

1. QUESTION

A Solutions Architect created a new Standard-class S3 bucket to store financial reports that are not frequently accessed but should immediately be available when an auditor requests them. To save costs, the Architect changed the storage class of the S3 bucket from Standard to Infrequent Access storage class.

In Amazon S3 Standard – Infrequent Access storage class, which of the following statements are true? (Select TWO.)

- It provides high latency and low throughput performance.
- It is designed for data that requires rapid access when needed.
- It automatically moves data to the most cost-effective access tier without any operational overhead.
- It is designed for data that is accessed less frequently.
- Ideal to use for data archiving.

Correct

Amazon S3 Standard – Infrequent Access (Standard – IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard – IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee.

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads



This combination of low cost and high performance make Standard – IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard – IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of Standard

- Designed for durability of 99.999999999% of objects
- Designed for 99.9% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL encryption of data in transit and at rest
- Lifecycle management for automatic migration of objects

Hence, the correct answers are:

- ***It is designed for data that is accessed less frequently.***
- ***It is designed for data that requires rapid access when needed.***

The option that says: ***It automatically moves data to the most cost-effective access tier without any operational overhead*** is incorrect as it actually refers to Amazon S3 – Intelligent Tiering, which is the only cloud storage class that delivers automatic cost savings by moving objects between different access tiers when access patterns change.

The option that says: ***It provides high latency and low throughput performance*** is incorrect as it should be “low latency” and “high throughput” instead. S3 automatically scales performance to meet user demands.

The option that says: ***Ideal to use for data archiving*** is incorrect because this statement refers to Amazon S3 Glacier. Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.

References:

<https://aws.amazon.com/s3/storage-classes/>

<https://aws.amazon.com/s3/faqs>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

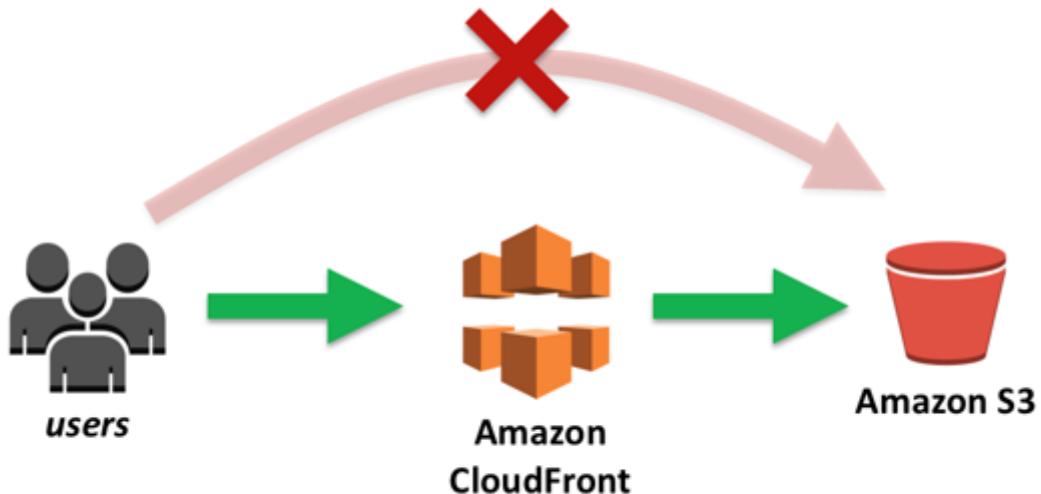
2. QUESTION

A Solutions Architect is working for a large global media company with multiple office locations all around the world. The Architect is instructed to build a system to distribute training videos to all employees. Using CloudFront, what method would be used to serve content that is stored in S3, but not publicly accessible from S3 directly?

- Add the CloudFront account security group.
- **Create an Origin Access Identity (OAI) for CloudFront and grant access to the objects in your S3 bucket to that OAI.**
 - Create an Identity and Access Management (IAM) user for CloudFront and grant access to the objects in your S3 bucket to that IAM user.
 - Create an S3 bucket policy that lists the CloudFront distribution ID as the principal and the target bucket as the Amazon Resource Name (ARN).

Correct

When you create or update a distribution in CloudFront, you can add an origin access identity (OAI) and automatically update the bucket policy to give the origin access identity permission to access your bucket. Alternatively, you can choose to manually change the bucket policy or change ACLs, which control permissions on individual objects in your bucket.



You can update the Amazon S3 bucket policy using either the AWS Management Console or the Amazon S3 API:

- Grant the CloudFront origin access identity the applicable permissions on the bucket.
- Deny access to anyone that you don't want to have access using Amazon S3 URLs.

Reference:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html#private-content-granting-permissions-to-oai>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

S3 Pre-signed URLs vs CloudFront Signed URLs vs Origin Access Identity (OAI)

<https://tutorialsdojo.com/s3-pre-signed-urls-vs-cloudfront-signed-urls-vs-origin-access-identity-oai/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

3. 3. QUESTION

A company hosted a web application in an Auto Scaling group of EC2 instances. The IT manager is concerned about the over-provisioning of the resources that can cause higher operating costs. A Solutions Architect has been instructed to create a cost-effective solution without affecting the performance of the application.

Which dynamic scaling policy should be used to satisfy this requirement?

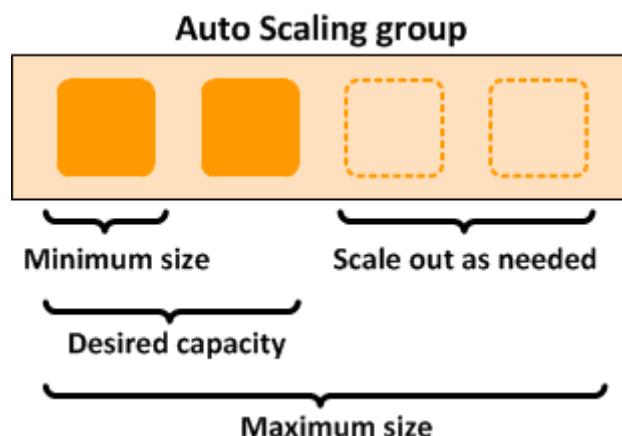
- Use simple scaling.

- Use scheduled scaling.
 - Use suspend and resume scaling.
 - Use target tracking scaling.

Incorrect

An **Auto Scaling group** contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service. The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

Step scaling policies and simple scaling policies are two of the dynamic scaling options available for you to use. Both require you to create CloudWatch alarms for the scaling policies. Both require you to specify the high and low thresholds for the alarms. Both require you to define whether to add or remove instances, and how many, or set the group to an exact size. The main difference between the policy types is the step adjustments that you get with step scaling policies. When step adjustments are applied, and they increase or decrease the current capacity of your Auto Scaling group, the adjustments vary based on the size of the alarm breach.



The primary issue with simple scaling is that after a scaling activity is started, the policy must wait for the scaling activity or health check replacement to complete and the cooldown period to expire before responding to additional alarms. Cooldown periods help to prevent the initiation of additional scaling activities before the effects of previous activities are visible.

With a target tracking scaling policy, you can increase or decrease the current capacity of the group based on a target value for a specific metric. This policy will help resolve the over-provisioning of your resources. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to changes in the metric due to a changing load pattern.

Hence, the correct answer is: ***Use target tracking scaling.***

The option that says: ***Use simple scaling*** is incorrect because **you need to wait for the cooldown period to complete before initiating additional scaling activities.** Target tracking or step scaling policies can trigger a scaling activity immediately without waiting for the cooldown period to expire.

The option that says: ***Use scheduled scaling*** is incorrect because **this policy is mainly used for predictable traffic patterns.** You need to use the target tracking scaling policy to optimize the cost of your infrastructure without affecting the performance.

The option that says: ***Use suspend and resume scaling*** is incorrect because **this type is used to temporarily pause scaling activities triggered by your scaling policies and scheduled actions.**

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

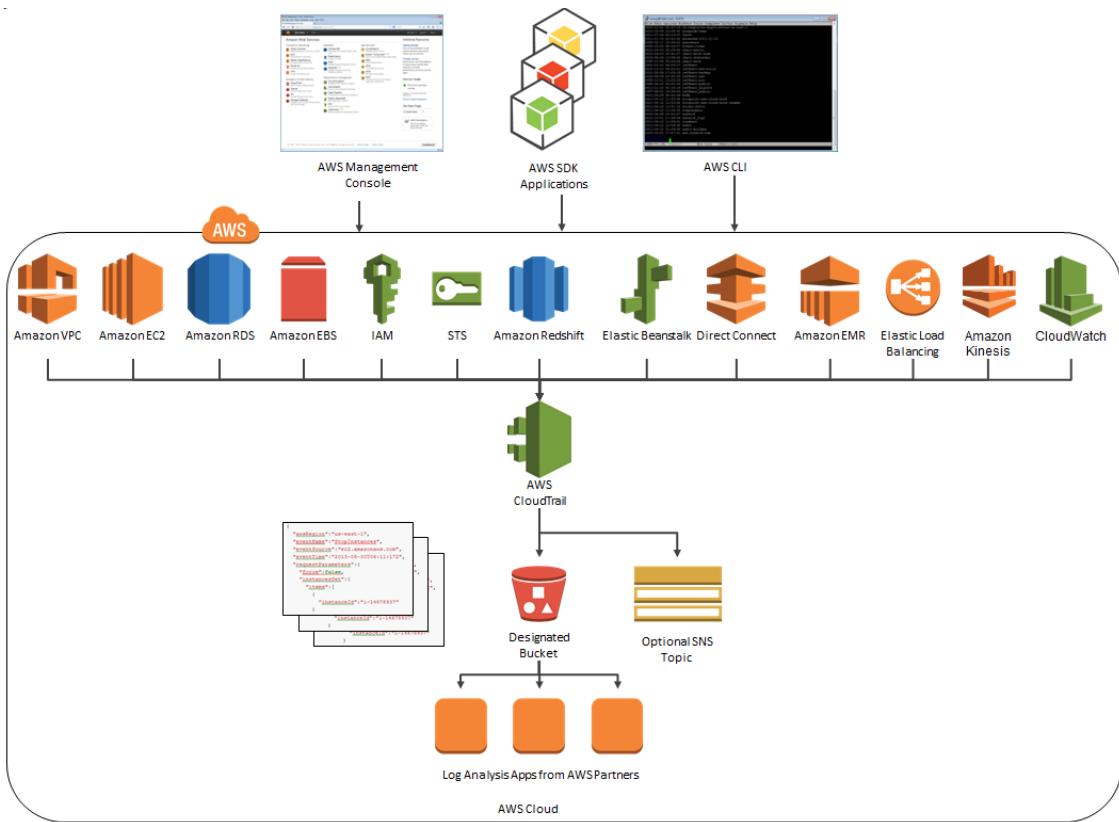
4. 4. QUESTION

A company needs to design an online analytics application that uses Redshift Cluster for its data warehouse. Which of the following services allows them to monitor all API calls in Redshift instance and can also provide secured data for auditing and compliance purposes?

- Amazon Redshift Spectrum
- AWS X-Ray
 - Amazon CloudWatch
 - **AWS CloudTrail**

Incorrect

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. By default, CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to Event history.



CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, API calls, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting.

Hence, the correct answer is: ***AWS CloudTrail.***

Amazon CloudWatch is incorrect. Although this is also a monitoring service, it cannot track the API calls to your AWS resources.

AWS X-Ray is incorrect because this is not a suitable service to use to track each API call to your AWS resources. It just helps you debug and analyze your microservices applications with request tracing so you can find the root cause of issues and performance.

Amazon Redshift Spectrum is incorrect because this is not a monitoring service but rather a feature of Amazon Redshift that enables you to query and analyze all of your data in Amazon S3 using the open data formats you already use, with no data loading or transformations needed.

References:

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

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>

Check out this AWS CloudTrail Cheat Sheet:

<https://tutorialsdojo.com/aws-cloudtrail/>

5. 5. QUESTION

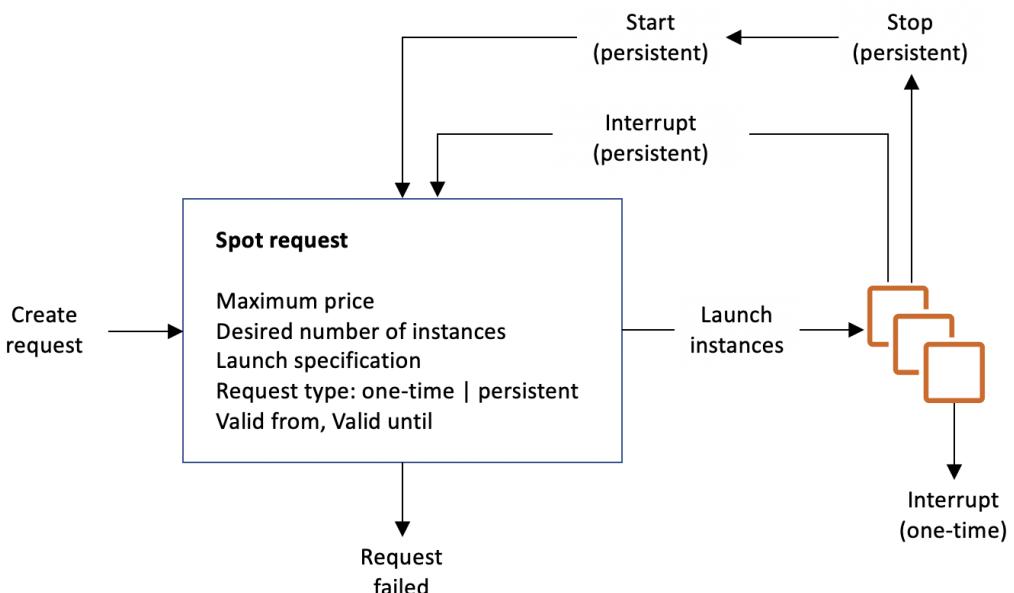
The media company that you are working for has a video transcoding application running on Amazon EC2. Each EC2 instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based on the queuing system. This application has a large backlog of videos which need to be transcoded. Your manager would like to reduce this backlog by adding more EC2 instances, however, these instances are only needed until the backlog is reduced.

In this scenario, which type of Amazon EC2 instance is the most cost-effective type to use?

- On-demand instances
- Reserved instances
- Spot instances
- Dedicated instances

Incorrect

You require an instance that will be used not as a primary server but as a spare compute resource to augment the transcoding process of your application. These instances should also be terminated once the backlog has been significantly reduced. In addition, the scenario mentions that if the current process is interrupted, the video can be transcoded by another instance based on the queuing system. This means that the application can gracefully handle an unexpected termination of an EC2 instance, like in the event of a Spot instance termination when the Spot price is greater than your set maximum price. Hence, an Amazon EC2 Spot instance is the best and cost-effective option for this scenario.



Amazon EC2 Spot instances are **spare** compute capacity in the AWS cloud available to you at steep discounts compared to On-Demand prices. EC2 Spot enables you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget. By simply selecting Spot when launching EC2 instances, you

can save up-to 90% on On-Demand prices. The only difference between **On-Demand instances** and **Spot Instances** is that Spot instances can be interrupted by EC2 with two minutes of notification when the EC2 needs the capacity back.

You can specify whether Amazon EC2 should hibernate, stop, or terminate Spot Instances when they are interrupted. You can choose the interruption behavior that meets your needs.

Take note that there is no “*bid price*” anymore for Spot EC2 instances **since March 2018**. You simply have to set your **maximum price** instead.

Reserved instances and ***Dedicated instances*** are incorrect as both do not act as spare compute capacity.

On-demand instances is a valid option but a Spot instance is much cheaper than **On-Demand**.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html>

<https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

6. QUESTION

A financial application is composed of an Auto Scaling group of EC2 instances, an Application Load Balancer, and a MySQL RDS instance in a Multi-AZ Deployments configuration. To protect the confidential data of your customers, you have to ensure that your RDS database can only be accessed using the profile credentials specific to your EC2 instances via an authentication token.

As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- Use a combination of IAM and STS to restrict access to your RDS instance via a temporary token.
- Enable the IAM DB Authentication.
- Create an IAM Role and assign it to your EC2 instances which will grant exclusive access to your RDS instance.
- Configure SSL in your application to encrypt the database connection to RDS.

Incorrect

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don’t need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An **authentication token** is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

Database options

DB cluster identifier [Info](#)
tutorialsdojo
If you do not provide one, a default identifier based on the instance identifier will be used.

Database name [Info](#)
tutorialsdojo
If you do not specify a database name, Amazon RDS does not create a database.

Port [Info](#)
TCP/IP port the DB instance will use for application connections.
3306

DB parameter group [Info](#)
default.aurora5.6

DB cluster parameter group [Info](#)
default.aurora5.6

Option group [Info](#)
default:aurora-5-6

IAM DB authentication [Info](#)
 Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
 Disable

IAM database authentication provides the following benefits:

1. Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
2. You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.
3. For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security.

Hence, ***enabling IAM DB Authentication*** is the correct answer based on the above reference.

Configuring SSL in your application to encrypt the database connection to RDS is incorrect because an SSL connection is not using an authentication token from IAM. Although configuring SSL to your application can improve the security of your data in flight, it is still not a suitable option to use in this scenario.

Creating an IAM Role and assigning it to your EC2 instances which will grant exclusive access to your RDS instance is incorrect because although you can create and assign an IAM Role to your EC2 instances, you still need to configure your RDS to use IAM DB Authentication.

Using a combination of IAM and STS to restrict access to your RDS instance via a temporary token is incorrect because you have to use IAM DB Authentication for this scenario, and not a combination of an IAM and STS. Although **STS is used to send temporary tokens for authentication**, this is not a compatible use case for RDS.

Reference:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html>

Check out this Amazon RDS cheat sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

6. 7. QUESTION

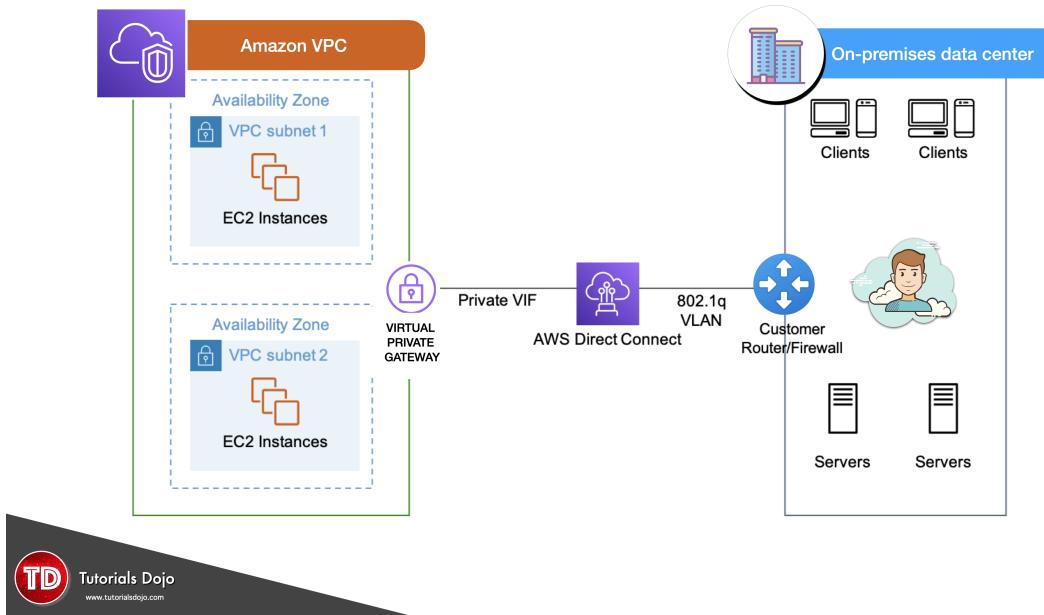
A company plans to implement a hybrid architecture. They need to create a dedicated connection from their Amazon Virtual Private Cloud (VPC) to their on-premises network. The connection must provide high bandwidth throughput and a more consistent network experience than Internet-based solutions.

Which of the following can be used to create a private connection between the VPC and the company's on-premises network?

- AWS Site-to-Site VPN
- Transit Gateway with equal-cost multipath routing (ECMP)
- Transit VPC
- AWS Direct Connect

Correct

AWS Direct Connect links your internal network to an AWS Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to an AWS Direct Connect router.



With this connection, you can create virtual interfaces directly to public AWS services (for example, to Amazon S3) or to Amazon VPC, bypassing internet service providers in your network path. An AWS Direct Connect location provides access to AWS in the region with which it is associated. You can use a single connection in a public Region or AWS GovCloud (US) to access public AWS services in all other public Regions.

Hence, the correct answer is: ***AWS Direct Connect.***

The option that says: ***Transit VPC*** is incorrect because this in itself is not enough to integrate your on-premises network to your VPC. You have to either use a VPN or a Direct Connect connection. A **transit VPC is primarily used to connect multiple VPCs and remote networks in order to create a global network transit center** and not for establishing a dedicated connection to your on-premises network.

The option that says: ***Transit Gateway with equal-cost multipath routing (ECMP)*** is incorrect because a **transit gateway is commonly used to connect multiple VPCs and on-premises networks through a central hub. Just like transit VPC, a transit gateway is not capable of establishing a direct and dedicated connection to your on-premises network.**

The option that says: ***AWS Site-to-Site VPN*** is incorrect because this type of connection traverses the public Internet. Moreover, it doesn't provide a high bandwidth throughput and a more consistent network experience than Internet-based solutions.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/connect-vpc/>

<https://docs.aws.amazon.com/directconnect/latest/UserGuide/Welcome.html>

Check out this AWS Direct Connect Cheat Sheet:

<https://tutorialsdojo.com/aws-direct-connect/>

S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball vs Snowmobile:

<https://tutorialsdojo.com/s3-transfer-acceleration-vs-direct-connect-vs-vpn-vs-snowball-vs-snowmobile/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

8. QUESTION

There was an incident in your production environment where the user data stored in the S3 bucket has been accidentally deleted by one of the Junior DevOps Engineers. The issue was escalated to your manager and after a few days, you were instructed to improve the security and protection of your AWS resources.

What combination of the following options will protect the S3 objects in your bucket from both accidental deletion and overwriting? (Select TWO.)

- **Enable Versioning**
- Provide access to S3 data strictly through pre-signed URL only
- **Enable Multi-Factor Authentication Delete**
- Enable Amazon S3 Intelligent-Tiering
- Disallow S3 Delete using an IAM bucket policy

Correct

By using Versioning and enabling MFA (Multi-Factor Authentication) Delete, you can secure and recover your S3 objects from accidental deletion or overwrite.

Versioning is a means of keeping multiple variants of an object in the same bucket. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

You can also optionally add another layer of security by configuring a bucket to enable **MFA (Multi-Factor Authentication) Delete**, which **requires additional authentication for** either of the following operations:

- Change the versioning state of your bucket
- Permanently delete an object version

MFA Delete requires two forms of authentication together:

- Your security credentials
- The concatenation of a valid serial number, a space, and the six-digit code displayed on an approved authentication device

Providing access to S3 data strictly through pre-signed URL only is incorrect since a pre-signed URL gives access to the object identified in the URL. Pre-signed URLs are useful when customers perform an object upload to your S3 bucket, but does not help in preventing accidental deletes.

