB に送信します。

C. Amazon Managed Streaming for Apache Kafka (Amazon MSK) を使用して、データを Amazon Aurora に送信します。

D. Amazon Kinesis Data firehouse を使用して、データを Amazon Keyspaces (Apache Cassandra 用) に送信します。

Answer: B
Explanation:
Amazon Kinesis Data Streams is a service that enables real-time data ingestion and

processing. Amazon DynamoDB is a NoSQL database that does not require fixed schemas for storage. By using Kinesis Data Streams and DynamoDB, the company can send the JSON data to a database that can handle schemaless data in real time. References:
https://docs.aws.amazon.com/streams/latest/dev/introduction.html
https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html

QUESTION NO: 29
ある企業は、GitHub Actions を使用して、AWS 上のリソースにアクセスする CI/CD パイプラインを実行しています。この企業には、パイプライン内の秘密鍵を使用して AWS への認証を行う IAM ユーザーがいます。ポリシーがアタッチされた既存の IAM ロールによって、リソースのデプロイに必要な権限が付与されています。
会社のセキュリティチームは、パイプラインで長期間有効な秘密鍵を使用できないという新しい要件を導入しました。ソリューションアーキテクトは、秘密鍵を短期間有効なソリューションに置き換える必要があります。
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しょうか?
A. IAM で IAM SAML 2.0 アイデンティティプロバイダー (IdP) を作成します。sts:AssumeRole API 呼び出しを許可する適切な信頼ポリシーを持つ新しい IAM ロールを作成します。既存の IAM ポリシーを新しい IAM ロールにアタッチします。パイプラインで SAML 認証を使用するように GitHub を更新します。
B. IAM で IAM OpenID Connect (OIDC) ID プロバイダー (IdP) を作成します。GitHub OIDC IdP からの sts:AssumeRoleWithWebIdentity API 呼び出しを許可する適切な信頼ポリシーを持つ新しい IAM ロールを作成します。パイプラインのロールを引き受けるように GitHub を更新します。
C. Amazon Cognito IDプールを作成します。GitHubを使用するように認証プロバイダーを設定します。GitHub認証プロバイダーからのsts:AssumeRoleWithWebIdentity API呼び出しを許可する適切な信頼ポリシーを持つ新しいIAMロールを作成します。パイプラインを設定して、Cognitoを認証プロバイダーとして使用します。
D. AWS プライベート CA へのトラストアンカーを作成します。AWS IAM Roles Anywhere で使用するクライアント証明書を生成します。sts:AssumeRole API 呼び出しを許可する適切な信頼ポリシーを持つ新しい IAM ロールを作成します。既存の IAM ポリシーを新しい IAM ロールにアタッチします。パイプラインを設定して、認証情報ヘルパーツールを使用し、クライアント証明書の公開鍵を参照して新しい IAM ロールを引き継ぎます。
Answer: B
Explanation:
This explanation is based on AWS documentation and best practices but is paraphrased, not a literal extract.
The current CI/CD pipeline uses an IAM user with long-lived access keys stored in GitHub Actions. The new requirement is that pipelines must not use long-lived secret keys. Instead, the solution should provide short- lived credentials with minimal operational overhead.
GitHub Actions natively supports integration with cloud providers using OpenID Connect (OIDC). With OIDC, GitHub acts as an identity provider that can issue OIDC tokens to workflows. On the AWS side, IAM supports configuring an OIDC identity provider and roles