Disallowing S3 Delete using an IAM bucket policy is incorrect since you still want users to be able to delete objects in the bucket, and you just want to prevent accidental deletions. Disallowing S3 Delete using an IAM bucket policy will restrict all delete operations to your bucket.

Enabling Amazon S3 Intelligent-Tiering is incorrect since S3 intelligent tiering does not help in this situation.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

7.9. QUESTION

A company plans to set up a cloud infrastructure in AWS. In the planning, it was discussed that you need to deploy two EC2 instances that should continuously run for three years. The CPU utilization of the EC2 instances is also expected to be stable and predictable.

Which is the most cost-efficient Amazon EC2 Pricing type that is most appropriate for this scenario?

- Reserved Instances
- On-Demand instances
- Spot instances
- Dedicated Hosts

Correct

Reserved Instances provide you with a significant discount (up to 75%) compared to **On-Demand instance pricing**. In addition, when Reserved Instances are assigned to a specific Availability Zone, they provide a capacity reservation, giving you additional confidence in your ability to launch instances when you need them.

The screenshot shows the 'Purchase Reserved Instances' interface. At the top, there's a search bar and a checkbox for 'Only show offerings that reserve capacity'. Below that is a filtering section with dropdowns for Platform (Linux/UNIX), Tenancy (Default), Offering Class (Convertible), Instance Type (c4.large), Term (Any), Payment Option (Any), and a 'Search' button. The main table lists three AWS offerings for c4.large instances over 36 months:

Seller	Term	Effective Rate	Upfront Price	Hourly Rate	Payment Option	Offering Class	Quantity Available	Desired Quantity	Add to Cart
AWS	36 months	\$0.059	\$1,555.00	\$0.000	All Upfront	convertible	Unlimited	1	Add to Cart
AWS	36 months	\$0.060	\$797.00	\$0.030	Partial Upfront	convertible	Unlimited	1	Add to Cart
AWS	36 months	\$0.070	\$0.00	\$0.070	No Upfront	convertible	Unlimited	1	Add to Cart

At the bottom, a message says 'You currently have no items in your cart.' with 'Cancel' and 'View Cart' buttons.

For applications that have steady state or predictable usage, Reserved Instances can provide significant savings compared to using On-Demand instances.

Reserved Instances are recommended for:

- Applications with steady state usage
- Applications that may require reserved capacity
- Customers that can commit to using EC2 over a 1 or 3 year term to reduce their total computing costs

References:

<https://aws.amazon.com/ec2/pricing/>

<https://aws.amazon.com/ec2/pricing/reserved-instances/>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

10. QUESTION

One member of your DevOps team consulted you about a connectivity problem in one of your Amazon EC2 instances. The application architecture is initially set up with four EC2 instances, each with an EIP address that all belong to a public non-default subnet. You launched another instance to handle the increasing workload of your application. The EC2 instances also belong to the same security group. Everything works well as expected except for one of the EC2 instances which is not able to send nor receive traffic over the Internet.

Which of the following is the MOST likely reason for this issue?

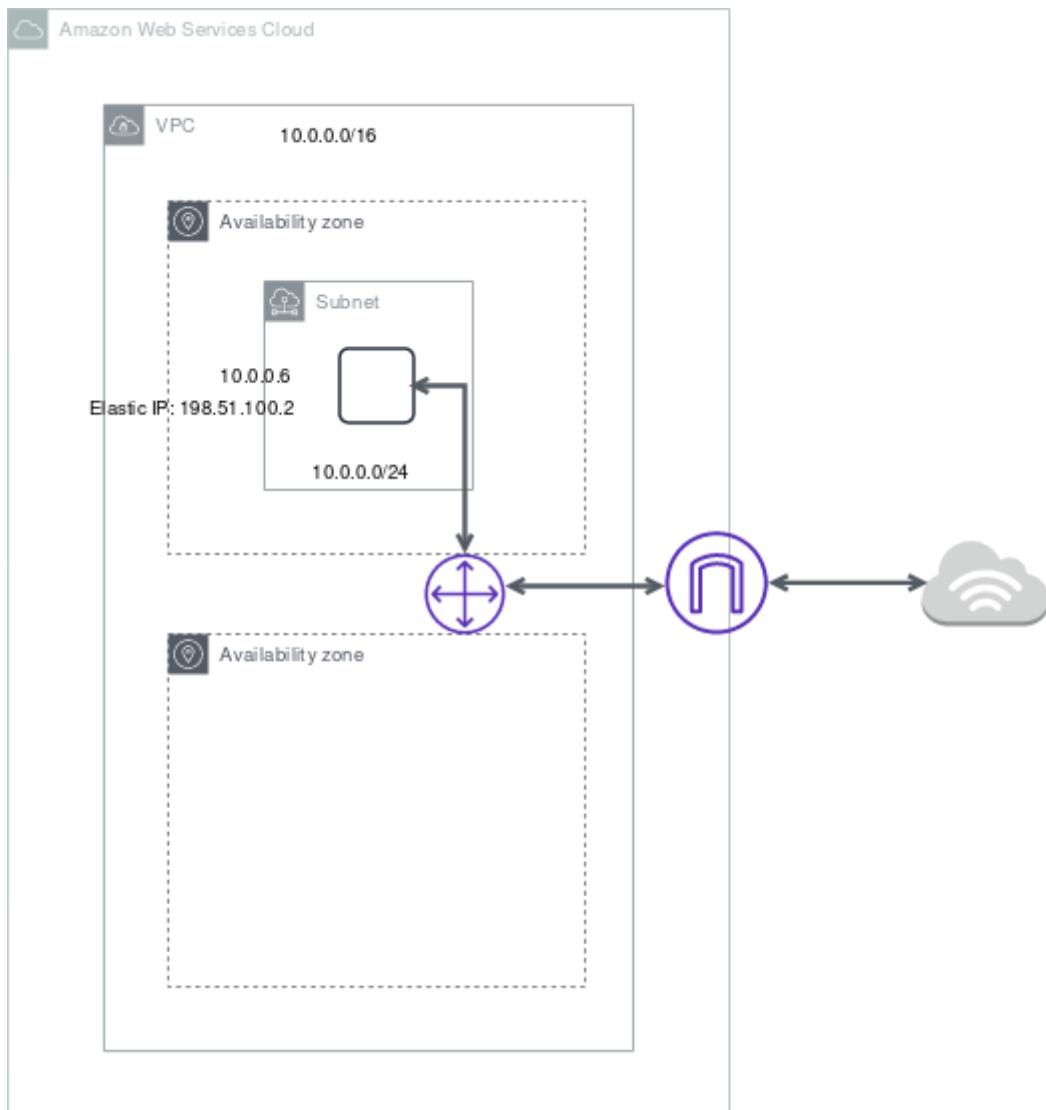
- The EC2 instance does not have a public IP address associated with it.
- The EC2 instance is running in an Availability Zone that is not connected to an Internet gateway.
- The route table is not properly configured to allow traffic to and from the Internet through the Internet gateway.
- The EC2 instance does not have a private IP address associated with it.

Incorrect

IP addresses enable resources in your VPC to communicate with each other, and with resources over the Internet. Amazon EC2 and Amazon VPC support the IPv4 and IPv6 addressing protocols.

By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol. When you create a VPC, you must assign it an IPv4 CIDR block (a range of private IPv4 addresses). Private IPv4 addresses are not reachable over the Internet. To connect to your instance over the Internet, or to enable communication between your instances and other AWS services that have public endpoints, you can assign a globally-unique public IPv4 address to your instance.

You can optionally associate an IPv6 CIDR block with your VPC and subnets, and assign IPv6 addresses from that block to the resources in your VPC. IPv6 addresses are public and reachable over the Internet.



All subnets have a modifiable attribute that determines whether a network interface created in that subnet is assigned a public IPv4 address and, if applicable, an IPv6 address. This includes the primary network interface (eth0) that's created for an instance when you launch an instance in that subnet. Regardless of the subnet attribute, you can still override this setting for a specific instance during launch.

By default, nondefault subnets have the IPv4 public addressing attribute set to `false`, and default subnets have this attribute set to `true`. An exception is a nondefault subnet created by the Amazon EC2 launch instance wizard — the wizard sets the attribute to `true`. You can modify this attribute using the Amazon VPC console.

In this scenario, there are 5 EC2 instances that belong to the same security group that should be able to connect to the Internet. The main route table is properly configured but there is a problem connecting to one instance. Since the other four instances are working fine, we can assume that the security group and the route table are correctly configured. One possible reason for this issue is that the problematic instance does not have a public or an EIP address.

Take note as well that the four EC2 instances all belong to a public **non-default** subnet. Which means that a new EC2 instance will not have a public IP address by default since the since IPv4 public addressing attribute is initially set to `false`.

Hence, the correct answer is the option that says: ***The EC2 instance does not have a public IP address associated with it.***

The option that says: ***The route table is not properly configured to allow traffic to and from the Internet through the Internet gateway*** is incorrect because the other three instances, which are associated with the same route table and security group, do not have any issues.

The option that says: ***The EC2 instance is running in an Availability Zone that is not connected to an Internet gateway*** is incorrect because there is no relationship between the Availability Zone and the Internet Gateway (IGW) that may have caused the issue.

References:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenario1.html

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html#vpc-ip-addressing-subnet>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

11. QUESTION

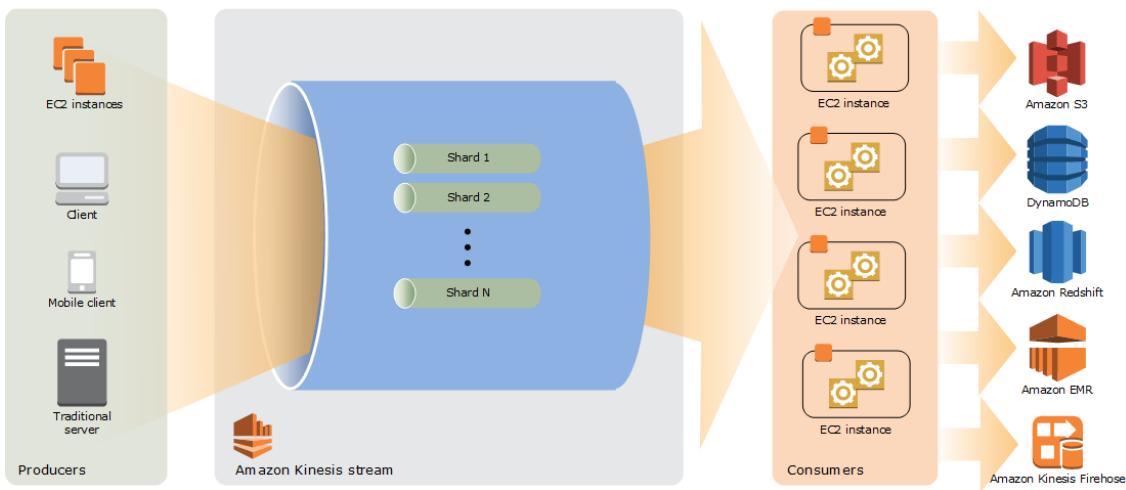
A startup is building IoT devices and monitoring applications. They are using IoT sensors to monitor the traffic in real-time by using an Amazon Kinesis Stream that is configured with default settings. It then sends the data to an Amazon S3 bucket every 3 days. When you checked the data in S3 on the 3rd day, only the data for the last day is present and no data is present from 2 days ago.

Which of the following is the MOST likely cause of this issue?

- Amazon S3 bucket has encountered a data loss.
- The access of the Kinesis stream to the S3 bucket is insufficient.
- Someone has manually deleted the record in Amazon S3.
- By default, data records in Kinesis are only accessible for 24 hours from the time they are added to a stream.

Incorrect

By default, records of a stream in Amazon Kinesis are accessible for up to 24 hours from the time they are added to the stream. You can raise this limit to up to 7 days by enabling extended data retention.



Hence, the correct answer is: ***By default, data records in Kinesis are only accessible for 24 hours from the time they are added to a stream.***

The option that says: ***Amazon S3 bucket has encountered a data loss*** is incorrect because Amazon S3 rarely experiences data loss. Amazon has an SLA for S3 that it commits to its customers. Amazon S3 Standard, S3 Standard-IA, S3 One Zone-IA, and S3 Glacier are all designed to provide 99.99999999% durability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.00000001% of objects. Hence, Amazon S3 bucket data loss is highly unlikely.

The option that says: ***Someone has manually deleted the record in Amazon S3*** is incorrect because if someone has deleted the data, this should have been visible in CloudTrail. Also, deleting that much data manually shouldn't have occurred in the first place if you have put in the appropriate security measures.

The option that says: ***The access of the Kinesis stream to the S3 bucket is insufficient*** is incorrect because having insufficient access is highly unlikely since you are able to access the bucket and view the contents of the previous day's data collected by Kinesis.

Reference:

<https://aws.amazon.com/kinesis/data-streams/faqs/>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/DataDurability.html>

Check out this Amazon Kinesis Cheat Sheet:

<https://tutorialsdojo.com/amazon-kinesis/>

12. QUESTION

A Solutions Architect working for a startup is designing a High Performance Computing (HPC) application which is publicly accessible for their customers. The startup founders want to mitigate distributed denial-of-service (DDoS) attacks on their application.

Which of the following options are not suitable to be implemented in this scenario? (Select TWO.)

- Add multiple Elastic Fabric Adapters (EFA) to each EC2 instance to increase the network bandwidth.

- Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by deploying it to a new private subnet.
 - Use Dedicated EC2 instances to ensure that each instance has the maximum performance possible.
 - Use AWS Shield and AWS WAF.
 - Use an Amazon CloudFront service for distributing both static and dynamic content.

Incorrect

Take note that the question asks about the viable mitigation techniques that are **NOT** suitable to prevent Distributed Denial of Service (DDoS) attack.

A Denial of Service (DoS) attack is an attack that can make your website or application unavailable to end users. To achieve this, attackers use a variety of techniques that consume network or other resources, disrupting access for legitimate end users.

To protect your system from DDoS attack, you can do the following:

- Use an Amazon CloudFront service for distributing both static and dynamic content.
- Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by deploying it to a new private subnet.
- Set up alerts in Amazon CloudWatch to look for high Network In and CPU utilization metrics.

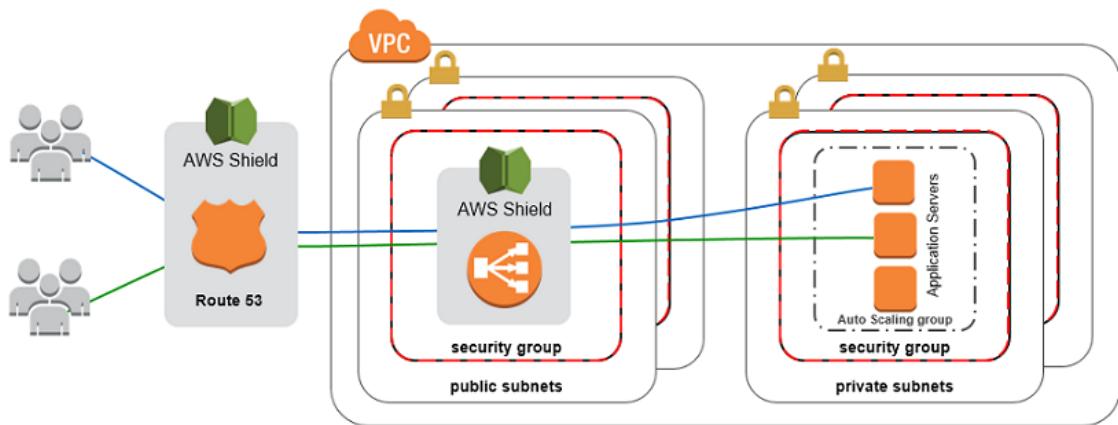
Services that are available within AWS Regions, like Elastic Load Balancing and Amazon Elastic Compute Cloud (EC2), allow you to build Distributed Denial of Service resiliency and scale to handle unexpected volumes of traffic within a given region. Services that are available in AWS edge locations, like Amazon CloudFront, AWS WAF, Amazon Route53, and Amazon API Gateway, allow you to take advantage of a global network of edge locations that can provide your application with greater fault tolerance and increased scale for managing larger volumes of traffic.

In addition, you can also use **AWS Shield** and **AWS WAF** to fortify your cloud network.

AWS Shield is a managed DDoS protection service that is available in two tiers:

Standard and Advanced. **AWS Shield Standard** applies always-on detection and inline mitigation techniques, such as deterministic packet filtering and priority-based traffic shaping, to minimize application downtime and latency.

AWS WAF is a web application firewall that helps protect web applications from common web exploits that could affect application availability, compromise security, or consume excessive resources. You can use AWS WAF to define customizable web security rules that control which traffic accesses your web applications. If you use AWS Shield Advanced, you can use AWS WAF at no extra cost for those protected resources and can engage the DRT to create WAF rules.



Using Dedicated EC2 instances to ensure that each instance has the maximum performance possible is not a viable mitigation technique because Dedicated EC2 instances are just an instance billing option. Although it may ensure that each instance gives the maximum performance, that by itself is not enough to mitigate a DDoS attack.

Adding multiple Elastic Fabric Adapters (EFA) to each EC2 instance to increase the network bandwidth is also not a viable option as this is mainly done for performance improvement, and not for DDoS attack mitigation. Moreover, you can attach only one EFA per EC2 instance. An Elastic Fabric Adapter (EFA) is a network device that you can attach to your Amazon EC2 instance to accelerate High-Performance Computing (HPC) and machine learning applications.

The following options are valid mitigation techniques that can be used to prevent DDoS:

- *- Use an Amazon CloudFront service for distributing both static and dynamic content.***
- *- Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by deploying it to a new private subnet.***
- *- Use AWS Shield and AWS WAF.***

References:

<https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/>

https://d0.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf

Best practices on DDoS Attack Mitigation:

13. QUESTION

A company is working with a government agency to improve traffic planning and maintenance of roadways to prevent accidents. The proposed solution is to manage the traffic infrastructure in real-time, alert traffic engineers and emergency response teams when problems are detected, and automatically change traffic signals to get emergency personnel to accident scenes faster by using sensors and smart devices.

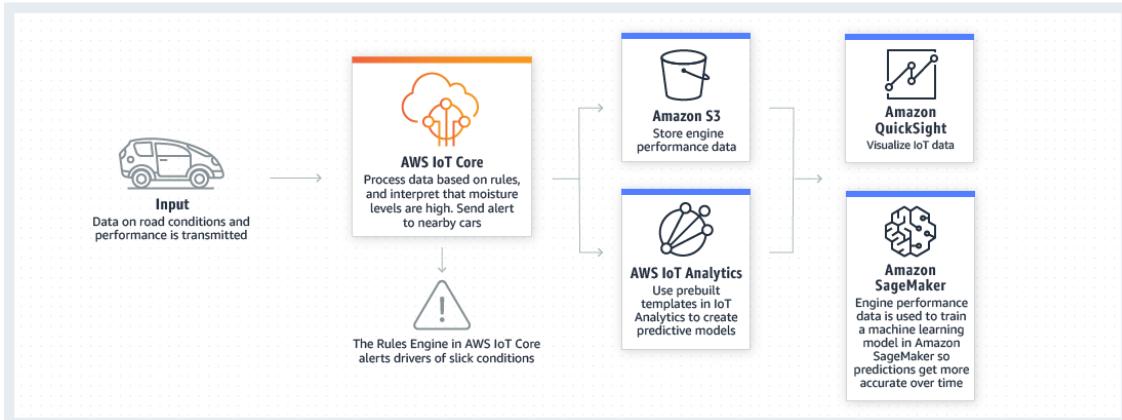
Which AWS service will allow the developers of the agency to connect the smart devices to the cloud-based applications?

- AWS CloudFormation

- AWS Elastic Beanstalk
- **AWS IoT Core**
- Amazon Elastic Container Service

Incorrect

AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT Core provides secure communication and data processing across different kinds of connected devices and locations so you can easily build IoT applications.



AWS IoT Core allows you to connect multiple devices to the cloud and to other devices without requiring you to deploy or manage any servers. You can also filter, transform, and act upon device data on the fly based on the rules you define. With AWS IoT Core, your applications can keep track of and communicate with all of your devices, all the time, even when they aren't connected.

Hence, the correct answer is: ***AWS IoT Core.***

AWS CloudFormation is incorrect because this is mainly used for creating and managing the architecture and not for handling connected devices. You have to use AWS IoT Core instead.

AWS Elastic Beanstalk is incorrect because this is just an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, and other programming languages. Elastic Beanstalk can't be used to connect smart devices to cloud-based applications.

Amazon Elastic Container Service is incorrect because this is mainly used for creating and managing docker instances and not for handling devices.

References:

<https://aws.amazon.com/iot-core/>

<https://aws.amazon.com/iot/>

14. QUESTION

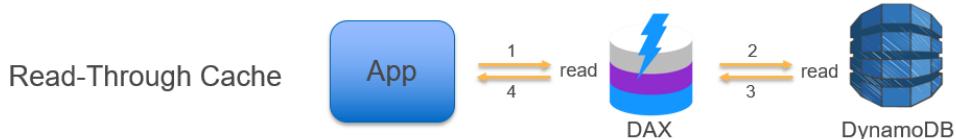
A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time.

To improve the game's performance, which AWS service can you use to reduce DynamoDB response times from milliseconds to microseconds?

- Amazon DynamoDB Accelerator (DAX)
- Amazon ElastiCache
 - AWS Device Farm
 - DynamoDB Auto Scaling

Incorrect

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache that can reduce Amazon DynamoDB response times from milliseconds to microseconds, even at millions of requests per second.



Amazon ElastiCache is incorrect because although you may use ElastiCache as your database cache, it will not reduce the DynamoDB response time from milliseconds to microseconds as compared with DynamoDB DAX.

AWS Device Farm is incorrect because this is an app testing service that lets you test and interact with your Android, iOS, and web apps on many devices at once, or reproduce issues on a device in real time.

DynamoDB Auto Scaling is incorrect because this is primarily used to automate capacity management for your tables and global secondary indexes.

References:

<https://aws.amazon.com/dynamodb/dax>

<https://aws.amazon.com/device-farm>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

15. QUESTION

A company is using Amazon VPC that has a CIDR block of 10.31.0.0/27 that is connected to the on-premises data center. There was a requirement to create a Lambda function that will process massive amounts of cryptocurrency transactions every minute and then store the results to EFS. After setting up the serverless architecture and connecting the Lambda function to the VPC, the Solutions Architect noticed an increase in invocation errors with EC2 error types such as EC2ThrottledException at certain times of the day.

Which of the following are the possible causes of this issue? (Select TWO.)

- The attached IAM execution role of your function does not have the necessary permissions to access the resources of your VPC.

- The associated security group of your function does not allow outbound connections.
- Your VPC does not have sufficient subnet ENIs or subnet IPs.
- Your VPC does not have a NAT gateway.
- You only specified one subnet in your Lambda function configuration. That single subnet runs out of available IP addresses and there is no other subnet or Availability Zone which can handle the peak load.

Incorrect

You can configure a function to connect to a virtual private cloud (VPC) in your account. Use Amazon Virtual Private Cloud (Amazon VPC) to create a private network for resources such as databases, cache instances, or internal services. Connect your function to the VPC to access private resources during execution.

AWS Lambda runs your function code securely within a VPC by default. However, to enable your Lambda function to access resources inside your private VPC, you must provide additional VPC-specific configuration information that includes VPC subnet IDs and security group IDs. AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function to connect securely to other resources within your private VPC.

Lambda functions cannot connect directly to a VPC with dedicated instance tenancy. To connect to resources in a dedicated VPC, peer it to a second VPC with default tenancy.

Your Lambda function automatically scales based on the number of events it processes. If your Lambda function accesses a VPC, you must make sure that your VPC has sufficient ENI capacity to support the scale requirements of your Lambda function. It is also recommended that you specify at least one subnet in each Availability Zone in your Lambda function configuration.

By specifying subnets in each of the Availability Zones, your Lambda function can run in another Availability Zone if one goes down or runs out of IP addresses. If your VPC does not have sufficient ENIs or subnet IPs, your Lambda function will not scale as requests increase, and you will see an increase in invocation errors with EC2 error types like `EC2ThrottledException`. For asynchronous invocation, if you see an increase in errors without corresponding CloudWatch Logs, invoke the Lambda function synchronously in the console to get the error responses.

Hence, the correct answers for this scenario are:

- You only specified one subnet in your Lambda function configuration. That single subnet runs out of available IP addresses and there is no other subnet or Availability Zone which can handle the peak load.

- Your VPC does not have sufficient subnet ENIs or subnet IPs.

The screenshot shows the AWS Lambda function configuration interface. It consists of three main sections:

- Execution role**: A dropdown menu titled "Use an existing role" contains the option "service-role/tutorialsdojo-lambda-vpc-role-xd5u9vhy". Below the dropdown is a link to "View the tutorialsdojo-lambda-vpc-role-xd5u9vhy role on the IAM console".
- Network**: A dropdown menu titled "Choose a VPC for your function to access" has the option "No VPC" selected.
- Concurrency**: Shows "Unreserved account concurrency 1000" and two radio button options: "Use unreserved account concurrency" (selected) and "Reserve concurrency".

The option that says: ***Your VPC does not have a NAT gateway*** is incorrect because an issue in the NAT Gateway is unlikely to cause a request throttling issue or produce an `EC2ThrottledException` error in Lambda. As per the scenario, the issue is happening only at certain times of the day, which means that the issue is only intermittent and the function works at other times. We can also conclude that an availability issue is not an issue since the application is already using a highly available NAT Gateway and not just a NAT instance.

The option that says: ***The associated security group of your function does not allow outbound connections*** is incorrect because if the associated security group does not allow outbound connections then the Lambda function will not work at all in the first place. Remember that as per the scenario, the issue only happens intermittently. In addition, Internet traffic restrictions do not usually produce `EC2ThrottledException` errors.

The option that says: ***The attached IAM execution role of your function does not have the necessary permissions to access the resources of your VPC*** is incorrect because just as what is explained above, the issue is intermittent and thus, the IAM execution role of the function does have the necessary permissions to access the resources of the VPC since it works at those specific times. In case the issue is indeed caused by a permission problem then an `EC2AccessDeniedException` the error would most likely be returned and not an `EC2ThrottledException` error.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/vpc.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/>

<https://aws.amazon.com/premiumsupport/knowledge-center/lambda-troubleshoot-invocation-error-502-500/>

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

<https://portal.tutorialsdojo.com/courses/free-aws-certified-solutions-architect-associate-practice-exams-sampler/lessons/free-practice-exam-timed-mode-4/quizzes/free-aws-certified-solutions-architect-associate-practice-exam-timed-mode/>

1. QUESTION

Category: CSAA – Design Resilient Architectures

A data analytics company keeps a massive volume of data that they store in their on-premises data center. To scale their storage systems, they are looking for cloud-backed storage volumes that they can mount using Internet Small Computer System Interface (iSCSI) devices from their on-premises application servers. They have an on-site data analytics application that frequently accesses the latest data subsets locally while the older data are rarely accessed. You are required to minimize the need to scale the on-premises storage infrastructure while still providing their web application with low-latency access to the data.

Which type of AWS Storage Gateway service will you use to meet the above requirements?

- Volume Gateway in stored mode
- Tape Gateway
- File Gateway
- **Volume Gateway in cached mode**

Correct

In this scenario, the technology company is looking for a storage service that will enable their analytics application to frequently access the latest data subsets and not the entire data set (as it was mentioned that the old data are rarely being used). This requirement can be fulfilled by setting up a Cached Volume Gateway in AWS Storage Gateway.

By using cached volumes, you can use Amazon S3 as your primary data storage, while retaining frequently accessed data locally in your storage gateway. Cached volumes minimize the need to scale your on-premises storage infrastructure, while still providing your applications with low-latency access to frequently accessed data. You can create storage volumes up to 32 TiB in size and afterward, attach these volumes as iSCSI devices to your on-premises application servers. When you write to these volumes, your gateway stores the data in Amazon S3. It retains the recently read data in your on-premises storage gateway's cache and uploads buffer storage.

Cached volumes can range from 1 GiB to 32 TiB in size and must be rounded to the nearest GiB. Each gateway configured for cached volumes can support up to 32 volumes for a total maximum storage volume of 1,024 TiB (1 PiB).

In the cached volumes solution, AWS Storage Gateway stores all your on-premises application data in a storage volume in Amazon S3. Hence, the correct answer is: ***Volume Gateway in cached mode.***

Volume Gateway in stored mode is incorrect because the requirement is to provide low latency access to the frequently accessed data subsets locally. Stored Volumes are used if you need low-latency access to your entire dataset.

Tape Gateway is incorrect because this is just a cost-effective, durable, long-term offsite alternative for data archiving, which is not needed in this scenario.

File Gateway is incorrect because the scenario requires you to mount volumes as iSCSI devices. File Gateway is used to store and retrieve Amazon S3 objects through NFS and SMB protocols.

References:

<https://docs.aws.amazon.com/storagegateway/latest/userguide/StorageGatewayConcepts.html#volume-gateway-concepts>

<https://docs.aws.amazon.com/storagegateway/latest/userguide/WhatIsStorageGateway.html>

AWS Storage Gateway Overview:

Check out this AWS Storage Gateway Cheat Sheet:

<https://tutorialsdojo.com/aws-storage-gateway/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate-saa-c02/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

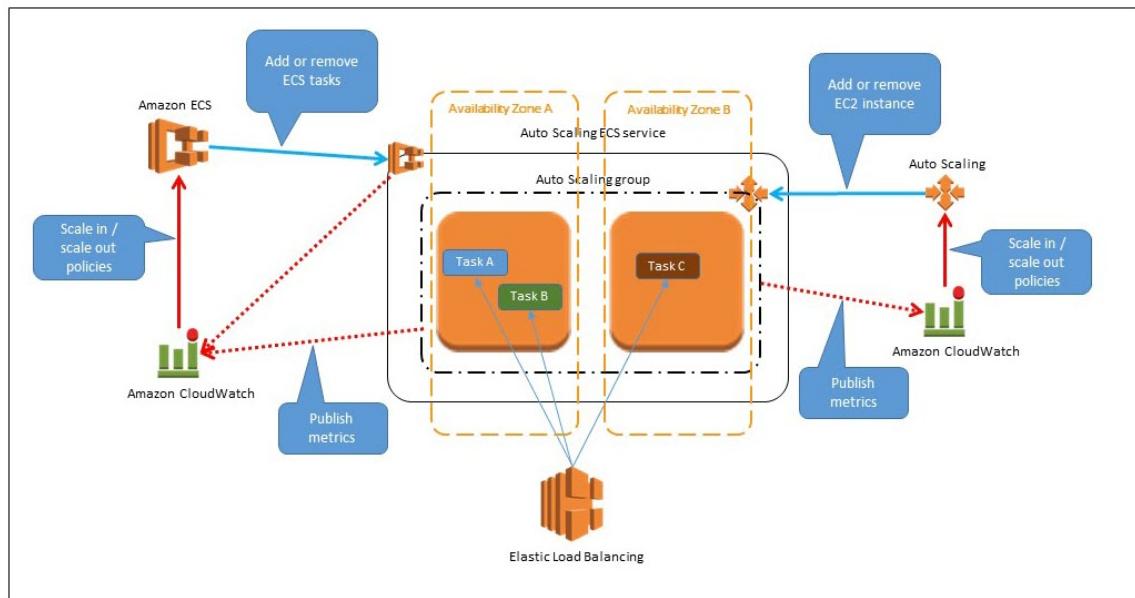
A loan processing application is hosted in a single On-Demand EC2 instance in your VPC. To improve the scalability of your application, you have to use Auto Scaling to automatically add new EC2 instances to handle a surge of incoming requests.

Which of the following items should be done in order to add an existing EC2 instance to an Auto Scaling group? (Select TWO.)

- You have to ensure that the instance is launched in one of the Availability Zones defined in your Auto Scaling group.
- You have to ensure that the instance is in a different Availability Zone as the Auto Scaling group.
- You have to ensure that the AMI used to launch the instance no longer exists.
- You have to ensure that the AMI used to launch the instance still exists.
- You must stop the instance first.

Incorrect

Amazon EC2 Auto Scaling provides you with an option to enable automatic scaling for one or more EC2 instances by attaching them to your existing Auto Scaling group. After the instances are attached, they become a part of the Auto Scaling group



The instance that you want to attach must meet the following criteria:

- The instance is in the `running` state.
- The AMI used to launch the instance must still exist.
- The instance is not a member of another Auto Scaling group.
- The instance is launched into one of the Availability Zones defined in your Auto Scaling group.
- If the Auto Scaling group has an attached load balancer, the instance and the load balancer must both be in EC2-Classic or the same VPC. If the Auto Scaling group has an attached target group, the instance and the load balancer must both be in the same VPC.

Based on the above criteria, the following are the correct answers among the given options:

- You have to ensure that the AMI used to launch the instance still exists.

– You have to ensure that the instance is launched in one of the Availability Zones defined in your Auto Scaling group.

The option that says: ***You must stop the instance first*** is incorrect because you can directly add a running EC2 instance to an Auto Scaling group without stopping it.

The option that says: ***You have to ensure that the AMI used to launch the instance no longer exists*** is incorrect because it should be the other way around. The AMI used to launch the instance should still exist.

The option that says: ***You have to ensure that the instance is in a different Availability Zone as the Auto Scaling group*** is incorrect because the instance should be launched in one of the Availability Zones defined in your Auto Scaling group.

References:

<http://docs.aws.amazon.com/autoscaling/latest/userguide/attach-instance-asg.html>

https://docs.aws.amazon.com/autoscaling/ec2/userguide/scaling_plan.html

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A media company is using Amazon EC2, ELB, and S3 for its video-sharing portal for filmmakers. They are using a standard S3 storage class to store all high-quality videos that are frequently accessed only during the first three months of posting.

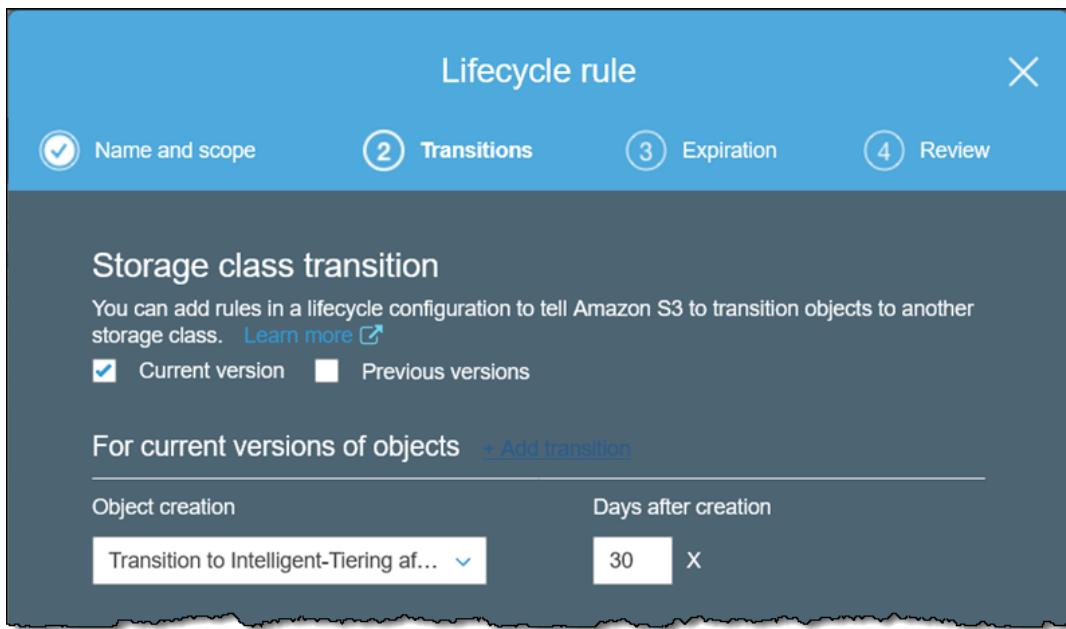
As a Solutions Architect, what should you do if the company needs to automatically transfer or archive media data from an S3 bucket to Glacier?

- Use Amazon SQS
- Use Amazon SWF
- Use a custom shell script that transfers data from the S3 bucket to Glacier
- **Use Lifecycle Policies**

Correct

You can create a lifecycle policy in S3 to automatically transfer your data to Glacier.

Lifecycle configuration enables you to specify the lifecycle management of objects in a bucket. The configuration is a set of one or more rules, where each rule defines an action for Amazon S3 to apply to a group of objects.



These actions can be classified as follows:

Transition actions – In which you define when objects transition to another storage class. For example, you may choose to transition objects to the STANDARD_IA (IA, for infrequent access) storage class 30 days after creation or archive objects to the GLACIER storage class one year after creation.

Expiration actions – In which you specify when the objects expire. Then Amazon S3 deletes the expired objects on your behalf.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

4. QUESTION

Category: CSAA – Design Resilient Architectures

A Solutions Architect is trying to enable Cross-Region Replication to an S3 bucket but this option is disabled. Which of the following options is a valid reason for this?

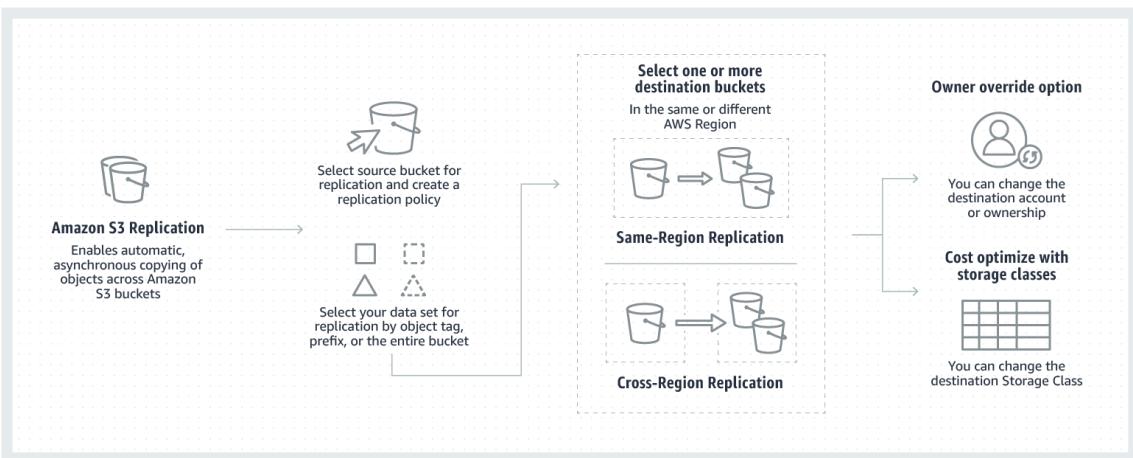
- In order to use the Cross-Region Replication feature in S3, you need to first enable versioning on the bucket.
- This is a premium feature which is only for AWS Enterprise accounts.
- The Cross-Region Replication feature is only available for Amazon S3 – One Zone-IA
- The Cross-Region Replication feature is only available for Amazon S3 – Infrequent Access.

Correct

To enable the cross-region replication feature in S3, the following items should be met:

1. The source and destination buckets must have **versioning** enabled.
2. The source and destination buckets must be in different AWS Regions.

3. Amazon S3 must have permissions to replicate objects from that source bucket to the destination bucket on your behalf.



The options that say: ***The Cross-Region Replication feature is only available for Amazon S3 - One Zone-IA*** and ***The Cross-Region Replication feature is only available for Amazon S3 - Infrequent Access*** are incorrect as this feature is available to all types of S3 classes.

The option that says: ***This is a premium feature which is only for AWS Enterprise accounts*** is incorrect as this CRR feature is available to all Support Plans.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/crr.html>

<https://aws.amazon.com/blogs/aws/new-cross-region-replication-for-amazon-s3/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

5. QUESTION

Category: CSAA – Design Resilient Architectures

A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture that is composed of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used by an online accounting application that requires path-based routing, host-based routing, and bi-directional communication channels using WebSockets.

Which is the most suitable type of Elastic Load Balancer that will satisfy the given requirement?

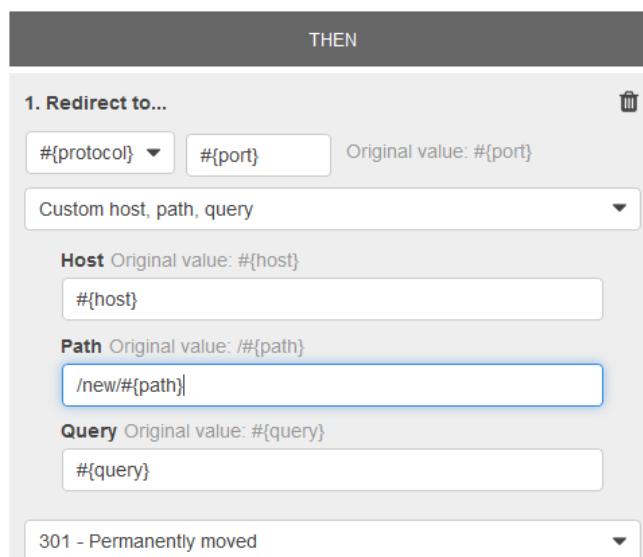
- Classic Load Balancer
- Network Load Balancer
- **Application Load Balancer**
- Either a Classic Load Balancer or a Network Load Balancer

Correct

Elastic Load Balancing supports three types of load balancers. You can select the appropriate load balancer based on your application needs.

If you need flexible application management and TLS termination then it is recommended to use Application Load Balancer. If extreme performance and static IP is needed for your application then it is recommend that you use Network Load Balancer. If your application is built within the EC2 Classic network then you should use Classic Load Balancer.

An **Application Load Balancer** functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.



Application Load Balancers support path-based routing, host-based routing, and support for containerized applications hence, ***Application Load Balancer*** is the correct answer.

Network Load Balancer, ***Classic Load Balancer***, and ***either a Classic Load Balancer or a Network Load Balancer*** are all incorrect as none of these support path-based routing and host-based routing, unlike an Application Load Balancer.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits>

<https://aws.amazon.com/elasticloadbalancing/faqs/>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

<https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/>

6. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company is using AWS IAM to manage access to AWS services. The Solutions Architect of the company created the following IAM policy for AWS Lambda:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "lambda>CreateFunction",  
                "lambda>DeleteFunction"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "lambda>CreateFunction",  
                "lambda>DeleteFunction",  
                "lambda>InvokeFunction",  
                "lambda>TagResource"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringLike": {"aws:RequestID": "aws:SecureTransport"}  
            }  
        }  
    ]  
}
```

```

    "IpAddress": {
        "aws:SourceIp": "187.5.104.11/32"
    }
}
]
}

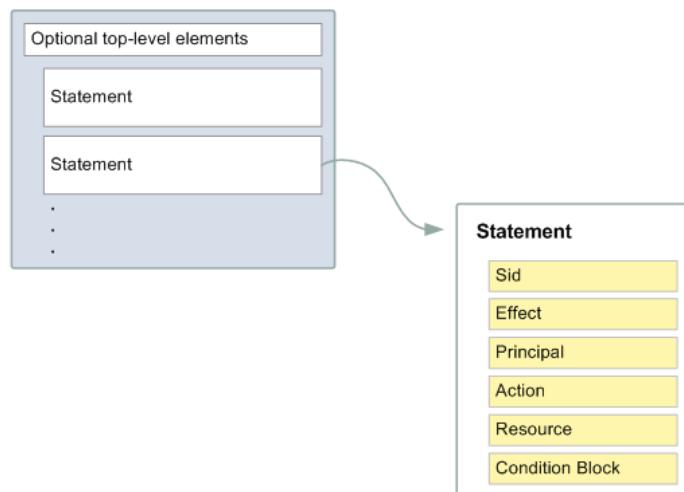
```

Which of the following options are allowed by this policy?

- Delete an AWS Lambda function using the `187.5.104.11/32` address.
- **Create an AWS Lambda function using the `100.220.0.11/32` address.**
- Delete an AWS Lambda function from any network address.
- Create an AWS Lambda function using the `187.5.104.11/32` address.

Correct

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.



You can use AWS Identity and Access Management (IAM) to manage access to the Lambda API and resources like functions and layers. Based on the given IAM policy, you can create and delete a Lambda function from any network address except for the IP address `187.5.104.11/32`. Since the IP address, `100.220.0.11/32` is not denied in the policy, you can use this address to create a Lambda function.

Hence, the correct answer is: ***Create an AWS Lambda function using the `100.220.0.11/32` address***.

The option that says: ***Delete an AWS Lambda function using the `187.5.104.11/32` address*** is incorrect because the source IP used in this option is denied by the IAM policy.

The option that says: ***Delete an AWS Lambda function from any network address*** is incorrect. You can't delete a Lambda function from any network address because the address `187.5.104.11/32` is denied by the policy.

The option that says: ***Create an AWS Lambda function using the `187.5.104.11/32` address*** is incorrect. Just like the option above, the IAM policy denied the IP address `187.5.104.11/32`.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-permissions.html>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

To save costs, your manager instructed you to analyze and review the setup of your AWS cloud infrastructure. You should also provide an estimate of how much your company will pay for all of the AWS resources that they are using.

In this scenario, which of the following will incur costs? (Select TWO.)

- Using an Amazon VPC
- Public Data Set
- A stopped On-Demand EC2 Instance
- **EBS Volumes attached to stopped EC2 Instances**
- **A running EC2 Instance**

Incorrect

Billing commences when Amazon EC2 initiates the boot sequence of an AMI instance. Billing ends when the instance terminates, which could occur through a web services command, by running “shutdown -h”, or through instance failure. When you stop an instance, AWS shuts it down but doesn’t charge hourly usage for a stopped instance or data transfer fees. However, AWS does charge for the storage of any Amazon EBS volumes.

Hence, ***a running EC2 Instance*** and ***EBS Volumes attached to stopped EC2 Instances*** are the right answers and conversely, ***a stopped On-Demand EC2 Instance*** is incorrect as there is no charge for a stopped EC2 instance that you have shut down.

Using Amazon VPC is incorrect because there are no additional charges for creating and using the VPC itself. Usage charges for other Amazon Web Services, including Amazon EC2, still apply at published rates for those resources, including data transfer charges.

Public Data Set is incorrect due to the fact that Amazon stores the data sets at no charge to the community and, as with all AWS services, you pay only for the compute and storage you use for your own applications.

References:

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

<https://aws.amazon.com/vpc/faqs>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-public-data-sets.html>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

8. QUESTION

Category: CSAA – Design High-Performing Architectures

A company plans to migrate a MySQL database from an on-premises data center to the AWS Cloud. This database will be used by a legacy batch application that has steady-state workloads in the morning but has its peak load at night for the end-of-day processing. You need to choose an EBS volume that can handle a maximum of 450 GB of data and can also be used as the system boot volume for your EC2 instance.

Which of the following is the most cost-effective storage type to use in this scenario?

- Amazon EBS Cold HDD (sc1)
- Amazon EBS General Purpose SSD (gp2)
- Amazon EBS Throughput Optimized HDD (st1)
- Amazon EBS Provisioned IOPS SSD (io1)

Incorrect

In this scenario, a legacy batch application which has steady-state workloads requires a ***relational MySQL database***. The EBS volume that you should use has to handle a maximum of 450 GB of data and can also be used as the system ***boot volume*** for your EC2 instance. **Since HDD volumes cannot be used as a bootable volume**, we can narrow down our options by selecting SSD volumes. In addition, SSD volumes are more suitable for transactional database workloads, as shown in the table below:

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

General Purpose SSD (`gp2`) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. AWS designs `gp2` volumes to deliver the provisioned performance 99% of the time. A `gp2` volume can range in size from 1 GiB to 16 TiB.

Amazon EBS Provisioned IOPS SSD (io1) is incorrect because this is not the most cost-effective EBS type and is primarily used for critical business applications that require sustained IOPS performance.

Amazon EBS Throughput Optimized HDD (st1) is incorrect because this is primarily used for frequently accessed, throughput-intensive workloads. Although it is a low-cost HDD volume, it cannot be used as a system boot volume.

Amazon EBS Cold HDD (sc1) is incorrect. Although Amazon EBS Cold HDD provides lower cost HDD volume compared to General Purpose SSD, it cannot be used as a system boot volume.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#EBSVolumeTypes_gp2

Amazon EBS Overview – SSD vs HDD:

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

9. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect is designing a monitoring application which generates audit logs of all operational activities of the company's cloud infrastructure. Their IT Security and Compliance team mandates that the application retain the logs for 5 years before the data can be deleted.

How can the Architect meet the above requirement?

- Store the audit logs in an EFS volume and use Network File System version 4 (NFSv4) file-locking mechanism.
- Store the audit logs in an EBS volume and then take EBS snapshots every month.
- Store the audit logs in an Amazon S3 bucket and enable Multi-Factor Authentication Delete (MFA Delete) on the S3 bucket.
- **Store the audit logs in a Glacier vault and use the Vault Lock feature.**

Correct

An **Amazon S3 Glacier (Glacier) vault** can have one resource-based vault access policy and one Vault Lock policy attached to it. A *Vault Lock policy* is a vault access policy that you can lock. Using a Vault Lock policy can help you enforce regulatory and compliance requirements. Amazon S3 Glacier provides a set of API operations for you to manage the Vault Lock policies.

Vault Lock policy for BusinessCritical

The Vault Lock policy for the vault is shown below. [Click here](#) to learn about writing a Vault Lock policy.