that can be assumed by principals presenting valid OIDC tokens through the sts:AssumeRoleWithWebIdentity API. This pattern enables short-lived, automatically rotated credentials for CI/CD jobs without storing long-lived secrets.
In the correct solution (option B), you configure an IAM OIDC identity provider for GitHub in the AWS account. You then create a new IAM role with a trust policy that allows the Github OIDC provider to call sts:
AssumeRoleWithWebIdentity, with conditions that restrict which repositories or workflows can assume the role. The existing IAM policy that grants deployment permissions is attached to that role. In GitHub Actions, you update the pipeline configuration to request an OIDC token and assume the IAM role at runtime. Each workflow run receives short-lived credentials without storing static keys, and AWS automatically handles the token verification and temporary credential issuance. This approach is the AWS-recommended pattern for integrating GitHub Actions with AWS without long-lived secrets and has low operational overhead once configured.
Option A uses SAML 2.0, which is typically used for enterprise single sign-on for users, not for GitHub Actions workflows. GitHub does not natively use SAML to obtain AWS credentials for CI/CD pipelines in the same streamlined way as OIDC, and implementing a SAML-based integration would add unnecessary complexity.
Option C introduces Amazon Cognito as an indirection layer. Although Cognito can federate with external identity providers, including social providers, using it as an intermediary to obtain temporary AWS credentials for a machine-to-machine CI/CD pipeline is not necessary when IAM OIDC federation with GitHub is directly supported. This adds additional configuration and operational overhead.
Option D uses IAM Roles Anywhere with client certificates from AWS Private CA. Roles Anywhere is designed for workloads running outside AWS that need to assume IAM roles using X.509 certificates instead of access keys. While technically possible, it requires managing private certificates, trust anchors, and a credential helper tool, which is more complex and operationally heavier than the direct OIDC integration specifically designed for GitHub Actions.
Therefore, configuring an IAM OIDC identity provider for GitHub and creating an IAM role to be assumed via sts:AssumeRoleWithWebIdentity (option B) meets the requirement to replace long-lived secret keys with short-lived credentials with the least operational overhead .
References:AWS documentation on configuring IAM OpenID Connect identity providers and roles for GitHub Actions integration.AWS security best practices recommending federation and temporary credentials over long-lived IAM user access keys for CI/CD pipelines.

QUESTION NO: 30
ある会社は、さまざまなベンダーからアプライアンスを購入しました。アプライアンスにはすべて IoT センサーが搭載されています。センサーは、情報を JSON に解析するレガシーアプリケーションに、ベンダー独自の形式でステータス情報を送信します。解析は単純ですが、各ベンダーには独自の形式があります。1 日 1 回、アプリケーションはすべての JSON レコードを解析し、分析のためにレコードをリレーショナル データベースに保存します。同社は、より迅速に提供し、コストを最適化できる新しいデータ分析ソリューションを設計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. IoT センサーを AWS IoT Core に接続します。AWS Lambda 関数を呼び出して情報を解析し、.csv ファイルを Amazon S3 に保存するルールを設定します。AWS Glue を使用してファイルをカタログ化します。分析には Amazon Athena と Amazon QuickSight を使用します。
B. アプリケーション サーバーを AWS Fargate に移行します。AWS Fargate は IoT センサーから情報を受け取り、その情報をリレーショナル形式に解析します。解析した情報を分析のために Amazon Redshift に保存します。
C. AWS Transfer for SFTP サーバーを作成します。IoT センサー コードを更新して、情報を .csv ファイルとして SFTP 経由でサーバーに送信します。AWS Glue を使用してファイルをカタログ化します。分析には Amazon Athena を使用します。
D. AWS Snowball Edge を使用して IoT センサーから直接データを収集し、ローカル分析を実行します。データを定期的に Amazon Redshift に収集して、グローバル分析を実行します。
Answer: A
Explanation:
Connect the IoT sensors to AWS IoT Core. Set a rule to invoke an AWS Lambda function to parse the information and save a .csv file to Amazon S3. Use AWS Glue to catalog the files. Use Amazon Athena and Amazon QuickSight for analysis. This solution meets the requirement of faster analysis and cost optimization by using AWS IoT Core to collect data from the IoT sensors in real-time and then using AWS Glue and Amazon Athena for efficient data analysis.
This solution involves connecting the IoT sensors to the AWS IoT Core, setting a rule to invoke an AWS Lambda function to parse the information, and saving a .csv file to Amazon S3. AWS Glue can be used to catalog the files and Amazon Athena and Amazon QuickSight can be used for analysis. This solution will enable faster and more cost-effective data analysis.
This solution is in line with the official Amazon Textbook and Resources for the AWS Certified Solutions Architect - Professional certification. In particular, the book states that: "AWS IoT Core can be used to ingest and process the data, AWS Lambda can be used to process and transform the data, and Amazon S3 can be used to store the data. AWS Glue can be used to catalog and access the data, Amazon Athena can be used to query the data, and Amazon QuickSight can be used to visualize the data." (Source:https://d1.awsstatic.com/training-and-certification/docs-sa-pro/AWS_Certified_Solutions_Architect_Professional_Exam_Guide_EN_v1.5.pdf)