```
Add a permission
```

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Principal": {"AWS": "*"},  
            "Action": "glacier:DeleteArchive",  
            "Resource": "arn:aws:glacier:us-east-1:  
                        :vaults/BusinessCritical",  
            "Condition": {  
                "NumericLessThanEquals": {  
                    "glacier:ArchiveAgeInDays": "365"  
                }  
            }  
        }  
    ]  
}
```

[Cancel](#) [Initiate Vault Lock](#)

As an example of a Vault Lock policy, suppose that you are required to retain archives for one year before you can delete them. To implement this requirement, you can create a Vault Lock policy that denies users permissions to delete an archive until the archive has existed for one year. You can test this policy before locking it down. After you lock the policy, the policy becomes immutable. For more information about the locking process, see Amazon S3 Glacier Vault Lock. If you want to manage other user permissions that can be changed, you can use the vault access policy

Amazon S3 Glacier supports the following archive operations: Upload, Download, and Delete. Archives are immutable and **cannot be modified**. Hence, the correct answer is to ***store the audit logs in a Glacier vault and use the Vault Lock feature***.

Storing the audit logs in an EBS volume and then taking EBS snapshots every month is incorrect because this is not a suitable and secure solution. Anyone who has access to the EBS Volume can simply delete and modify the audit logs. Snapshots can be deleted too.

Storing the audit logs in an Amazon S3 bucket and enabling Multi-Factor Authentication Delete (MFA Delete) on the S3 bucket is incorrect because this would still not meet the requirement. If someone has access to the S3 bucket and also has the proper MFA privileges then the audit logs can be edited.

Storing the audit logs in an EFS volume and using Network File System version 4 (NFSv4) file-locking mechanism is incorrect because the data integrity of the audit logs can still be compromised if it is stored in an EFS volume with Network File System version 4 (NFSv4) file-locking mechanism and hence, not suitable as storage for the files. Although it will provide some sort of security, the file lock can still be overridden and the audit logs might be edited by someone else.

References:

<https://docs.aws.amazon.com/amazonglacier/latest/dev/vault-lock.html>

<https://docs.aws.amazon.com/amazonglacier/latest/dev/vault-lock-policy.html>

<https://aws.amazon.com/blogs/aws/glacier-vault-lock/>

Amazon S3 and S3 Glacier Overview:

Check out this Amazon S3 Glacier Cheat Sheet:

<https://tutorialsdojo.com/amazon-glacier/>

10. QUESTION

Category:CSAA – Design High-Performing Architectures

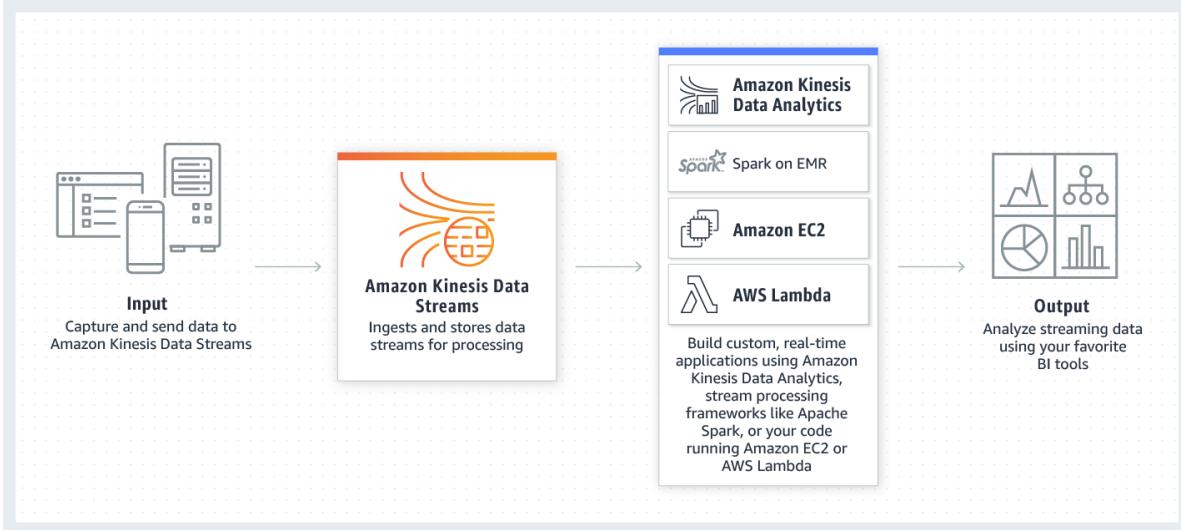
A company plans to develop a custom messaging service that will also be used to train their AI for an automatic response feature which they plan to implement in the future. Based on their research and tests, the service can receive up to thousands of messages a day, and all of these data are to be sent to Amazon EMR for further processing. It is crucial that none of the messages are lost, no duplicates are produced, and that they are processed in EMR in the same order as their arrival.

Which of the following options can satisfy the given requirement?

- Create an Amazon Kinesis Data Stream to collect the messages.
- Set up a default Amazon SQS queue to handle the messages.
- Set up an Amazon SNS Topic to handle the messages.
- Create a pipeline using AWS Data Pipeline to handle the messages.

Incorrect

Two important requirements that the chosen AWS service should fulfill is that data should not go missing, is durable, and streams data in the sequence of arrival. Kinesis can do the job just fine because of its architecture. A **Kinesis data stream** is a set of shards that has a sequence of data records, and each data record has a sequence number that is assigned by Kinesis Data Streams. Kinesis can also easily handle the high volume of messages being sent to the service.



Amazon Kinesis Data Streams enables real-time processing of streaming big data. It provides ordering of records, as well as the ability to read and/or replay records in the same order to multiple Amazon Kinesis Applications. The Amazon Kinesis Client Library (KCL) delivers all records for a given partition key to the same record processor, making it easier to build multiple applications reading from the same Amazon Kinesis data stream (for example, to perform counting, aggregation, and filtering).

Setting up a default Amazon SQS queue to handle the messages is incorrect because although SQS is a valid messaging service, it is not suitable for scenarios where you need to process the data based on the order they were received. Take note that a default queue in SQS is just a standard queue and not a FIFO (First-In-First-Out) queue. In addition, SQS does not guarantee that no duplicates will be sent.

Setting up an Amazon SNS Topic to handle the messages is incorrect because SNS is a pub-sub messaging service in AWS. SNS might not be capable of handling such a large volume of messages being received and sent at a time. It does not also guarantee that the data will be transmitted in the same order they were received.

Creating a pipeline using AWS Data Pipeline to handle the messages is incorrect because this is primarily used as a cloud-based data workflow service that helps you process and move data between different AWS services and on-premises data sources. It is not suitable for collecting data from distributed sources such as users, IoT devices, or clickstreams.

References:

<https://docs.aws.amazon.com/streams/latest/dev/introduction.html>

For additional information, read the **When should I use Amazon Kinesis Data Streams, and when should I use Amazon SQS?** section of the Kinesis Data Stream FAQ:

<https://aws.amazon.com/kinesis/data-streams/faqs/>

Check out this Amazon Kinesis Cheat Sheet:

<https://tutorialsdojo.com/amazon-kinesis/>

11. QUESTION

Category: CSAA – Design High-Performing Architectures

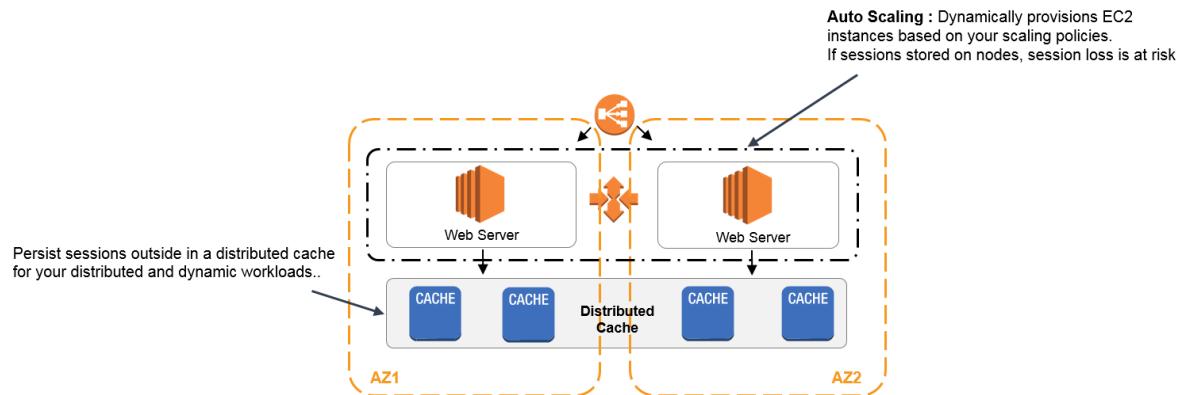
A company has a fleet of running Spot EC2 instances behind an Application Load Balancer. The incoming traffic comes from various users across multiple AWS regions and you would like to have the user's session shared among the fleet of instances. You are required to set up a distributed session management layer that will provide a scalable and shared data storage for the user sessions.

Which of the following would be the best choice to meet the requirement while still providing sub-millisecond latency for the users?

- Multi-AZ RDS
- Multi-master DynamoDB
- **ElastiCache in-memory caching**
- ELB sticky sessions

Incorrect

For sub-millisecond latency caching, **ElastiCache** is the best choice. In order to address scalability and to provide a shared data storage for sessions that can be accessed from any individual web server, you can abstract the HTTP sessions from the web servers themselves. A common solution for this is to leverage an In-Memory Key/Value store such as Redis and Memcached.



ELB sticky sessions is incorrect because the scenario does not require you to route a user to the particular web server that is managing that individual user's session. Since the session state is shared among the instances, the use of the ELB sticky sessions feature is not recommended in this scenario.

Multi-master DynamoDB and ***Multi-AZ RDS*** are incorrect. Although you can use DynamoDB and RDS for storing session state, these two are not the best choices in terms of cost-effectiveness and performance when compared to ElastiCache. There is a significant difference in terms of latency if you used DynamoDB and RDS when you store the session data.

References:

<https://aws.amazon.com/caching/session-management/>

<https://d0.awsstatic.com/whitepapers/performance-at-scale-with-amazon-elasticache.pdf>

Check out this Amazon ElastiCache Cheat Sheet:

<https://tutorialsdojo.com/amazon-elasticache/>

Redis (cluster mode enabled vs disabled) vs Memcached:

<https://tutorialsdojo.com/redis-cluster-mode-enabled-vs-disabled-vs-memcached/>

12. QUESTION

Category: CSAA – Design Resilient Architectures

A company deployed an online enrollment system database on a prestigious university, which is hosted in RDS. The Solutions Architect is required to monitor the database metrics in Amazon CloudWatch to ensure the availability of the enrollment system.

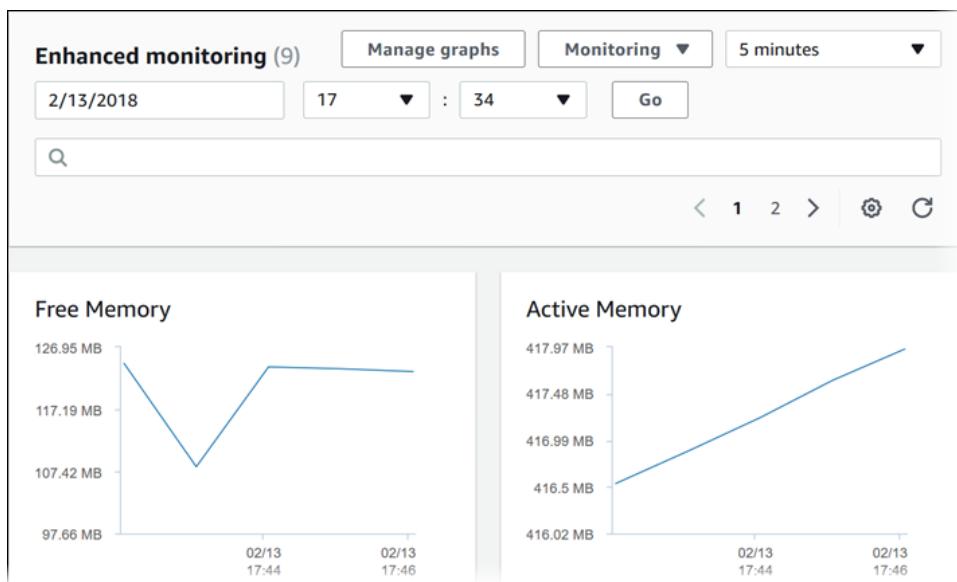
What are the enhanced monitoring metrics that Amazon CloudWatch gathers from Amazon RDS DB instances which provide more accurate information? (Select TWO.)

- CPU Utilization
- Database Connections
- OS processes
- Freeable Memory
- RDS child processes

Incorrect

Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from CloudWatch Logs in a monitoring system of your choice.

CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and Enhanced Monitoring gathers its metrics from an agent on the instance. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work. The differences can be greater if your DB instances use smaller instance classes, because then there are likely more virtual machines (VMs) that are managed by the hypervisor layer on a single physical instance. Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU.



In RDS, the Enhanced Monitoring metrics shown in the Process List view are organized as follows:

RDS child processes – Shows a summary of the RDS processes that support the DB instance, for example `aurora` for Amazon Aurora DB clusters and `mysql` for MySQL DB instances. Process threads appear nested beneath the parent process. Process threads show CPU utilization only as other metrics are the same for all threads for the process. The console displays a maximum of 100 processes and threads. The results are a combination of the top CPU consuming and memory consuming processes and threads. If there are more than 50 processes and more than 50 threads, the console displays the top 50 consumers in each category. This display helps you identify which processes are having the greatest impact on performance.

RDS processes – Shows a summary of the resources used by the RDS management agent, diagnostics monitoring processes, and other AWS processes that are required to support RDS DB instances.

OS processes – Shows a summary of the kernel and system processes, which generally have minimal impact on performance.

CPU Utilization, Database Connections, and ***Freeable Memory*** are incorrect because these are just the regular items provided by Amazon RDS Metrics in CloudWatch. Remember that the scenario is asking for the Enhanced Monitoring metrics.

References:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/rds-metricscollected.html>

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Monitoring.OS.html#USER_Monitoring.OS.CloudWatchLogs

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

13. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A travel company has a suite of web applications hosted in an Auto Scaling group of On-Demand EC2 instances behind an Application Load Balancer that handles traffic from various web domains such as `i-love-manila.com`, `i-love-boracay.com`, `i-love-cebu.com` and many others. To improve security and lessen the overall cost, you are instructed to secure the system by allowing multiple domains to serve SSL traffic without the need to reauthenticate and reprovision your certificate everytime you add a new domain. This migration from HTTP to HTTPS will help improve their SEO and Google search ranking.

Which of the following is the most cost-effective solution to meet the above requirement?

- Upload all SSL certificates of the domains in the ALB using the console and bind multiple certificates to the same secure listener on your load balancer. ALB will automatically choose the optimal TLS certificate for each client using Server Name Indication (SNI).
- Add a Subject Alternative Name (SAN) for each additional domain to your certificate.
- Use a wildcard certificate to handle multiple sub-domains and different domains.
- Create a new CloudFront web distribution and configure it to serve HTTPS requests using dedicated IP addresses in order to associate your alternate domain names with a dedicated IP address in each CloudFront edge location.

Correct

SNI Custom SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname which the viewers are trying to connect to.

You can host multiple TLS secured applications, each with its own TLS certificate, behind a single load balancer. In order to use SNI, all you need to do is bind multiple certificates to the same secure listener on your load balancer. ALB will automatically choose the optimal TLS certificate for each client. These features are provided at no additional charge.

The screenshot shows the AWS EC2 Dashboard with the 'Load Balancers' section selected. A table lists a single load balancer named 'MyFancyALB' with a DNS name of 'MyFancyALB-347622664.us...' and an active state. Below the table, the 'Listeners' tab is selected for 'MyFancyALB', showing a single listener for HTTPS port 443 using the ELBSecurityPolicy-2016-08 security policy and a default SSL certificate.

To meet the requirements in the scenario, you can upload all SSL certificates of the domains in the ALB using the console and bind multiple certificates to the same secure listener on your load balancer. ALB will automatically choose the optimal TLS certificate for each client using Server Name Indication (SNI).

Hence, the correct answer is the option that says: ***Upload all SSL certificates of the domains in the ALB using the console and bind multiple certificates to the same secure listener on your load balancer. ALB will automatically choose the optimal TLS certificate for each client using Server Name Indication (SNI).***

Using a wildcard certificate to handle multiple sub-domains and different domains is incorrect because a wildcard certificate can only handle multiple sub-domains but not different domains.

Adding a Subject Alternative Name (SAN) for each additional domain to your certificate is incorrect because although using SAN is correct, you will still have to reauthenticate and reprovision your certificate every time you add a new domain. One of the requirements in the scenario is that you should not have to reauthenticate and reprovision your certificate hence, this solution is incorrect.

The option that says: ***Create a new CloudFront web distribution and configure it to serve HTTPS requests using dedicated IP addresses in order to associate your alternate domain names with a dedicated IP address in each CloudFront edge location*** is incorrect because although it is valid to use dedicated IP addresses to meet this requirement, this solution is not cost-effective. Remember that if you configure CloudFront to serve HTTPS requests using dedicated IP addresses, you incur an additional monthly charge. The charge begins when you associate your SSL/TLS certificate with your CloudFront distribution. You can just simply upload the certificates to the ALB and use SNI to handle multiple domains in a cost-effective manner.

References:

<https://aws.amazon.com/blogs/aws/new-application-load-balancer-sni/>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/cnames-https-dedicated-ip-or-sni.html#cnames-https-dedicated-ip>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

SNI Custom SSL vs Dedicated IP Custom SSL:

<https://tutorialsdojo.com/sni-custom-ssl-vs-dedicated-ip-custom-ssl/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

14. QUESTION

Category: CSAA – Design Secure Applications and Architectures

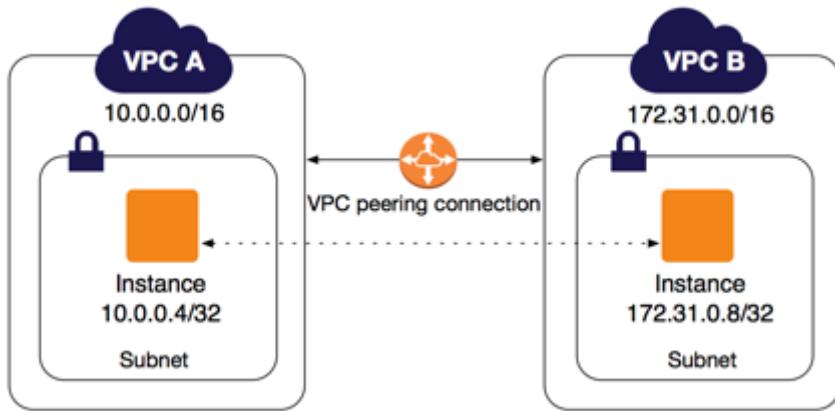
An operations team has an application running on EC2 instances inside two custom VPCs. The VPCs are located in the Ohio and N.Virginia Region respectively. The team wants to transfer data between the instances without traversing the public internet.

Which combination of steps will achieve this? (Select TWO.)

- Set up a VPC peering connection between the VPCs.
- Launch a NAT Gateway in the public subnet of each VPC.
- Create an Egress-only Internet Gateway.
- Re-configure the route table's target and destination of the instances' subnet.
- Deploy a VPC endpoint on each region to enable a private connection.

Incorrect

A **VPC peering connection** is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. The VPCs can be in different regions (also known as an inter-region VPC peering connection).



Inter-Region VPC Peering provides a simple and cost-effective way to share resources between regions or replicate data for geographic redundancy. Built on the same horizontally scaled, redundant, and highly available technology that powers VPC today, Inter-Region VPC Peering encrypts inter-region traffic with no single point of failure or bandwidth bottleneck. Traffic using Inter-Region VPC Peering always stays on the global AWS backbone and never traverses the public internet, thereby reducing threat vectors, such as common exploits and DDoS attacks.

Hence, the correct answers are:

– Set up a VPC peering connection between the VPCs.

– Re-configure the route table's target and destination of the instances' subnet.

The option that says: ***Create an Egress only Internet Gateway*** is incorrect because this will just enable outbound IPv6 communication from instances in a VPC to the internet. Take note that the scenario requires private communication to be enabled between VPCs from two different regions.

The option that says: ***Launch a NAT Gateway in the public subnet of each VPC*** is incorrect because NAT Gateways are used to allow instances in private subnets to access the public internet. Note that the requirement is to make sure that communication between instances will not traverse the internet.

The option that says: ***Deploy a VPC endpoint on each region to enable private connection*** is incorrect. VPC endpoints are region-specific only and do not support inter-region communication.

References:

<https://docs.aws.amazon.com/vpc/latest/peering/what-is-vpc-peering.html>

<https://aws.amazon.com/about-aws/whats-new/2017/11/announcing-support-for-inter-region-vpc-peering/>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

15. QUESTION

Category: CSAA – Design Secure Applications and Architectures

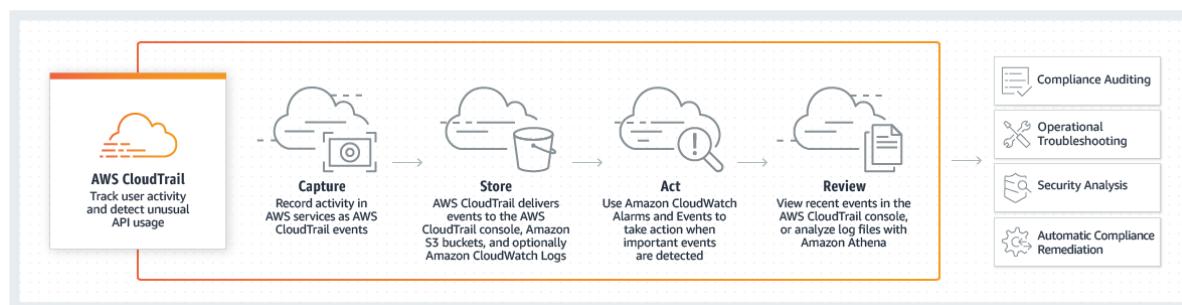
A startup has resources deployed on the AWS Cloud. It is now going through a set of scheduled audits by an external auditing firm for compliance.

Which of the following services available in AWS can be utilized to help ensure the right information are present for auditing purposes?

- **AWS CloudTrail**
- Amazon VPC
- Amazon EC2
- Amazon CloudWatch

Incorrect

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting.



CloudTrail provides visibility into user activity by recording actions taken on your account. CloudTrail records important information about each action, including who made the request, the services used, the actions performed, parameters for the actions, and the response elements returned by the AWS service. This information helps you to track changes made to your AWS resources and troubleshoot operational issues. CloudTrail makes it easier to ensure compliance with internal policies and regulatory standards.

Hence, the correct answer is: ***AWS CloudTrail.***

Amazon VPC is incorrect because a VPC is a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. It does not provide you the auditing information that were asked for in this scenario.

Amazon EC2 is incorrect because EC2 is a service that provides secure, resizable compute capacity in the cloud and does not provide the needed information in this scenario just like the option above.

Amazon CloudWatch is incorrect because this is a monitoring tool for your AWS resources. Like the above options, it does not provide the needed information to satisfy the requirement in the scenario.

Reference:

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

Check out this AWS CloudTrail Cheat Sheet:

<https://tutorialsdojo.com/aws-cloudtrail/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A commercial bank has designed its next-generation online banking platform to use a distributed system architecture. As their Software Architect, you have to ensure that their architecture is highly scalable, yet still cost-effective.

Which of the following will provide the most suitable solution for this scenario?

- Launch multiple On-Demand EC2 instances to host your application services and an SQS queue which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.
- Launch multiple EC2 instances behind an Application Load Balancer to host your application services, and SWF which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.
- Launch an Auto-Scaling group of EC2 instances to host your application services and an SQS queue. Include an Auto Scaling trigger to watch the SQS queue size which will either scale in or scale out the number of EC2 instances based on the queue.
- Launch multiple EC2 instances behind an Application Load Balancer to host your application services and SNS which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.

Correct

There are three main parts in a distributed messaging system: the components of your distributed system which can be hosted on EC2 instance; your queue (distributed on Amazon SQS servers); and the messages in the queue.

To improve the scalability of your distributed system, you can add Auto Scaling group to your EC2 instances.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-using-sqs-queue.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

An application is hosted in an Auto Scaling group of EC2 instances. To improve the monitoring process, you have to configure the current capacity to increase or decrease based on a set of scaling adjustments. This should be done by specifying the scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process.

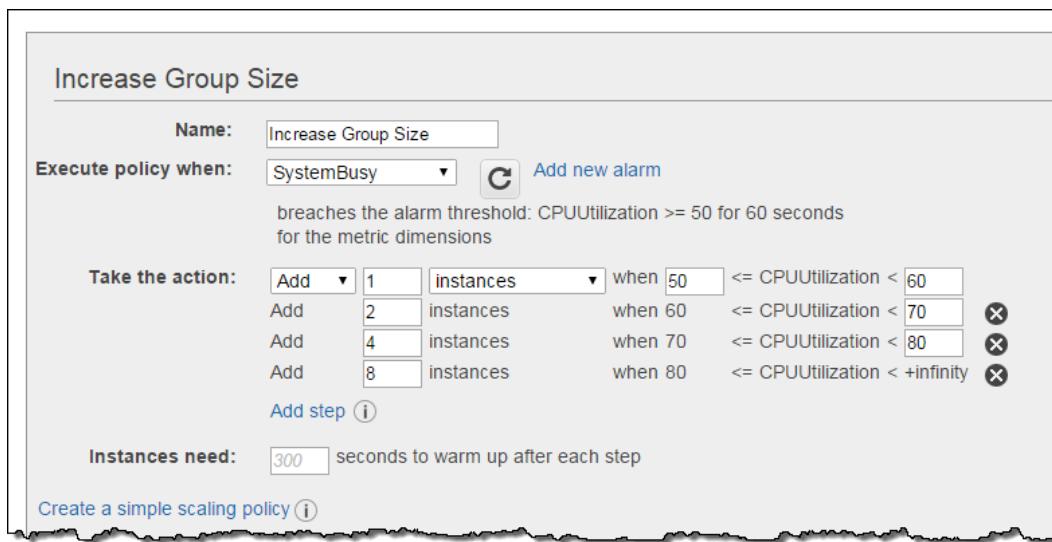
Which of the following is the most suitable type of scaling policy that you should use?

- Target tracking scaling
- **Step scaling**
- Simple scaling
- Scheduled Scaling

Correct

With step scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process as well as define how your scalable target should be scaled when a threshold is breached for a specified number of evaluation periods. Step scaling policies increase or decrease the current capacity of a scalable target based on a set of scaling adjustments, known as step adjustments. The adjustments vary based on the size of the alarm breach. After a scaling activity is started, the policy continues to respond to additional alarms, even while a scaling activity is in progress. Therefore, all alarms that are breached are evaluated by Application Auto Scaling as it receives the alarm messages.

When you configure dynamic scaling, you must define how to scale in response to changing demand. For example, you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay at around 50 percent when the load on the application changes. This gives you extra capacity to handle traffic spikes without maintaining an excessive amount of idle resources. You can configure your Auto Scaling group to scale automatically to meet this need. The policy type determines how the scaling action is performed.



Amazon EC2 Auto Scaling supports the following types of scaling policies:

Target tracking scaling – Increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home – you select a temperature and the thermostat does the rest.

Step scaling – Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as *step adjustments*, that vary based on the size of the alarm breach.

Simple scaling – Increase or decrease the current capacity of the group based on a single scaling adjustment.

If you are scaling based on a utilization metric that increases or decreases proportionally to the number of instances in an Auto Scaling group, then it is recommended that you use target tracking scaling policies. Otherwise, it is better to use step scaling policies instead.

Hence, the correct answer in this scenario is ***Step Scaling***.

Target tracking scaling is incorrect because the target tracking scaling policy increases or decreases the current capacity of the group based on a **target value for a specific metric**, instead of a set of scaling adjustments.

Simple scaling is incorrect because the simple scaling policy increases or decreases the current capacity of the group based on a **single** scaling adjustment, instead of a set of scaling adjustments.

Scheduled Scaling is incorrect because the scheduled scaling policy is based on a schedule that allows you to set your own scaling schedule for **predictable** load changes. This is not considered as one of the types of dynamic scaling.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scale-based-on-demand.html>

<https://docs.aws.amazon.com/autoscaling/application/userguide/application-auto-scaling-step-scaling-policies.html>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

An online shopping platform is hosted on an Auto Scaling group of On-Demand EC2 instances with a default Auto Scaling termination policy and no instance protection configured. The system is deployed across three Availability Zones in the US West region (us-west-1) with an Application Load Balancer in front to provide high availability and fault tolerance for the shopping platform. The us-west-1a, us-west-1b, and us-west-1c Availability Zones have 10, 8 and 7 running instances respectively. Due to the low number of incoming traffic, the scale-in operation has been triggered.

Which of the following will the Auto Scaling group do to determine which instance to terminate first in this scenario? (Select THREE.)

- Choose the Availability Zone with the most number of instances, which is the us-west-1a Availability Zone in this scenario.
- Select the instances with the oldest launch configuration.
- Select the instance that is farthest to the next billing hour.
- Choose the Availability Zone with the least number of instances, which is the us-west-1c Availability Zone in this scenario.
- Select the instance that is closest to the next billing hour.
- Select the instances with the most recent launch configuration.

Correct

The default termination policy is designed to help ensure that your network architecture spans Availability Zones evenly. With the default termination policy, the behavior of the Auto Scaling group is as follows:

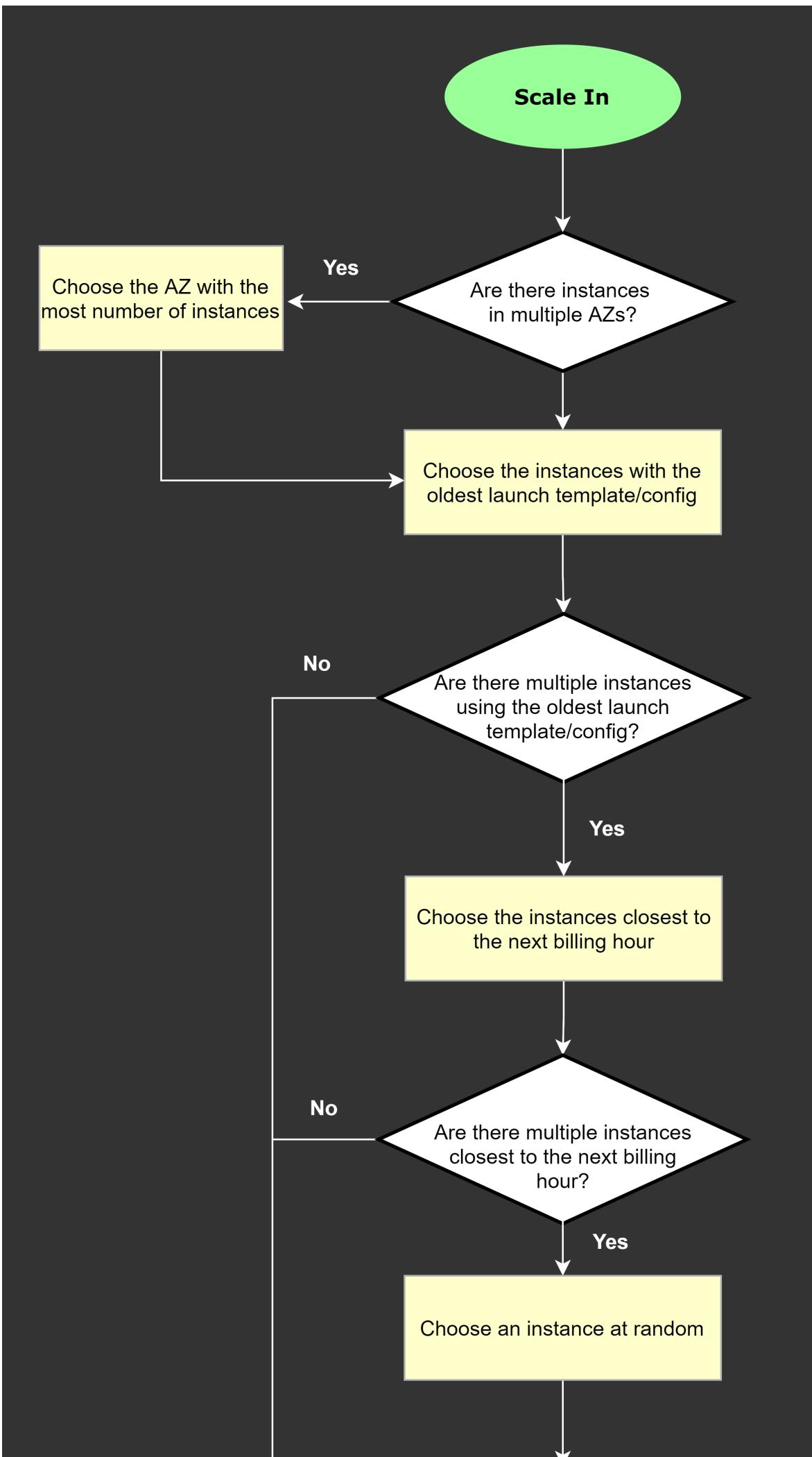
\1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that use the oldest launch configuration.

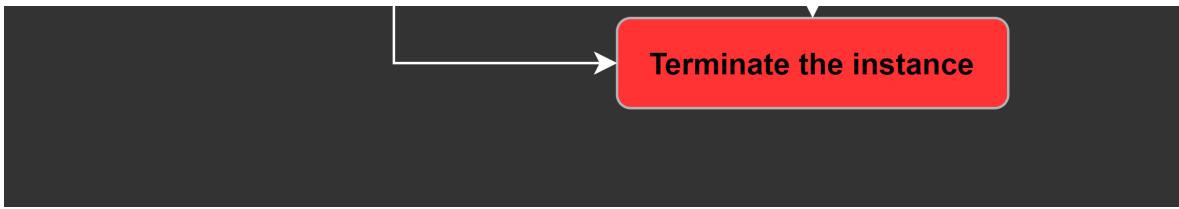
\2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.

\3. If there are multiple instances to terminate based on the above criteria, determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there is one such instance, terminate it.

\4. If there is more than one unprotected instance closest to the next billing hour, choose one of these instances at random.

The following flow diagram illustrates how the default termination policy works:





Terminate the instance

Reference:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html#default-termination-policy>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

4. QUESTION

Category: CSAA – Design Resilient Architectures

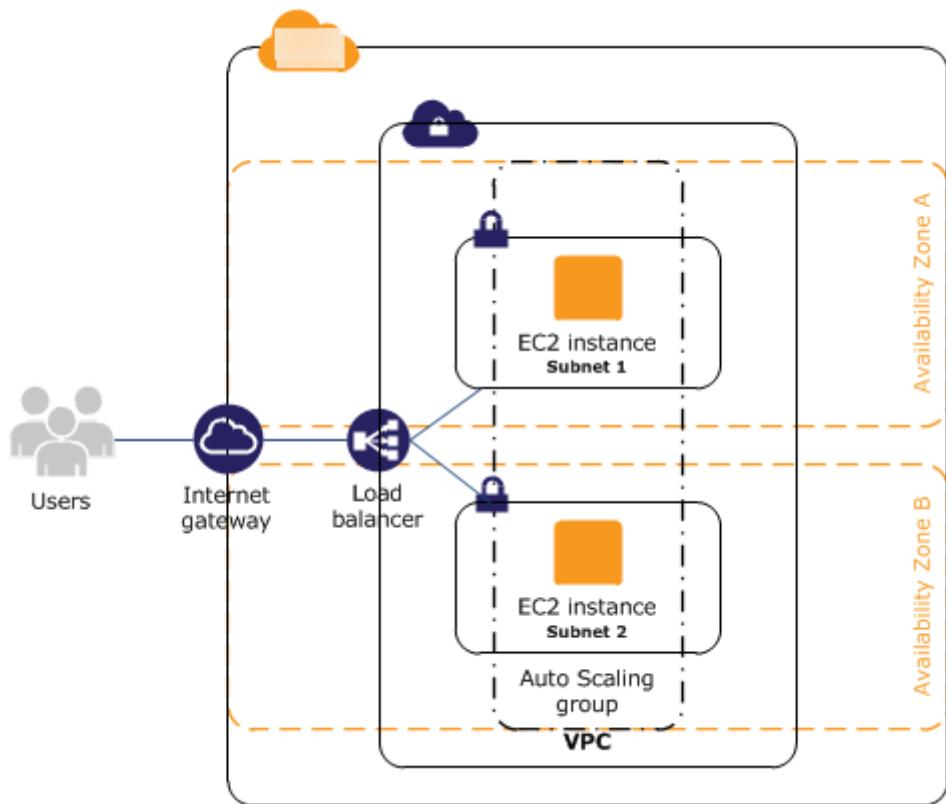
A major TV network has a web application running on eight Amazon T3 EC2 instances. The number of requests that the application processes are consistent and do not experience spikes. To ensure that eight instances are running at all times, the Solutions Architect should create an Auto Scaling group and distribute the load evenly between all instances.

Which of the following options can satisfy the given requirements?

- Deploy two EC2 instances with Auto Scaling in four regions behind an Amazon Elastic Load Balancer.
- Deploy four EC2 instances with Auto Scaling in one Availability Zone and four in another availability zone in the same region behind an Amazon Elastic Load Balancer.
- Deploy four EC2 instances with Auto Scaling in one region and four in another region behind an Amazon Elastic Load Balancer.
- Deploy eight EC2 instances with Auto Scaling in one Availability Zone behind an Amazon Elastic Load Balancer.

Incorrect

The best option to take is to deploy four EC2 instances in one Availability Zone and four in another availability zone in the same region behind an Amazon Elastic Load Balancer. In this way, if one availability zone goes down, there is still another available zone that can accommodate traffic.



When the first AZ goes down, the second AZ will only have an initial 4 EC2 instances. This will eventually be scaled up to 8 instances since the solution is using Auto Scaling.

The 110% compute capacity for the 4 servers might cause some degradation of the service, but not a total outage since there are still some instances that handle the requests.

Depending on your scale-up configuration in your Auto Scaling group, the additional 4 EC2 instances can be launched in a matter of minutes.

T3 instances also have a Burstable Performance capability to burst or go beyond the current compute capacity of the instance to higher performance as required by your workload. So your 4 servers will be able to manage 110% compute capacity for a short period of time. This is the power of cloud computing versus our on-premises network architecture. It provides elasticity and unparalleled scalability.

Take note that **Auto Scaling will launch additional EC2 instances to the remaining Availability Zone/s in the event of an Availability Zone outage in the region**. Hence, the correct answer is the option that says: ***Deploy four EC2 instances with Auto Scaling in one Availability Zone and four in another availability zone in the same region behind an Amazon Elastic Load Balancer.***

The option that says: ***Deploy eight EC2 instances with Auto Scaling in one Availability Zone behind an Amazon Elastic Load Balancer*** is incorrect because this architecture is not highly available. If that Availability Zone goes down then your web application will be unreachable.

The options that say: ***Deploy four EC2 instances with Auto Scaling in one region and four in another region behind an Amazon Elastic Load Balancer*** and ***Deploy two EC2 instances with Auto Scaling in four regions behind an Amazon Elastic Load Balancer*** are incorrect because the **ELB is designed to only run in one region and not across multiple regions.**

References:

<https://aws.amazon.com/elasticloadbalancing/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-increase-availability.html>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

5. QUESTION

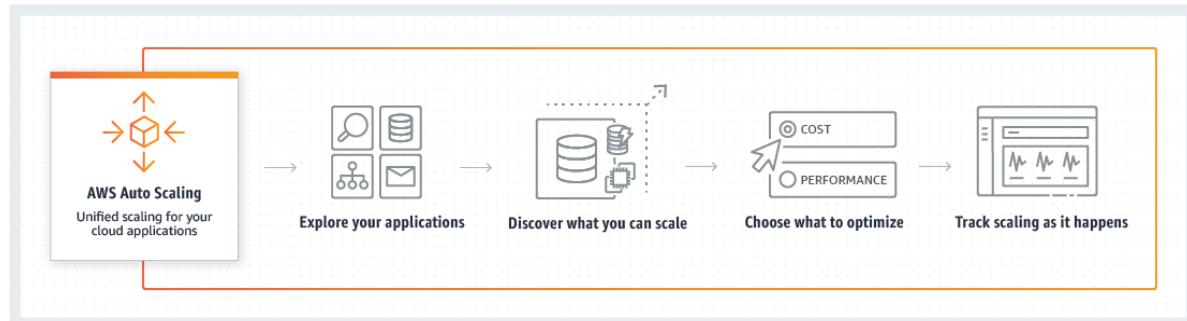
Category: CSAA – Design Resilient Architectures

A tech company is currently using Auto Scaling for their web application. A new AMI now needs to be used for launching a fleet of EC2 instances. Which of the following changes needs to be done?

- Create a new launch configuration.
- Do nothing. You can start directly launching EC2 instances in the Auto Scaling group with the same launch configuration.
- Create a new target group.
- Create a new target group and launch configuration.

Incorrect

A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.



You can specify your launch configuration with multiple Auto Scaling groups. However, you can only specify one launch configuration for an Auto Scaling group at a time, and you can't modify a launch configuration after you've created it. Therefore, if you want to change the launch configuration for an Auto Scaling group, you must create a launch configuration and then update your Auto Scaling group with the new launch configuration.

For this scenario, you have to create a new launch configuration. Remember that **you can't modify a launch configuration after you've created it.**

Hence, the correct answer is: ***Create a new launch configuration.***

The option that says: ***Do nothing. You can start directly launching EC2 instances in the Auto Scaling group with the same launch configuration*** is incorrect because what you are trying to achieve is change the AMI being used by your fleet of EC2 instances. Therefore, you need to change the launch configuration to update what your instances are using.

The option that says: ***create a new target group*** and ***create a new target group and launch configuration*** are both incorrect because you only want to change the AMI being used by your instances, and not the instances themselves. Target groups are primarily used in ELBs and not in Auto Scaling. The scenario didn't mention that the architecture has a load balancer. Therefore, you should be updating your launch configuration, not the target group.

References:

<http://docs.aws.amazon.com/autoscaling/latest/userguide/LaunchConfiguration.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

6. QUESTION

Category: CSAA – Design Resilient Architectures

A commercial bank has a forex trading application. They created an Auto Scaling group of EC2 instances that allow the bank to cope with the current traffic and achieve cost-efficiency. They want the Auto Scaling group to behave in such a way that it will follow a predefined set of parameters before it scales down the number of EC2 instances, which protects the system from unintended slowdown or unavailability.

Which of the following statements are true regarding the cooldown period? (Select TWO.)

- Its default value is 600 seconds.
- Its default value is 300 seconds.
- It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.
- It ensures that before the Auto Scaling group scales out, the EC2 instances have an ample time to cooldown.
- It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.