QUESTION NO: 31
ある会社は、ビジネス ユニットごとに社内のクラウド請求戦略を変更したいと考えています。現在、クラウド ガバナンス チームは、クラウド支出全体のレポートを各ビジネス ユニットの責任者と共有しています。同社は AWS Organizations を使用して、各ビジネス ユニットの個別の AWS アカウントを管理しています。Organizations の既存のタグ付け基準には、アプリケーション、環境、および所有者が含まれます。クラウド ガバナンス チームは、各ビジネス ユニットがクラウド支出に関する月次レポートを受け取るための一元化されたソリューショ

ンを望んでいます。ソリューションは、設定されたしきい値を超えるクラウドの支出についても通知を送信する必要があります。
これらの要件を満たすための最も費用対効果の高い方法はどれですか?
A. 各アカウントで AWS Budgets
を構成し、アプリケーション、環境、所有者ごとにグループ化された予算アラートを構成します。各アラートの Amazon SNS トピックに各ビジネス
ユニットを追加します。各アカウントで Cost Explorer を使用して、各ビジネス
ユニットの月次レポートを作成します。
B. 組織のマスター アカウントで AWS Budgets
を構成し、アプリケーション、環境、所有者ごとにグループ化された予算アラートを構成します。各アラートの Amazon SNS トピックに各ビジネス
ユニットを追加します。組織のマスター アカウントで Cost Explorer
を使用して、各ビジネス ユニットの月次レポートを作成します。
C. 各アカウントで AWS Budgets
を構成し、アプリケーション、環境、所有者ごとにグループ化された予算アラートを構成します。各アラートの Amazon SNS トピックに各ビジネス
ユニットを追加します。各アカウントで AWS Billing and Cost Management
ダッシュボードを使用して、各ビジネス ユニットの月次レポートを作成します。
D. 組織のマスター アカウントで AWS
のコストと使用状況レポートを有効にし、アプリケーション、環境、所有者別にグループ化されたレポートを構成します。AWS
のコストと使用状況レポートを処理し、予算アラートを送信し、月次レポートを各ビジネス
ユニットの E メール リストに送信する AWS Lambda 関数を作成します。
Answer: B
Explanation:
Configure AWS Budgets in the organization #€™s master account and configure budget alerts that are grouped by application, environment, and owner. Add each business unit to an Amazon SNS topic for each alert. Use Cost Explorer in the organization #€™s master account to create monthly reports for each business unit.
https://aws.amazon.com/about-aws/whats-new/2019/07/introducing-aws-budgets-reports/#:~:text=AWS%
20Budgets%20gives%20you%20the,below%20the%20threshold%20you%20define

QUESTION NO: 32
ある会社は、新しい Web ベースのアプリケーションを展開しており、Linux
アプリケーション サーバー用のストレージ
ソリューションを必要としています。この会社は、すべてのインスタンスのアプリケーション
データを更新するための単一の場所を作成したいと考えています。アクティブなデータセットのサイズは最大 100 GB です。ソリューション アーキテクトは、ピーク操作が毎日 3
時間発生し、合計 225 MiBps の読み取りスループットが必要になると判断しました。
ソリューションアーキテクトは、災害復旧 (DR) のために別の AWS
リージョンでデータのコピーを利用できるようにするマルチ AZ
ソリューションを設計する必要があります。DR コピーの RPO は 1 時間未満です。
これらの要件を満たすソリューションはどれですか?