Correct

In Auto Scaling, the following statements are correct regarding the cooldown period:

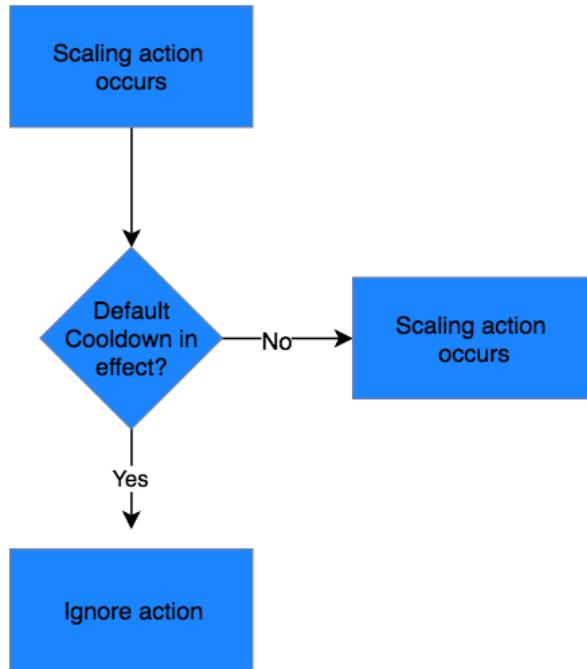
1. It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.
2. Its default value is 300 seconds.
3. It is a configurable setting for your Auto Scaling group.

The following options are incorrect:

- *– ***It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.****
- *– ***It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.****
- *– ***Its default value is 600 seconds.****

These statements are inaccurate and don't depict what the word "cooldown" actually means for Auto Scaling. The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect. After the Auto Scaling group dynamically scales using a simple scaling policy, it waits for the cooldown period to complete before resuming scaling activities.

The figure below demonstrates the scaling cooldown:



Reference:

<http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-termination.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

7. QUESTION

Category: CSAA – Design Resilient Architectures

A suite of web applications is hosted in an Auto Scaling group of EC2 instances across three Availability Zones and is configured with default settings. There is an Application Load Balancer that forwards the request to the respective target group on the URL path. The scale-in policy has been triggered due to the low number of incoming traffic to the application.

Which EC2 instance will be the first one to be terminated by your Auto Scaling group?

- The instance will be randomly selected by the Auto Scaling group
- The EC2 instance which has the least number of user sessions
- The EC2 instance which has been running for the longest time
- **The EC2 instance launched from the oldest launch configuration**

Incorrect

The default termination policy is designed to help ensure that your network architecture spans Availability Zones evenly. With the default termination policy, the behavior of the Auto Scaling group is as follows:

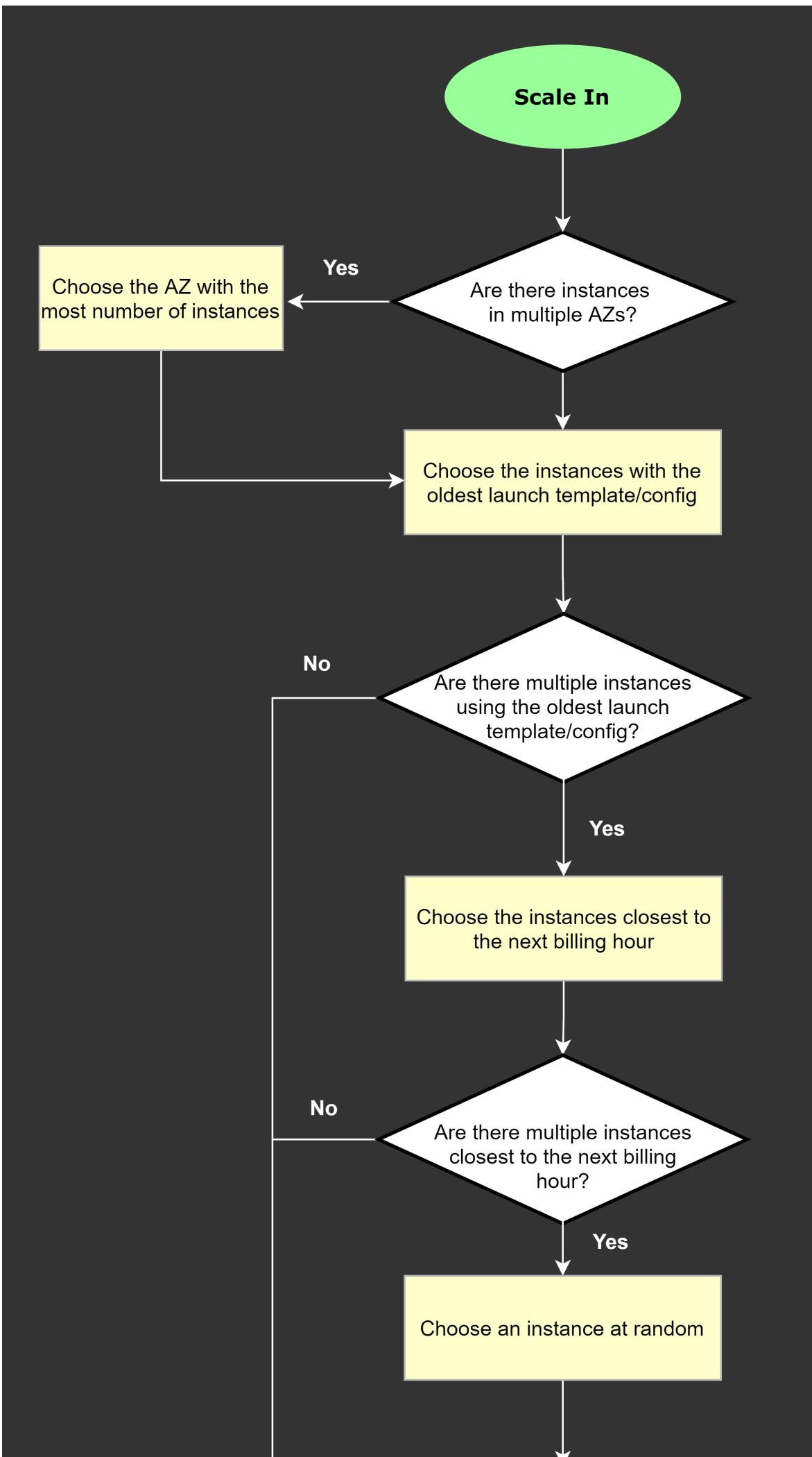
\1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that use the oldest launch configuration.

\2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.

\3. If there are multiple instances to terminate based on the above criteria, determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there is one such instance, terminate it.

\4. If there is more than one unprotected instance closest to the next billing hour, choose one of these instances at random.

The following flow diagram illustrates how the default termination policy works:



```
graph LR; A[ ] --> B[Terminate the instance]
```

Terminate the instance

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html#default-termination-policy>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

8. QUESTION

Category: CSAA – Design High-Performing Architectures

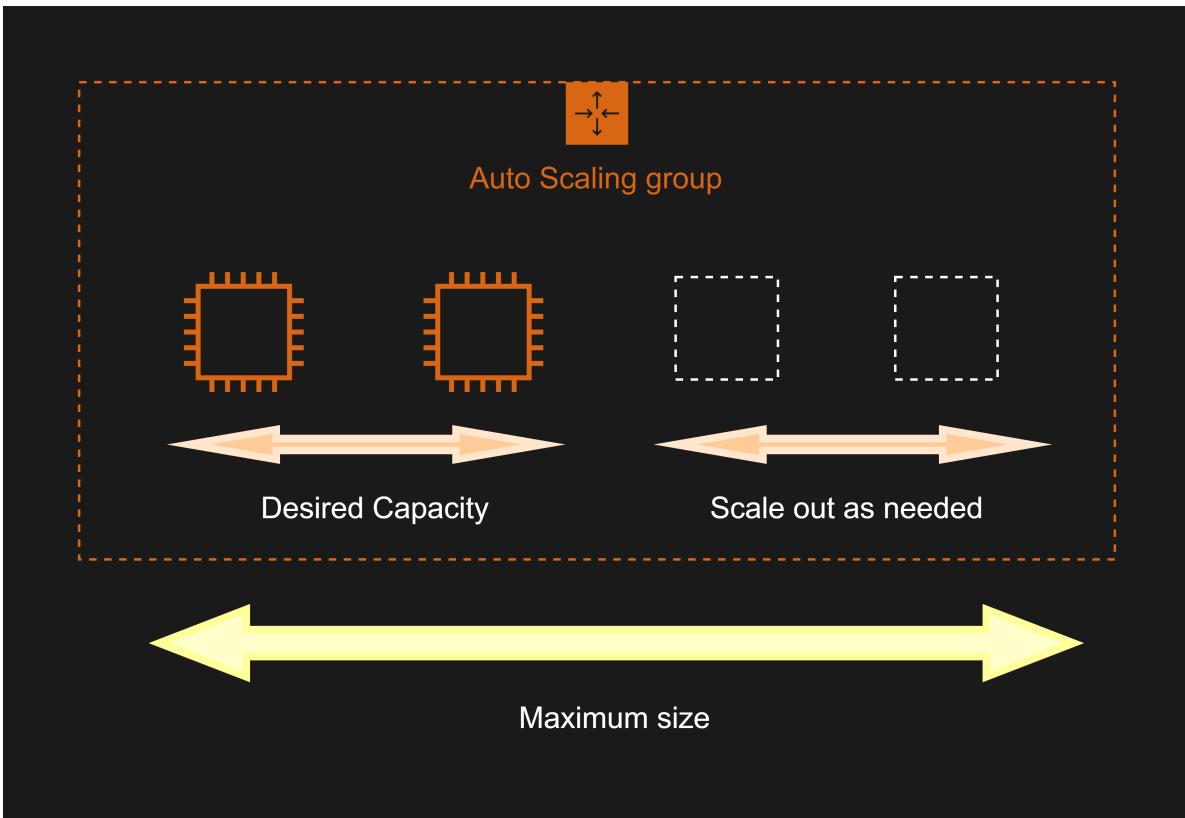
A tech company has a CRM application hosted on an Auto Scaling group of On-Demand EC2 instances. The application is extensively used during office hours from 9 in the morning till 5 in the afternoon. Their users are complaining that the performance of the application is slow during the start of the day but then works normally after a couple of hours.

Which of the following can be done to ensure that the application works properly at the beginning of the day?

- Configure a Scheduled scaling policy for the Auto Scaling group to launch new instances before the start of the day.
- Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the Memory utilization.
- Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization.
- Set up an Application Load Balancer (ALB) to your architecture to ensure that the traffic is properly distributed on the instances.

Correct

Scaling based on a schedule allows you to scale your application in response to predictable load changes. For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling activities based on the predictable traffic patterns of your web application.



To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. To create a scheduled scaling action, you specify the start time when the scaling action should take effect, and the new minimum, maximum, and desired sizes for the scaling action. At the specified time, Amazon EC2 Auto Scaling updates the group with the values for minimum, maximum, and desired size specified by the scaling action. You can create scheduled actions for scaling one time only or for scaling on a recurring schedule.

Hence, ***configuring a Scheduled scaling policy for the Auto Scaling group to launch new instances before the start of the day*** is the correct answer. You need to configure a Scheduled scaling policy. This will ensure that the instances are already scaled up and ready before the start of the day since this is when the application is used the most.

Configuring a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization and ***configuring a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the Memory utilization*** are both incorrect because although these are valid solutions, it is still better to configure a Scheduled scaling policy as you already know the exact peak hours of your application. By the time either the CPU or Memory hits a peak, the application already has performance issues, so you need to ensure the scaling is done beforehand using a Scheduled scaling policy.

Setting up an Application Load Balancer (ALB) to your architecture to ensure that the traffic is properly distributed on the instances is incorrect. Although the Application load balancer can also balance the traffic, it cannot increase the instances based on demand.

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/schedule_time.html

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

1. QUESTION

Category: CSAA – Design Resilient Architectures

A company is in the process of migrating their applications to AWS. One of their systems requires a database that can scale globally and handle frequent schema changes. The application should not have any downtime or performance issues whenever there is a schema change in the database. It should also provide a low latency response to high-traffic queries.

Which is the most suitable database solution to use to achieve this requirement?

- An Amazon RDS instance in Multi-AZ Deployments configuration
- **Amazon DynamoDB**
- An Amazon Aurora database with Read Replicas
- Redshift

Incorrect

Before we proceed in answering this question, we must first be clear with the actual definition of a “**schema**”. Basically, the english definition of a schema is: *a representation of a plan or theory in the form of an outline or model*.

Just think of a schema as the “structure” or a “model” of your data in your database. Since the scenario requires that the schema, or the structure of your data, changes frequently, then you have to pick a database which provides a non-rigid and flexible way of adding or removing new types of data. This is a classic example of choosing between a relational database and non-relational (NoSQL) database.

Characteristic	Relational Database Management System (RDBMS)	Amazon DynamoDB
Optimal Workloads	Ad hoc queries; data warehousing; OLAP (online analytical processing).	Web-scale applications, including social networks, gaming, media sharing, and IoT (Internet of Things).
Data Model	The relational model requires a well-defined schema, where data is normalized into tables, rows and columns. In addition, all of the relationships are defined among tables, columns, indexes, and other database elements.	DynamoDB is schemaless. Every table must have a primary key to uniquely identify each data item, but there are no similar constraints on other non-key attributes. DynamoDB can manage structured or semi-structured data, including JSON documents.
Data Access	SQL (Structured Query Language) is the standard for storing and retrieving data. Relational databases offer a rich set of tools for simplifying the development of database-driven applications, but all of these tools use SQL.	You can use the AWS Management Console or the AWS CLI to work with DynamoDB and perform ad hoc tasks. Applications can leverage the AWS software development kits (SDKs) to work with DynamoDB using object-based, document-centric, or low-level interfaces.
Performance	Relational databases are optimized for storage, so performance generally depends on the disk subsystem. Developers and database administrators must optimize queries, indexes, and table structures in order to achieve peak performance.	DynamoDB is optimized for compute, so performance is mainly a function of the underlying hardware and network latency. As a managed service, DynamoDB insulates you and your applications from these implementation details, so that you can focus on designing and building robust, high-performance applications.
Scaling	It is easiest to scale up with faster hardware. It is also possible for database tables to span across multiple hosts in a distributed system, but this requires additional investment. Relational databases have maximum sizes for the number and size of files, which imposes upper limits on scalability.	DynamoDB is designed to scale out using distributed clusters of hardware. This design allows increased throughput without increased latency. Customers specify their throughput requirements, and DynamoDB allocates sufficient resources to meet those requirements. There are no upper limits on the number of items per table, nor the total size of that table.

A relational database is known for having a rigid schema, with a lot of constraints and limits as to which (and what type of) data can be inserted or not. It is primarily used for scenarios where you have to support complex queries which fetch data across a number of tables. It is best for scenarios where you have complex table relationships but for use cases where you need to have a flexible schema, this is not a suitable database to use.

For NoSQL, it is not as rigid as a relational database because you can easily add or remove rows or elements in your table/collection entry. It also has a more flexible schema because it can store complex hierarchical data within a single item which, unlike a relational database, does not entail changing multiple related tables. Hence, the best answer to be used here is a NoSQL database, like DynamoDB. When your business requires a low-latency response to high-traffic queries, taking advantage of a NoSQL system generally makes technical and economic sense.

Amazon DynamoDB helps solve the problems that limit the relational system scalability by avoiding them. In DynamoDB, you design your schema specifically to make the most common and important queries as fast and as inexpensive as possible. Your data structures are tailored to the specific requirements of your business use cases.

Remember that a relational database system **does not scale** well for the following reasons:

- It normalizes data and stores it on multiple tables that require multiple queries to write to disk.
- It generally incurs the performance costs of an ACID-compliant transaction system.
- It uses expensive joins to reassemble required views of query results.

For DynamoDB, it scales well due to these reasons:

- Its **schema flexibility** lets DynamoDB store complex hierarchical data within a single item. DynamoDB is not a totally *schemaless* database since the very definition of a schema is just the model or structure of your data.
- Composite key design lets it store related items close together on the same table.

An Amazon RDS instance in Multi-AZ Deployments configuration and ***an Amazon Aurora database with Read Replicas*** are incorrect because both of them are a type of relational database.

Redshift is incorrect because it is primarily used for OLAP systems.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-general-nosql-design.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-relational-modeling.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SQLtoNoSQL.html>

Also check the **AWS Certified Solutions Architect Official Study Guide: Associate Exam** 1st Edition and turn to page 161 which talks about NoSQL Databases.

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

2. QUESTION

Category: CSAA – Design High-Performing Architectures

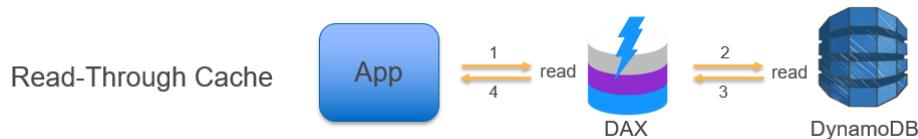
A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time.

To improve the game's performance, which AWS service can you use to reduce DynamoDB response times from milliseconds to microseconds?

- **Amazon DynamoDB Accelerator (DAX)**
- AWS Device Farm
- DynamoDB Auto Scaling
- Amazon ElastiCache

Correct

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache that can reduce Amazon DynamoDB response times from milliseconds to microseconds, even at millions of requests per second.



Amazon ElastiCache is incorrect because although you may use ElastiCache as your database cache, it will not reduce the DynamoDB response time from milliseconds to microseconds as compared with DynamoDB DAX.

AWS Device Farm is incorrect because this is an app testing service that lets you test and interact with your Android, iOS, and web apps on many devices at once, or reproduce issues on a device in real time.

DynamoDB Auto Scaling is incorrect because this is primarily used to automate capacity management for your tables and global secondary indexes.

References:

<https://aws.amazon.com/dynamodb/dax>

<https://aws.amazon.com/device-farm>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

A company currently has an Augment Reality (AR) mobile game that has a serverless backend. It is using a DynamoDB table which was launched using the AWS CLI to store all the user data and information gathered from the players and a Lambda function to pull the data from DynamoDB. The game is being used by millions of users each day to read and store data.

How would you design the application to improve its overall performance and make it more scalable while keeping the costs low? (Select TWO)

- Since Auto Scaling is enabled by default, the provisioned read and write capacity will adjust automatically. Also enable DynamoDB Accelerator (DAX) to improve the performance from milliseconds to microseconds.
- Use API Gateway in conjunction with Lambda and turn on the caching on frequently accessed data and enable DynamoDB global replication.
- Use AWS SSO and Cognito to authenticate users and have them directly access DynamoDB using single-sign on. Manually set the provisioned read and write capacity to a higher RCU and WCU.
- Configure CloudFront with DynamoDB as the origin; cache frequently accessed data on the client device using ElastiCache.
- Enable DynamoDB Accelerator (DAX) and ensure that the Auto Scaling is enabled and increase the maximum provisioned read and write capacity.

Incorrect

The correct answers are the options that say:

– Enable DynamoDB Accelerator (DAX) and ensure that the Auto Scaling is enabled and increase the maximum provisioned read and write capacity.

– Use API Gateway in conjunction with Lambda and turn on the caching on frequently accessed data and enable DynamoDB global replication.

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for DynamoDB that delivers up to a 10x performance improvement – from milliseconds to microseconds – even at millions of requests per second. DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management.

Movies Close

Overview Items Metrics Alarms Capacity Indexes Triggers Access control Tags

Scaling activities

Provisioned capacity

	Read capacity units	Write capacity units
Table	5	5

! Consumed read capacity >= 4 for 5 minutes

Estimated cost \$2.91 / month ([Capacity calculator](#))

Auto Scaling

<input checked="" type="checkbox"/> Read capacity	<input type="checkbox"/> Write capacity
Target utilization	70 %
Minimum provisioned capacity	5 units
Maximum provisioned capacity	40000 units
<input checked="" type="checkbox"/> Apply same settings to global secondary indexes	

IAM Role I authorize DynamoDB to scale capacity using the following role:

New role: DynamoDBAutoscaleRole
 Existing role with pre-defined policies [[Instructions](#)]

Role Name*

Save **Cancel**

Amazon API Gateway lets you create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as code running on AWS Lambda. Amazon API Gateway handles all of the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs.

AWS Lambda scales your functions automatically on your behalf. Every time an event notification is received for your function, AWS Lambda quickly locates free capacity within its compute fleet and runs your code. Since your code is stateless, AWS Lambda can start as many copies of your function as needed without lengthy deployment and configuration delays.

The option that says: ***Configure CloudFront with DynamoDB as the origin; cache frequently accessed data on the client device using ElastiCache*** is incorrect. Although CloudFront delivers content faster to your users using edge locations, you still cannot integrate DynamoDB table with CloudFront as these two are incompatible.

The option that says: ***Use AWS SSO and Cognito to authenticate users and have them directly access DynamoDB using single-sign on. Manually set the provisioned read and write capacity to a higher RCU and WCU*** is incorrect because AWS Single Sign-On (SSO) is a cloud SSO service that just makes it easy to centrally manage SSO access to multiple AWS

accounts and business applications. This will not be of much help on the scalability and performance of the application. It is costly to manually set the provisioned read and write capacity to a higher RCU and WCU because this capacity will run round the clock and will still be the same even if the incoming traffic is stable and there is no need to scale.

The option that says: ***Since Auto Scaling is enabled by default, the provisioned read and write capacity will adjust automatically. Also enable DynamoDB Accelerator (DAX) to improve the performance from milliseconds to microseconds*** is incorrect because, by default, Auto Scaling is not enabled in a DynamoDB table which is created using the AWS CLI.

References:

<https://aws.amazon.com/lambda/faqs/>

<https://aws.amazon.com/api-gateway/faqs/>

<https://aws.amazon.com/dynamodb/dax/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

4. QUESTION

Category: CSAA – Design High-Performing Architectures

In a startup company you are working for, you are asked to design a web application that requires a NoSQL database that has no limit on the storage size for a given table. The startup is still new in the market and it has very limited human resources who can take care of the database infrastructure.

Which is the most suitable service that you can implement that provides a fully managed, scalable and highly available NoSQL service?

- Amazon Neptune
- **DynamoDB**
- SimpleDB
- Amazon Aurora

Correct

The term “**fully managed**” means that Amazon will manage the underlying infrastructure of the service hence, you don’t need an additional human resource to support or maintain the service. Therefore, Amazon DynamoDB is the right answer. Remember that Amazon RDS is a managed service but not “fully managed” as you still have the option to maintain and configure the underlying server of the database.

Amazon Neptune is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity make it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications.

Amazon Neptune is incorrect because this is primarily used as a graph database.

Amazon Aurora is incorrect because this is a relational database and not a NoSQL database.

SimpleDB is incorrect. Although SimpleDB is also a highly available and scalable NoSQL database, it has a limit on the request capacity or storage size for a given table, unlike DynamoDB.

Reference:

<https://aws.amazon.com/dynamodb/>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Amazon DynamoDB Overview:

<https://youtu.be/3ZOyUNleorU>

5. QUESTION

Category: CSAA – Design High-Performing Architectures

A popular social network is hosted in AWS and is using a DynamoDB table as its database. There is a requirement to implement a ‘follow’ feature where users can subscribe to certain updates made by a particular user and be notified via email.

Which of the following is the most suitable solution that you should implement to meet the requirement?

- Using the Kinesis Client Library (KCL), write an application that leverages on DynamoDB Streams Kinesis Adapter that will fetch data from the DynamoDB Streams endpoint. When there are updates made by a particular user, notify the subscribers via email using SNS.
- Create a Lambda function that uses DynamoDB Streams Kinesis Adapter which will fetch data from the DynamoDB Streams endpoint. Set up an SNS Topic that will notify the subscribers via email when there is an update made by a particular user.
- Set up a DAX cluster to access the source DynamoDB table. Create a new DynamoDB trigger and a Lambda function. For every update made in the user data, the trigger will send data to the Lambda function which will then notify the subscribers via email using SNS.
- **Enable DynamoDB Stream and create an AWS Lambda trigger, as well as the IAM role which contains all of the permissions that the Lambda function will need at runtime. The data from the stream record will be processed by the Lambda function which will then publish a message to SNS Topic that will notify the subscribers via email.**

Correct

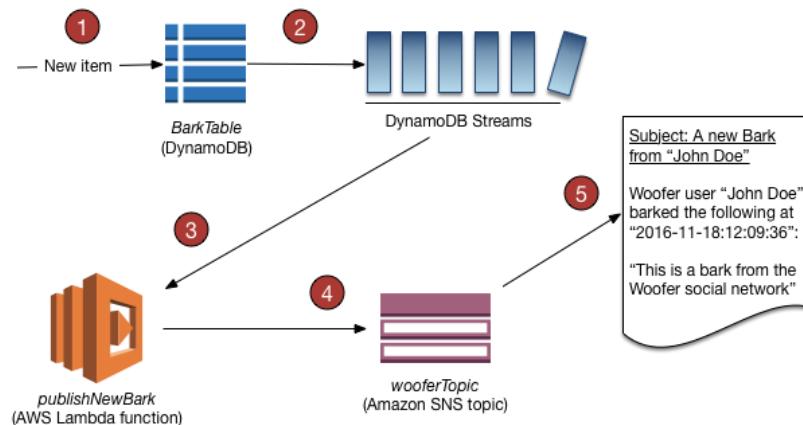
A **DynamoDB stream** is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A *stream record* contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the “before” and “after” images of modified items.

Amazon DynamoDB is integrated with AWS Lambda so that you can create *triggers*—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables.

If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table’s stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. The Lambda function can perform any actions you specify, such as sending a notification or initiating a workflow.

Hence, the correct answer in this scenario is the option that says: ***Enable DynamoDB Stream and create an AWS Lambda trigger, as well as the IAM role which contains all of the permissions that the Lambda function will need at runtime. The data from the stream record will be processed by the Lambda function which will then publish a message to SNS Topic that will notify the subscribers via email***.



The option that says: ***Using the Kinesis Client Library (KCL), write an application that leverages on DynamoDB Streams Kinesis Adapter that will fetch data from the DynamoDB Streams endpoint. When there are updates made by a particular user, notify the subscribers via email using SNS*** is incorrect because although this is a valid solution, it is missing a vital step which is to enable DynamoDB Streams. With the DynamoDB Streams Kinesis Adapter in place, you can begin developing applications via the KCL interface, with the API calls seamlessly directed at the DynamoDB Streams endpoint. Remember that the DynamoDB Stream feature is not enabled by default.

The option that says: ***Create a Lambda function that uses DynamoDB Streams Kinesis Adapter which will fetch data from the DynamoDB Streams endpoint. Set up an SNS Topic that will notify the subscribers via email when there is an update made by a particular user*** is incorrect because just like in the above, you have to manually enable DynamoDB Streams first before you can use its endpoint.

The option that says: ***Set up a DAX cluster to access the source DynamoDB table. Create a new DynamoDB trigger and a Lambda function. For every update made in the user data, the trigger will send data to the Lambda function which will then notify the subscribers via email using SNS*** is incorrect because the DynamoDB Accelerator (DAX) feature is primarily used to significantly improve the in-memory read performance of your database, and not to capture the time-ordered sequence of item-level modifications. You should use DynamoDB Streams in this scenario instead.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.Tutorial.html>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A Docker application, which is running on an Amazon ECS cluster behind a load balancer, is heavily using DynamoDB. You are instructed to improve the database performance by distributing the workload evenly and using the provisioned throughput efficiently.

Which of the following would you consider to implement for your DynamoDB table?

- Reduce the number of partition keys in the DynamoDB table.
- Use partition keys with high-cardinality attributes, which have a large number of distinct values for each item.
- Use partition keys with low-cardinality attributes, which have a few number of distinct values for each item.
- Avoid using a composite primary key, which is composed of a partition key and a sort key.

Correct

The partition key portion of a table's primary key determines the logical partitions in which a table's data is stored. This in turn affects the underlying physical partitions. Provisioned I/O capacity for the table is divided evenly among these physical partitions. Therefore a partition key design that doesn't distribute I/O requests evenly can create "hot" partitions that result in throttling and use your provisioned I/O capacity inefficiently.

The optimal usage of a table's provisioned throughput depends not only on the workload patterns of individual items, but also on the partition-key design. This doesn't mean that you must access all partition key values to achieve an efficient throughput level, or even that the percentage of accessed partition key values must be high. It does mean that the more distinct partition key values that your workload accesses, the more those requests will be spread across the partitioned space. In general, you will use your provisioned throughput more efficiently as the ratio of partition key values accessed to the total number of partition key values increases.

One example for this is the use of ***partition keys with high-cardinality attributes, which have a large number of distinct values for each item***.

Reducing the number of partition keys in the DynamoDB table is incorrect. Instead of doing this, you should actually add more to improve its performance to distribute the I/O requests evenly and not avoid “hot” partitions.

Using partition keys with low-cardinality attributes, which have a few number of distinct values for each item is incorrect because this is the exact opposite of the correct answer. Remember that the more distinct partition key values your workload accesses, the more those requests will be spread across the partitioned space. Conversely, the less distinct partition key values, the less evenly spread it would be across the partitioned space, which effectively slows the performance.

The option that says: ***Avoid using a composite primary key, which is composed of a partition key and a sort key*** is incorrect because as mentioned, a composite primary key will provide more partition for the table and in turn, improves the performance. Hence, it should be used and not avoided.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-uniform-load.html>

<https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Amazon DynamoDB Overview:

<https://youtu.be/3ZOyUNleorU>

7. QUESTION

Category: CSAA – Design High-Performing Architectures

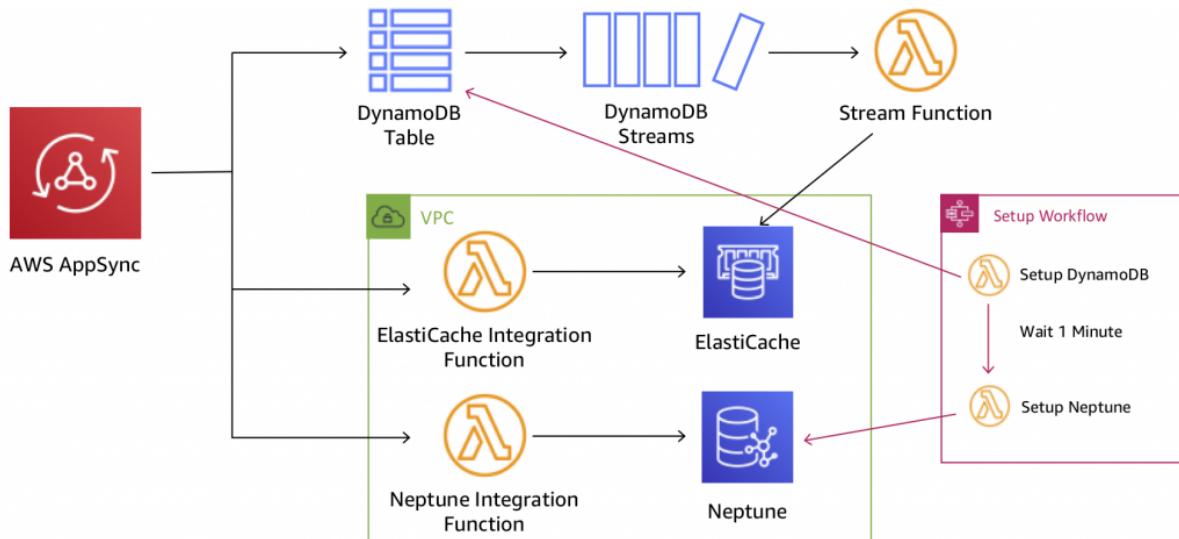
A Solutions Architect needs to deploy a mobile application that can collect votes for a popular singing competition. Millions of users from around the world will submit votes using their mobile phones. These votes must be collected and stored in a highly scalable and highly available data store which will be queried for real-time ranking.

Which of the following combination of services should the architect use to meet this requirement?

- Amazon Redshift and AWS Mobile Hub
- Amazon Aurora and Amazon Cognito
- **Amazon DynamoDB and AWS AppSync**
- Amazon Relational Database Service (RDS) and Amazon MQ

Correct

When the word durability pops out, the first service that should come to your mind is Amazon S3. Since this service is not available in the answer options, we can look at the other data store available which is Amazon DynamoDB.



DynamoDB is durable, scalable, and highly available data store which can be used for real-time tabulation. You can also use ***AppSync*** with DynamoDB to make it easy for you to build collaborative apps that keep shared data updated in real time. You just specify the data for your app with simple code statements and AWS AppSync manages everything needed to keep the app data updated in real time. This will allow your app to access data in Amazon DynamoDB, trigger AWS Lambda functions, or run Amazon Elasticsearch queries and combine data from these services to provide the exact data you need for your app.

Amazon Redshift and AWS Mobile Hub are incorrect as Amazon Redshift is mainly used as a data warehouse and for online analytic processing (*OLAP*). Although this service can be used for this scenario, DynamoDB is still the top choice given its better durability and scalability.

Amazon Relational Database Service (RDS) and Amazon MQ and ***Amazon Aurora and Amazon Cognito*** are possible answers in this scenario, however, DynamoDB is much more suitable for simple mobile apps that do not have complicated data relationships compared with enterprise web applications. It is stated in the scenario that the mobile app will be used from around the world, which is why you need a data storage service which can be supported globally. It would be a management overhead to implement multi-region deployment for your RDS and Aurora database instances compared to using the Global table feature of DynamoDB.

References:

<https://aws.amazon.com/dynamodb/faqs/>

<https://aws.amazon.com/appsync/>

Amazon DynamoDB Overview:

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

8. QUESTION

Category: CSAA – Design High-Performing Architectures

A leading IT consulting company has an application which processes a large stream of financial data by an Amazon ECS Cluster then stores the result to a DynamoDB table. You have to design a solution to detect new entries in the DynamoDB table then automatically trigger a Lambda function to run some tests to verify the processed data.

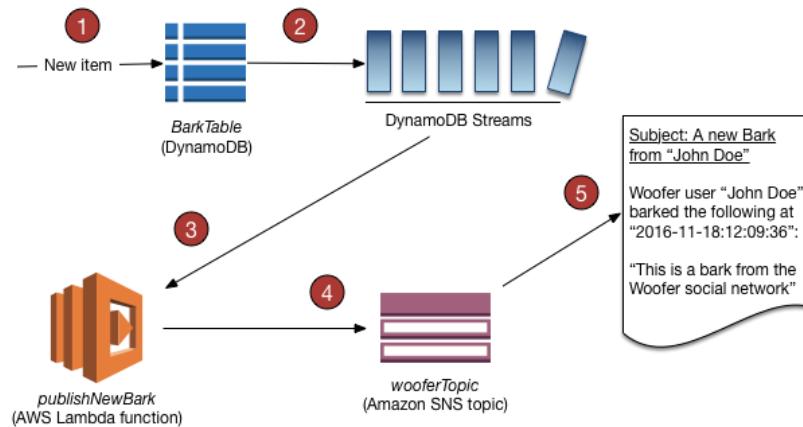
What solution can be easily implemented to alert the Lambda function of new entries while requiring minimal configuration change to your architecture?

- Use Systems Manager Automation to detect new entries in the DynamoDB table then automatically invoke the Lambda function for processing.
- **Enable DynamoDB Streams to capture table activity and automatically trigger the Lambda function.**
- Invoke the Lambda functions using SNS each time that the ECS Cluster successfully processed financial data.
- Use CloudWatch Alarms to trigger the Lambda function whenever a new entry is created in the DynamoDB table.

Correct

Amazon DynamoDB is integrated with AWS Lambda so that you can create *triggers*—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables.

If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table's stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records.



You can create a Lambda function which can perform a specific action that you specify, such as sending a notification or initiating a workflow. For instance, you can set up a Lambda function to simply copy each stream record to persistent storage, such as EFS or S3, to create a permanent audit trail of write activity in your table.

Suppose you have a mobile gaming app that writes to a `TutorialsDojoCourses` table. Whenever the `Topcourse` attribute of the `TutorialsDojoscores` table is updated, a corresponding stream record is written to the table's stream. This event could then trigger a Lambda function that posts a congratulatory message on a social media network. (The function would simply ignore any stream records that are not updates to `TutorialsDojoCourses` or that do not modify the `TopCourse` attribute.)

Hence, ***enabling DynamoDB Streams to capture table activity and automatically trigger the Lambda function*** is the correct answer because the requirement can be met with minimal configuration change using DynamoDB streams which can automatically trigger Lambda functions whenever there is a new entry.

Using CloudWatch Alarms to trigger the Lambda function whenever a new entry is created in the DynamoDB table is incorrect because CloudWatch Alarms only monitor service metrics, not changes in DynamoDB table data.

Invoking the Lambda functions using SNS each time that the ECS Cluster successfully processed financial data is incorrect because you don't need to create an SNS topic just to invoke Lambda functions. You can enable DynamoDB streams instead to meet the requirement with less configuration.

Using Systems Manager Automation to detect new entries in the DynamoDB table then automatically invoking the Lambda function for processing is incorrect because the Systems Manager Automation service is primarily used to simplify common maintenance and deployment tasks of Amazon EC2 instances and other AWS resources. It does not have

the capability to detect new entries in a DynamoDB table.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>

Check out this Amazon DynamoDB cheat sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An application is hosted in an Auto Scaling group of EC2 instances and a Microsoft SQL Server on Amazon RDS. There is a requirement that all in-flight data between your web servers and RDS should be secured.

Which of the following options is the MOST suitable solution that you should implement? (Select TWO.)

- Download the Amazon RDS Root CA certificate. Import the certificate to your servers and configure your application to use SSL to encrypt the connection to RDS.
- Configure the security groups of your EC2 instances and RDS to only allow traffic to and from port 443.
- Enable the IAM DB authentication in RDS using the AWS Management Console.
- Force all connections to your DB instance to use SSL by setting the `rds.force_ssl` parameter to true. Once done, reboot your DB instance.
- Specify the TDE option in an RDS option group that is associated with that DB instance to enable transparent data encryption (TDE).

Incorrect

You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS regions for all supported SQL Server editions.

When you create an SQL Server DB instance, Amazon RDS creates an SSL certificate for it. The SSL certificate includes the DB instance endpoint as the Common Name (CN) for the SSL certificate to guard against spoofing attacks.

There are 2 ways to use SSL to connect to your SQL Server DB instance:

- Force SSL for all connections — this happens transparently to the client, and the client doesn't have to do any work to use SSL.
- Encrypt specific connections — this sets up an SSL connection from a specific client computer, and you must do work on the client to encrypt connections.

Console1 - [Console Root\Certificates (Local Computer)\Trusted Root Certification Authorities\Certificates]					
File Action View Favorites Window Help					
		Issued To	Issued By	Expiration Date	
				Intended Purposes	
		AddTrust External CA Root	AddTrust External CA Root	5/30/2020	Server Authentication
		Amazon Corporate Systems Cert...	Amazon.com Internal Root Certific...	9/20/2018	<All>
		Amazon Corporate Systems Cert...	Amazon.com Internal Root Certific...	10/9/2018	<All>
		Amazon RDS Root 2019 CA	Amazon RDS Root 2019 CA	8/22/2024	<All>

You can force all connections to your DB instance to use SSL, or you can encrypt connections from specific client computers only. To use SSL from a specific client, you must obtain certificates for the client computer, import certificates on the client computer, and then encrypt the connections from the client computer.

If you want to force SSL, use the `rds.force_ssl` parameter. By default, the `rds.force_ssl` parameter is set to `false`. Set the `rds.force_ssl` parameter to `true` to force connections to use SSL. The `rds.force_ssl` parameter is static, so after you change the value, you must reboot your DB instance for the change to take effect.

Hence, the correct answers for this scenario are the options that say:

– Force all connections to your DB instance to use SSL by setting the `rds.force_ssl` parameter to true. Once done, reboot your DB instance.

– Download the Amazon RDS Root CA certificate. Import the certificate to your servers and configure your application to use SSL to encrypt the connection to RDS.

Specifying the TDE option in an RDS option group that is associated with that DB instance to enable transparent data encryption (TDE) is incorrect because transparent data encryption (TDE) is primarily used to encrypt stored data on your DB instances running Microsoft SQL Server, and not the data that are in transit.

Enabling the IAM DB authentication in RDS using the AWS Management Console is incorrect because IAM database authentication is only supported in MySQL and PostgreSQL database engines. With IAM database authentication, you don't need to use a password when you connect to a DB instance but instead, you use an authentication token.

Configuring the security groups of your EC2 instances and RDS to only allow traffic to and from port 443 is incorrect because it is not enough to do this. You need to either force all connections to your DB instance to use SSL, or you can encrypt connections from specific client computers, just as mentioned above.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/SQLServer.Concepts.General_SSL.Using.html

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.SQLServer.Options_TDE.html

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

Due to the large volume of query requests, the database performance of an online reporting application significantly slowed down. The Solutions Architect is trying to convince her client to use Amazon RDS Read Replica for their application instead of setting up a Multi-AZ Deployments configuration.

What are two benefits of using Read Replicas over Multi-AZ that the Architect should point out? (Select TWO.)

- It elastically scales out beyond the capacity constraints of a single DB instance for read-heavy database workloads.
- Provides synchronous replication and automatic failover in the case of Availability Zone service failures.
- It enhances the read performance of your primary database by increasing its IOPS and accelerates its query processing via AWS Global Accelerator.
- Allows both read and write operations on the read replica to complement the primary database.
- Provides asynchronous replication and improves the performance of the primary database by taking read-heavy database workloads from it.

Incorrect

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads.

You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. Read replicas can also be promoted when needed to become standalone DB instances.

For the MySQL, MariaDB, PostgreSQL, and Oracle database engines, Amazon RDS creates a second DB instance using a snapshot of the source DB instance. It then uses the engines' native asynchronous replication to update the read replica whenever there is a change to the source DB instance. The read replica operates as a DB instance that allows only read-only connections; applications can connect to a read replica just as they would to any DB instance. Amazon RDS replicates all databases in the source DB instance.

Multi-AZ deployments	Multi-Region deployments	Read replicas
Main purpose is high availability	Main purpose is disaster recovery and local performance	Main purpose is scalability
Non-Aurora: synchronous replication; Aurora: asynchronous replication	Asynchronous replication	Asynchronous replication
Non-Aurora: only the primary instance is active; Aurora: all instances are active	All regions are accessible and can be used for reads	All read replicas are accessible and can be used for readscaling
Non-Aurora: automated backups are taken from standby; Aurora: automated backups are taken from shared storage layer	Automated backups can be taken in each region	No backups configured by default
Always span at least two Availability Zones within a single region	Each region can have a Multi-AZ deployment	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together	Non-Aurora: database engine version upgrade is independent from source instance; Aurora: all instances are updated together
Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected	Aurora allows promotion of a secondary region to be the master	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

When you create a read replica for Amazon RDS for MySQL, MariaDB, PostgreSQL, and Oracle, Amazon RDS sets up a secure communications channel using public-key encryption between the source DB instance and the read replica, even when replicating across regions. Amazon RDS establishes any AWS security configurations such as adding security

group entries needed to enable the secure channel.

You can also create read replicas within a Region or between Regions for your Amazon RDS for MySQL, MariaDB, PostgreSQL, and Oracle database instances encrypted at rest with AWS Key Management Service (KMS).

Hence, the correct answers are:

- It elastically scales out beyond the capacity constraints of a single DB instance for read-heavy database workloads.

- Provides asynchronous replication and improves the performance of the primary database by taking read-heavy database workloads from it.

The option that says: ***Allows both read and write operations on the read replica to complement the primary database*** is incorrect as Read Replicas are primarily used to offload read-only operations from the primary database instance. By default, you can't do a write operation to your Read Replica.

The option that says: ***Provides synchronous replication and automatic failover in the case of Availability Zone service failures*** is incorrect as this is a benefit of Multi-AZ and not of a Read Replica. Moreover, Read Replicas provide an asynchronous type of replication and not synchronous replication.

The option that says: ***It enhances the read performance of your primary database by increasing its IOPS and accelerates its query processing via AWS Global Accelerator*** is incorrect because Read Replicas do not do anything to upgrade or increase the read throughput on the primary DB instance per se, but it provides a way for your application to fetch data from replicas. In this way, it improves the overall performance of your entire database-tier (and not just the primary DB instance). It doesn't increase the IOPS nor use AWS Global Accelerator to accelerate the compute capacity of your primary database. AWS Global Accelerator is a networking service, not related to RDS, that direct user traffic to the nearest application endpoint to the client, thus reducing internet latency and jitter. It simply routes the traffic to the closest edge location via Anycast.

References:

<https://aws.amazon.com/rds/details/read-replicas/>

<https://aws.amazon.com/rds/features/multi-az/>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

Additional tutorial - How do I make my RDS MySQL read replica writable?

<https://youtu.be/j5da6d2TIPc>

3. QUESTION

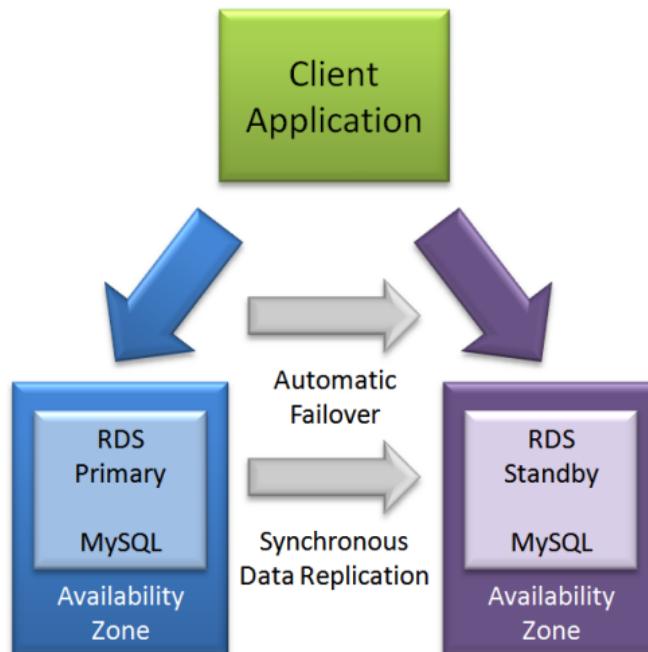
Category: CSAA – Design Resilient Architectures

An accounting application uses an RDS database configured with Multi-AZ deployments to improve availability. What would happen to RDS if the primary database instance fails?

- The canonical name record (CNAME) is switched from the primary to standby instance.
- A new database instance is created in the standby Availability Zone.
- The primary database instance will reboot.
- The IP address of the primary DB instance is switched to the standby DB instance.

Correct

In **Amazon RDS**, failover is automatically handled so that you can resume database operations as quickly as possible without administrative intervention in the event that your primary database instance went down. When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary.