A. 新しい Amazon Elastic File System (Amazon EFS) マルチ AZ ファイル システムをデプロイします。プロビジョニングされたスループットが 75 MiBps になるようにファイル システムを構成します。DR リージョンのファイル システムにレプリケーションを実装します。
B. 新しい Amazon FSx for Lustre ファイル システムをデプロイします。ファイル システムの Bursting Throughput モードを構成します。AWS Backup を使用して、ファイル システムを DR リージョンにバックアップします。
C. スループットが 225 MiBps の汎用 SSD (gp3) Amazon Elastic Block Store (Amazon EBS) ボリュームをデプロイします。EBS ボリュームのマルチアタッチを有効にします。AWS Elastic Disaster Recovery を使用して、EBS ボリュームを DR リージョンに複製します。
D. Amazon FSx for OpenZFS ファイル システムを本番リージョンと DR リージョンの両方にデプロイします。AWS DataSync のスケジュールされたタスクを作成して、本番ファイル システムから DR ファイル システムに 10 分ごとにデータをレプリケートします。
Answer: A
Explanation:
The company should deploy a new Amazon Elastic File System (Amazon EFS) Multi-AZ file system. The company should configure the file system for 75 MiBps of provisioned throughput. The company should implement replication to a file system in the DR Region. This solution will meet the requirements because Amazon EFS is a serverless, fully elastic file storage service that lets you share file data without provisioning or managing storage capacity and performance. Amazon EFS is built to scale on demand to petabytes without disrupting applications, growing and shrinking automatically as you add and remove files1. By deploying a new Amazon EFS Multi-AZ file system, the company can create a single location for updates to application data for all instances. A Multi-AZ file system replicates data across multiple Availability Zones (AZs) within a Region, providing high availability and durability2. By configuring the file system for 75 MiBps of provisioned throughput, the company can ensure that it meets the peak operations requirement of 225 MiBps of read throughput. Provisioned throughput is a feature that enables you to specify a level of throughput that the file system can drive independent of the file system's size or burst credit balance3. By implementing replication to a file system in the DR Region, the company can make a copy of the data available in another AWS Region for disaster recovery. Replication is a feature that enables you to replicate data from one EFS file system to another EFS file system across AWS Regions. The replication process has an RPO of less than 1 hour.
The other options are not correct because:
Deploying a new Amazon FSx for Lustre file system would not provide a single location for updates to application data for all instances. Amazon FSx for Lustre is a fully managed service that provides cost- effective, high-performance storage for compute workloads. However, it does not support concurrent write access from multiple instances. Using AWS Backup to back up the file system to the DR Region would not provide real-time replication of data. AWS Backup is a service that enables you to centralize and automate data protection across AWS services. However, it does not support continuous data replication or cross-Region disaster recovery.
Deploying a General Purpose SSD (gp3) Amazon Elastic Block Store (Amazon EBS) volume

with 225 MiBps of throughput would not provide a single location for updates to application data for all instances.
Amazon EBS is a service that provides persistent block storage volumes for use with Amazon EC2 instances.
However, it does not support concurrent access from multiple instances, unless Multi-Attach is enabled.
Enabling Multi-Attach for the EBS volume would not provide Multi-AZ resilience or cross-Region replication. Multi-Attach is a feature that enables you to attach an EBS volume to multiple EC2 instances within the same Availability Zone. Using AWS Elastic Disaster Recovery to replicate the EBS volume to the DR Region would not provide real-time replication of data. AWS Elastic Disaster Recovery (AWS DRS) is a service that enables you to orchestrate and automate disaster recovery workflows across AWS Regions.
However, it does not support continuous data replication or sub-hour RPOs.
Deploying an Amazon FSx for OpenZFS file system in both the production Region and the DR Region would not be as simple or cost-effective as using Amazon EFS. Amazon FSx for OpenZFS is a fully managed service that provides high-performance storage with strong data consistency and advanced data management features for Linux workloads. However, it requires more configuration and management than Amazon EFS, which is serverless and fully elastic. Creating an AWS DataSync scheduled task to replicate the data from the production file system to the DR file system every 10 minutes would not provide real-time replication of data.
AWS DataSync is a service that enables you to transfer data between on-premises storage and AWS services, or between AWS services. However, it does not support continuous data replication or sub-minute RPOs.
References:
https://aws.amazon.com/efs/
https://docs.aws.amazon.com/efs/latest/ug/how-it-works.html#how-it-works-azs
https://docs.aws.amazon.com/efs/latest/ug/performance.html#provisioned-throughput
https://docs.aws.amazon.com/efs/latest/ug/replication.html
https://aws.amazon.com/fsx/lustre/
https://aws.amazon.com/backup/
https://aws.amazon.com/ebs/
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes-multi.html

QUESTION NO: 33
ある会社のウェブアプリケーションでは、Amazon API Gateway API、AWS Lambda 関数、Amazon DynamoDB
グローバルテーブルを使用してバックエンドリクエストを処理しています。ウェブアプリケーションは、アクティブ/パッシブモデルの 2 つの AWS
リージョンにデプロイされています。この会社では、DNS に Amazon Route 53
を使用しています。ウェブアプリケーションでは、セカンダリリージョンにフェイルオーバーするために、手動での DNS 更新が必要です。分析 Lambda 関数は同じ AWS
アカウントで実行されています。この関数により、Lambda
の同時実行数が平均的な日に現在のクォータの 90%
に達しています。最近、分析ワークロードのトラフィックが急増したため、Lambda
リクエストが抑制され、ウェブアプリケーションユーザーのユーザーエクスペリエンスが低

下しました。ソリューションアーキテクトは、ウェブアプリケーションの信頼性を高める必要があります。このソリューションでは、Lambda
の同時実行数が特定の使用率しきい値に達したときに、Amazon CloudWatch
アラームを使用して Amazon SNS
通知を送信する必要があります。どのソリューションが、運用オーバーヘッドが最も少なく、これらの要件を満たすでしょうか。

A. ウェブアプリケーションのLambda関数に予約同時実行を設定します。Route 53ヘルスチェックとフェイルオーバーレコードを実装し、トラフィックをセカンダリリージョンにルーティングします。AWS Trusted AdvisorのServiceLimitUsageメトリクスを使用し、SNS通知を送信するようにCloudWatchアラームを設定します。

B. ウェブアプリケーションのLambda関数に予約同時実行を設定します。Route 53ヘルスチェックとレイテンシーレコードを実装し、トラフィックをセカンダリリージョンにルーティングします。CloudWatchアラームを設定し、AWS Trusted AdvisorのServiceLimitUsageメトリクスを使用し、SNS通知を送信します。

C.
ウェブアプリケーションのLambda関数にプロビジョニングされた同時実行性を設定します。Route
53のヘルスチェックとフェイルオーバーレコードを実装し、トラフィックをセカンダリリージョンにルーティングします。LambdaのConcurrentExecutionsメトリクスを使用し、SNS通知を送信するようにCloudWatchアラームを設定します。

D.
ウェブアプリケーションのLambda関数にプロビジョニングされた同時実行性を設定します。Route
53ヘルスチェックと位置情報レコードを実装し、トラフィックをセカンダリリージョンにルーティングします。CloudWatchアラームを設定して、Lambda ProvisionedConcurrencyInvocationsメトリクスを使用し、SNS通知を送信します。

Answer: C

Explanation:

The use of provisioned concurrency ensures the web application's Lambda functions have pre-initialized execution environments, removing cold start latency and maintaining performance during high-traffic periods.

Route 53 health checks and failover records automate DNS failover to the secondary Region, improving application availability and reliability.

The CloudWatch alarm is configured to monitor the Lambda ConcurrentExecutions metric and send an SNS notification if concurrency usage nears the limit, enabling quick operational response.

This approach minimizes manual management, ensuring reliability and performance during peak traffic while meeting best practices for AWS Lambda and Route 53 failover.

QUESTION NO: 34

ある会社が、複数の Amazon DynamoDB
テーブルにデータを保存しています。ソリューション アーキテクトは、サーバーレス
アーキテクチャを使用して、HTTPS を介した単純な API
を介してデータにパブリックにアクセスできるようにする必要があります。ソリューション

は、需要に応じて自動的にスケーリング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？（2つ選んでください。）
A. Amazon API Gateway REST API を作成します。API Gateway の AWS 統合タイプを使用して、DynamoDB への直接統合でこの API を設定します。
B. Amazon API Gateway HTTP API を作成します。API Gateway の AWS 統合タイプを使用して、Dynamo DB への直接統合でこの API を設定します。
C. Amazon API Gateway HTTP API を作成します。DynamoDB テーブルからデータを返す AWS Lambda 関数への統合を使用して、この API を設定します。
D. AWS Global Accelerator でアクセラレーターを作成します。DynamoDB テーブルからデータを返す AWS Lambda@Edge
関数の統合を使用して、このアクセラレータを構成します。
E. Network Load Balancer を作成します。リクエストを適切な AWS Lambda 関数に転送するようにリスナー ルールを設定する
Answer: A C
Explanation:
https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-overview-developer-experience.
html

QUESTION NO: 35
ある企業がAWS上でコンテナ化されたワークロードを実行しています。このワークロードは、Amazon
EC2インスタンスのグループ上で実行される複数のデータ処理サービスで構成されています。
同社は毎晩Amazon
S3バケットに新しいデータをアップロードしています。各EC2インスタンスのcronジョブが毎晩データ処理を開始します。アップロードされるデータの量は変動するため、データ処理タスクの実行には数時間かかることがあります。データ処理後、サービスは翌晩の次の処理ウィンドウまでアイドル状態のままです。同社は、アーキテクチャを近代化し、運用オーバーヘッドを削減するソリューションを必要としています。
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. コンテナイメージを実行するAWS Lambda関数にワークロードを移行します。Amazon EventBridgeルールを設定して、S3イベントをフィルタリングし、データがS3バケットにアップロードされたときにLambda関数を呼び出します。
B. AWS Fargate 上で実行される Amazon ECS クラスター内のタスクとして実行するようにワークロードを移行します。Fargate タスクを呼び出すための AWS Step Functions ステートマシンを作成します。データが S3 バケットにアップロードされたときにステートマシンタスクを呼び出すように S3 イベント通知を設定します。
C. AWS Fargate 上で実行される Amazon ECS クラスター内のタスクとして実行するようにワークロードを移行します。Fargate タスクを呼び出すための AWS Step Functions ステートマシンを作成します。データが S3 バケットにアップロードされたときにステートマシンを呼び出すように Amazon EventBridge ルールを設定します。

D. コンテナイメージを Lambda レイヤーとしてパッケージ化して、ワークロードを AWS Lambda 関数に移行します。データが S3 バケットにアップロードされたときに Lambda 関数を呼び出すように S3 イベント通知を構成します。

Answer: C

Explanation:

The workload is containerized, runs for hours, and is event-driven by nightly data arrival in Amazon S3. The current architecture uses EC2 instances and cron jobs, which results in operational overhead (managing instances, patching, scaling, scheduling) and idle compute between processing windows.

A key constraint is that the processing tasks can take hours. AWS Lambda has maximum execution duration limits that make it unsuitable for multi-hour batch processing. Even though Lambda can run container images, it still must complete within Lambda's runtime limit. Packaging container images as Lambda layers is also not an appropriate pattern for long-running container workloads and adds complexity.

A modern, low-ops approach for long-running, containerized batch jobs is to run containers on AWS Fargate.

Fargate removes the need to manage EC2 instances and allows tasks to run for extended periods as needed, scaling based on demand. Because the workload is composed of several data-processing services that likely need orchestration (for example, fan-out, sequencing, retries, parallelism), AWS Step Functions is well suited to coordinate the workflow and invoke the appropriate ECS tasks.

For triggering based on new S3 data, Amazon EventBridge provides a managed, scalable event bus for AWS service events, including S3 object events, and can route events to targets such as Step Functions state machines. Using EventBridge reduces the need for direct point-to-point notification wiring and provides centralized event routing, filtering, and monitoring.

Option C combines all the right elements: it runs the containers as ECS tasks on Fargate to eliminate EC2 management and idle capacity, uses Step Functions to orchestrate tasks that can run for hours, and uses EventBridge to trigger the state machine when new data is uploaded to S3. This replaces the per-instance cron scheduling with an event-driven serverless orchestration model and significantly reduces operational overhead.

Option B is close but is less appropriate as written because S3 Event Notifications are typically configured to send to Amazon SQS, Amazon SNS, or AWS Lambda. Triggering Step Functions directly is more naturally handled through EventBridge rules. EventBridge is also the recommended event routing layer for integrating service events into workflows.

Option A is not suitable because Lambda is not designed for multi-hour processing jobs due to runtime limits.

Option D is incorrect because Lambda layers are for sharing libraries and runtime dependencies, not for packaging multi-hour container workloads. It also still depends on Lambda runtime limits and does not match the operational model for long-running batch processing.

Therefore, option C is the best modernization approach with the least operational overhead.

References:AWS documentation on AWS Fargate for running container workloads without managing EC2 instances and supporting long-running tasks.AWS documentation on AWS Step Functions for orchestrating long-running workflows, retries, parallelism, and service

integrations including Amazon ECS.AWS documentation on Amazon EventBridge for routing Amazon S3 object events to targets such as Step Functions state machines for event-driven architectures.