The option that says: ***The IP address of the primary DB instance is switched to the standby DB instance*** is incorrect since IP addresses are per subnet, and subnets cannot span multiple AZs.

The option that says: ***The primary database instance will reboot*** is incorrect since in the event of a failure, there is no database to reboot with.

The option that says: ***A new database instance is created in the standby Availability Zone*** is incorrect since with multi-AZ enabled, you already have a standby database in another AZ.

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://aws.amazon.com/rds/faqs/>

Amazon RDS Overview:

Check out this Amazon RDS Cheat Sheet:

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A financial application is composed of an Auto Scaling group of EC2 instances, an Application Load Balancer, and a MySQL RDS instance in a Multi-AZ Deployments configuration. To protect the confidential data of your customers, you have to ensure that your RDS database can only be accessed using the profile credentials specific to your EC2 instances via an authentication token.

As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- **Enable the IAM DB Authentication.**
- Configure SSL in your application to encrypt the database connection to RDS.
- Create an IAM Role and assign it to your EC2 instances which will grant exclusive access to your RDS instance.
- Use a combination of IAM and STS to restrict access to your RDS instance via a temporary token.

Correct

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An **authentication token** is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

Database options

DB cluster identifier [Info](#)
tutorialsdojo
If you do not provide one, a default identifier based on the instance identifier will be used.

Database name [Info](#)
tutorialsdojo
If you do not specify a database name, Amazon RDS does not create a database.

Port [Info](#)
TCP/IP port the DB instance will use for application connections.
3306

DB parameter group [Info](#)
default.aurora5.6

DB cluster parameter group [Info](#)
default.aurora5.6

Option group [Info](#)
default:aurora-5-6

IAM DB authentication [Info](#)
 Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
 Disable

IAM database authentication provides the following benefits:

1. Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
2. You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.

3. For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security

Hence, ***enabling IAM DB Authentication*** is the correct answer based on the above reference.

Configuring SSL in your application to encrypt the database connection to RDS is incorrect because an SSL connection is not using an authentication token from IAM. Although configuring SSL to your application can improve the security of your data in flight, it is still not a suitable option to use in this scenario.

Creating an IAM Role and assigning it to your EC2 instances which will grant exclusive access to your RDS instance is incorrect because although you can create and assign an IAM Role to your EC2 instances, you still need to configure your RDS to use IAM DB Authentication.

Using a combination of IAM and STS to restrict access to your RDS instance via a temporary token is incorrect because you have to use IAM DB Authentication for this scenario, and not a combination of an IAM and STS. Although STS is used to send temporary tokens for authentication, this is not a compatible use case for RDS.

Reference:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html>

Check out this Amazon RDS cheat sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

5. QUESTION

Category: CSAA – Design High-Performing Architectures

A company launched a global news website that is deployed to AWS and is using MySQL RDS. The website has millions of viewers from all over the world which means that the website has read-heavy database workloads. All database transactions must be ACID compliant to ensure data integrity.

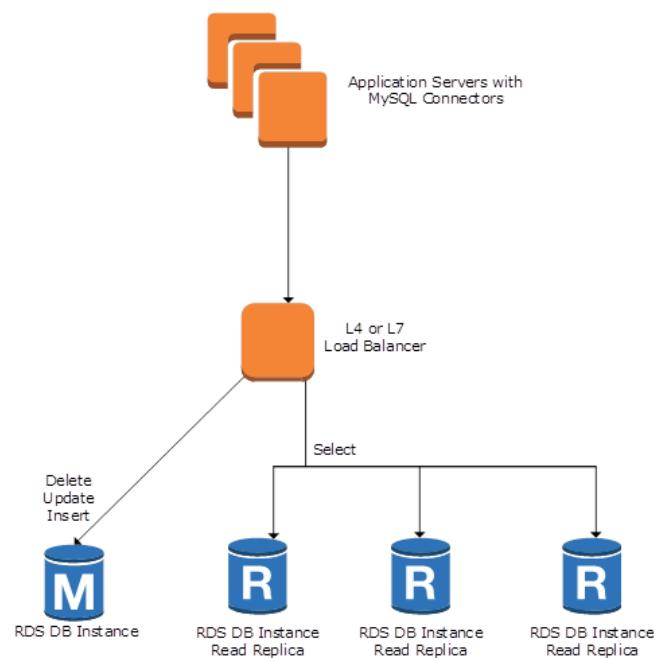
In this scenario, which of the following is the best option to use to increase the read throughput on the MySQL database?

- Enable Multi-AZ deployments
- Use SQS to queue up the requests
- Enable Amazon RDS Standby Replicas
- **Enable Amazon RDS Read Replicas**

Correct

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput.

Read replicas can also be promoted when needed to become standalone DB instances. Read replicas are available in Amazon RDS for MySQL, MariaDB, Oracle, and PostgreSQL as well as Amazon Aurora.



Enabling Multi-AZ deployments is incorrect because the Multi-AZ deployments feature is mainly used to achieve high availability and failover support for your database.

Enabling Amazon RDS Standby Replicas is incorrect because a Standby replica is used in Multi-AZ deployments and hence, it is not a solution to reduce read-heavy database workloads.

Using SQS to queue up the requests is incorrect. Although an SQS queue can effectively manage the requests, it won't be able to entirely improve the read-throughput of the database by itself.

References:

<https://aws.amazon.com/rds/details/read-replicas/>

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html

Amazon RDS Overview:

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

6. QUESTION

Category: CSAA – Design Resilient Architectures

An application that records weather data every minute is deployed in a fleet of Spot EC2 instances and uses a MySQL RDS database instance. Currently, there is only one RDS instance running in one Availability Zone. You plan to improve the database to ensure high availability by synchronous data replication to another RDS instance.

Which of the following performs synchronous data replication in RDS?

- **RDS DB instance running as a Multi-AZ deployment**
- RDS Read Replica
- DynamoDB Read Replica
- CloudFront running as a Multi-AZ deployment

Correct

When you create or modify your DB instance to run as a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous **standby** replica in a different Availability Zone. Updates to your DB Instance are synchronously replicated across Availability Zones to the standby in order to keep both in sync and protect your latest database updates against DB instance failure.

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance

RDS Read Replica is incorrect as a Read Replica provides an asynchronous replication instead of synchronous.

DynamoDB Read Replica and ***CloudFront running as a Multi-AZ deployment*** are incorrect as both DynamoDB and CloudFront do not have a Read Replica feature.

Reference:

<https://aws.amazon.com/rds/details/multi-az/>

Amazon RDS Overview:

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An online events registration system is hosted in AWS and uses ECS to host its front-end tier and an RDS configured with Multi-AZ for its database tier. What are the events that will make Amazon RDS automatically perform a failover to the standby replica? (Select TWO.)

- Loss of availability in primary Availability Zone
- Storage failure on primary

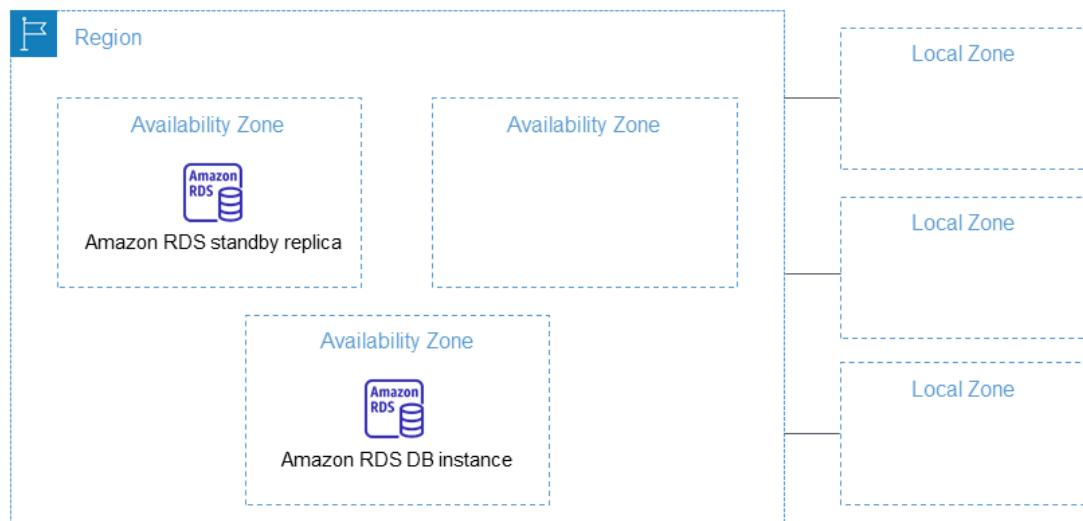
- Storage failure on secondary DB instance
- In the event of Read Replica failure
- Compute unit failure on secondary DB instance

Correct

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM).

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption.

Amazon RDS detects and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can resume database operations as quickly as possible without administrative intervention.



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.

Amazon RDS automatically performs a failover in the event of any of the following:

1. Loss of availability in primary Availability Zone.
2. Loss of network connectivity to primary.
3. Compute unit failure on primary.
4. Storage failure on primary.

Hence, the correct answers are:

- *- ***Loss of availability in primary Availability Zone****
- *- ***Storage failure on primary****

The following options are incorrect because all these scenarios do not affect the primary database. Automatic failover only occurs if the primary database is the one that is affected.

– Storage failure on secondary DB instance

– In the event of Read Replica failure

– Compute unit failure on secondary DB instance

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

8. QUESTION

Category: CSAA – Design Resilient Architectures

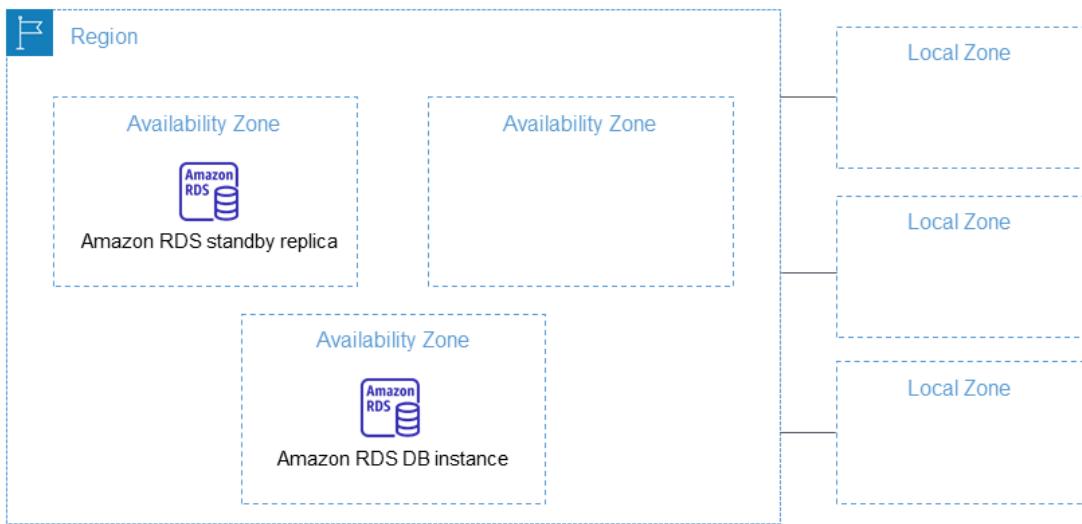
A Forex trading platform, which frequently processes and stores global financial data every minute, is hosted in your on-premises data center and uses an Oracle database. Due to a recent cooling problem in their data center, the company urgently needs to migrate their infrastructure to AWS to improve the performance of their applications. As the Solutions Architect, you are responsible in ensuring that the database is properly migrated and should remain available in case of database server failure in the future.

Which of the following is the most suitable solution to meet the requirement?

- Convert the database schema using the AWS Schema Conversion Tool and AWS Database Migration Service. Migrate the Oracle database to a non-cluster Amazon Aurora with a single instance.
- **Create an Oracle database in RDS with Multi-AZ deployments.**
- Launch an Oracle Real Application Clusters (RAC) in RDS.
- Launch an Oracle database instance in RDS with Recovery Manager (RMAN) enabled.

Correct

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.



In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora), so that you can resume database operations as soon as the failover is complete. Since the endpoint for your DB Instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention.

In this scenario, the best RDS configuration to use is an Oracle database in RDS with Multi-AZ deployments to ensure high availability even if the primary database instance goes down. Hence, ***creating an Oracle database in RDS with Multi-AZ deployments*** is the correct answer.

Launching an Oracle database instance in RDS with Recovery Manager (RMAN) enabled and ***launching an Oracle Real Application Clusters (RAC) in RDS*** are incorrect because Oracle RMAN and RAC are not supported in RDS.

The option that says: ***Convert the database schema using the AWS Schema Conversion Tool and AWS Database Migration Service. Migrate the Oracle database to a non-cluster Amazon Aurora with a single instance*** is incorrect because although this solution is feasible, it takes time to migrate your Oracle database to Aurora, which is not acceptable. Based on this option, the Aurora database is only using a single instance with no Read Replica and is not configured as an Amazon Aurora DB cluster, which could have improved the availability of the database.

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A game development company operates several virtual reality (VR) and augmented reality (AR) games which use various RESTful web APIs hosted on their on-premises data center. Due to the unprecedented growth of their company, they decided to migrate their system to AWS Cloud to scale out their resources as well to minimize costs.

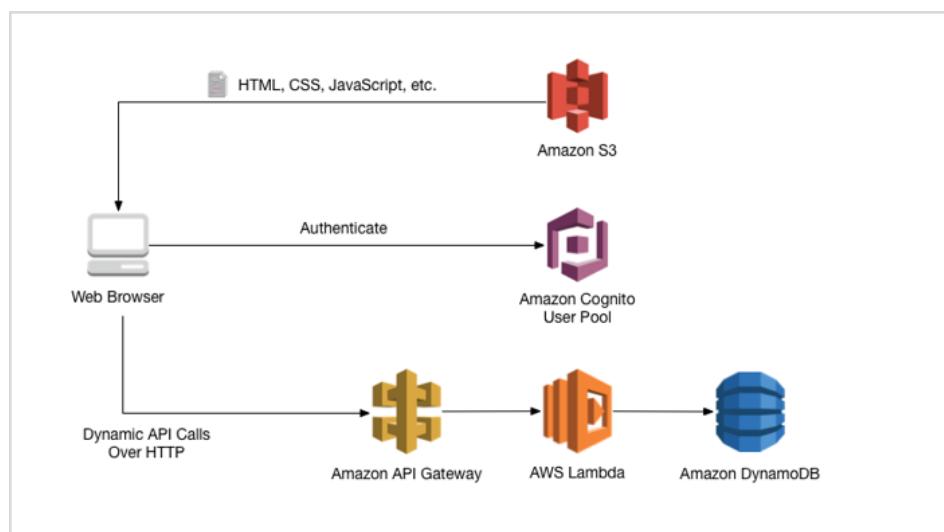
Which of the following should you recommend as the most cost-effective and scalable solution to meet the above requirement?

- Use a Spot Fleet of Amazon EC2 instances, each with an Elastic Fabric Adapter (EFA) for more consistent latency and higher network throughput. Set up an Application Load Balancer to distribute traffic to the instances.
- Set up a micro-service architecture with ECS, ECR, and Fargate.
- **Use AWS Lambda and Amazon API Gateway.**
- Host the APIs in a static S3 web hosting bucket behind a CloudFront web distribution.

Correct

With AWS Lambda, you pay only for what you use. You are charged based on the number of requests for your functions and the duration, the time it takes for your code to execute.

Lambda counts a request each time it starts executing in response to an event notification or invoke call, including test invokes from the console. You are charged for the total number of requests across all your functions. Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 100ms. The price depends on the amount of memory you allocate to your function. The Lambda free tier includes 1M free requests per month and over 400,000 GB-seconds of compute time per month.



The best possible answer here is to use Lambda and API Gateway because this solution is both scalable and cost-effective. You will only be charged when you use your Lambda function, unlike having an EC2 instance which always runs even though you don't use it.

Setting up a micro-service architecture with ECS, ECR, and Fargate is incorrect because ECS is mainly used to host Docker applications and in addition, using ECS, ECR, and Fargate alone is not scalable and not recommended for this type of scenarios.

Hosting the APIs in a static S3 web hosting bucket behind a CloudFront web distribution is not a suitable option as there is no compute capability for S3 and you can only use it as a static website. Although this solution is scalable since it is using CloudFront, the use of S3 to host the web APIs or the dynamic website is still incorrect.

The option that says: ***Use a Spot Fleet of Amazon EC2 instances, each with an Elastic Fabric Adapter (EFA) for more consistent latency and higher network throughput. Set up an Application Load Balancer to distribute traffic to the instances*** is incorrect because EC2 alone, without Auto Scaling, is not scalable. Even though you use Spot EC2 instance, it is still more expensive compared to Lambda because you will be charged only when your function is being used. An Elastic Fabric Adapter (EFA) is simply a network device that you can attach to your Amazon EC2 instance that enables you to achieve the application performance of an on-premises HPC cluster, with the scalability, flexibility, and elasticity provided by the AWS Cloud. Although EFA is scalable, the Spot Fleet configuration of this option doesn't have Auto Scaling involved.

References:

<https://docs.aws.amazon.com/apigateway/latest/developerguide/getting-started-with-lambda-integration.html>

<https://aws.amazon.com/lambda/pricing/>

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

EC2 Container Service (ECS) vs Lambda:

<https://tutorialsdojo.com/ec2-container-service-ecs-vs-lambda/>

2. QUESTION

Category: CSAA – Design Resilient Architectures

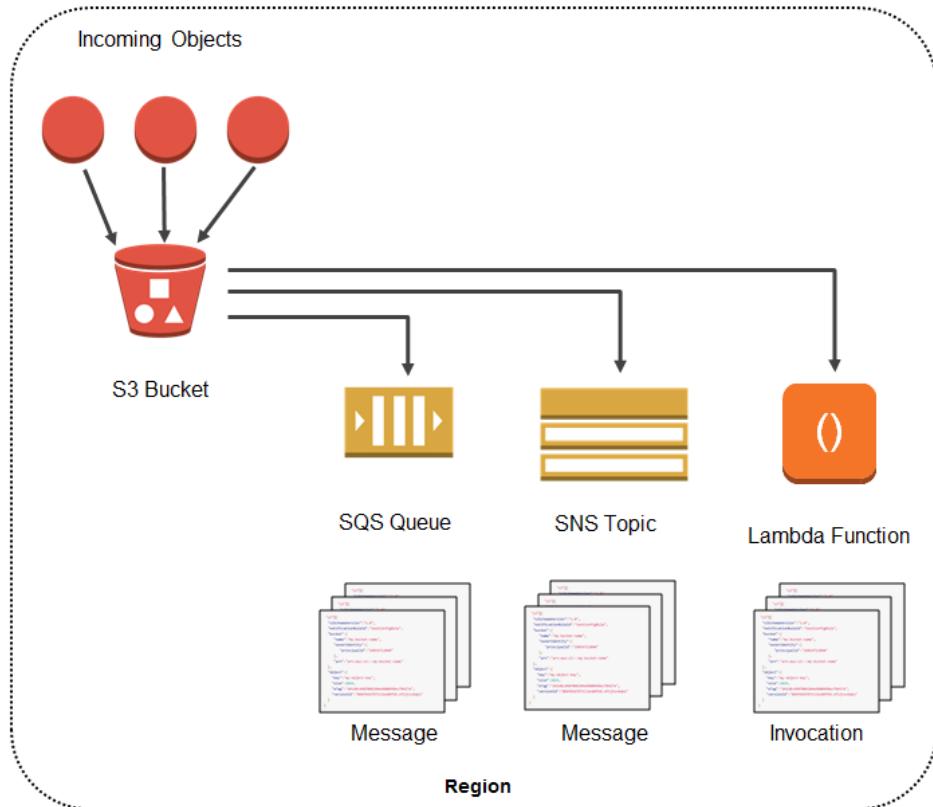
A Data Engineer is working for a litigation firm for their case history application. The engineer needs to keep track of all the cases that the firm has handled. The static assets like .jpg, .png, and .pdf files are stored in S3 for cost efficiency and high durability. As these files are critical to the business, the engineer wants to keep track of what's happening in the S3 bucket. The engineer found out that S3 has an event notification whenever a delete or write operation happens within the S3 bucket.

What are the possible Event Notification destinations available for S3 buckets? (Select TWO.)

- **Lambda function**
- SWF
- SES
- **SQS**
- Kinesis

Incorrect

The **Amazon S3 notification** feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration identifying the events you want Amazon S3 to publish, and the destinations where you want Amazon S3 to send the event notifications.



Amazon S3 supports the following destinations where it can publish events:

Amazon Simple Notification Service (Amazon SNS) topic – A web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients.

Amazon Simple Queue Service (Amazon SQS) queue – Offers reliable and scalable hosted queues for storing messages as they travel between computer.

AWS Lambda – AWS Lambda is a compute service where you can upload your code and the service can run the code on your behalf using the AWS infrastructure. You package up and upload your custom code to AWS Lambda when you create a Lambda function

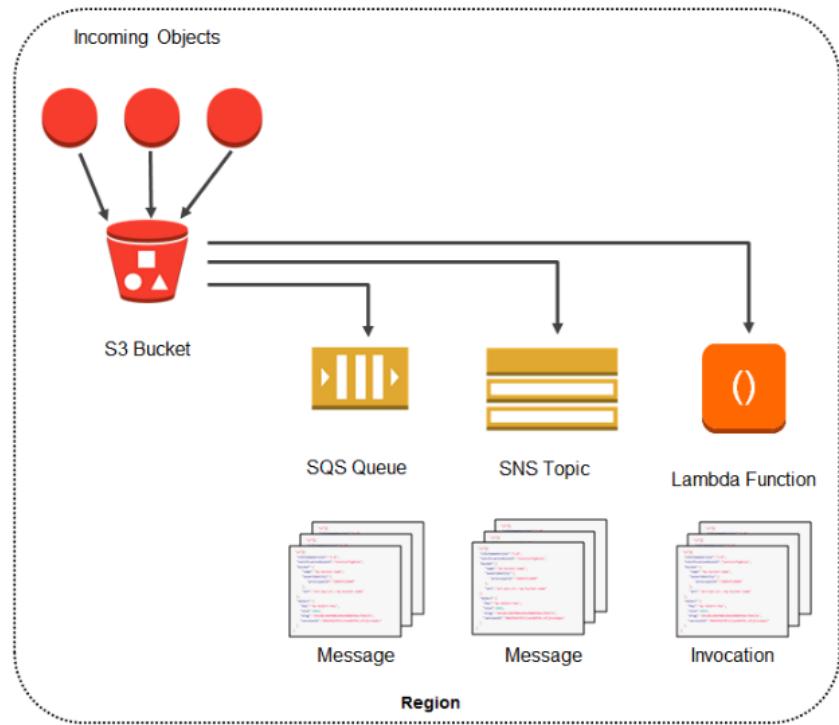
Kinesis is incorrect because this is used to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information, and not used for event notifications. You have to use SNS, SQS or Lambda.

SES is incorrect because this is mainly used for sending emails designed to help digital marketers and application developers send marketing, notification, and transactional emails, and not for sending event notifications from S3. You have to use SNS, SQS or Lambda.

SWF is incorrect because this is mainly used to build applications that use Amazon's cloud to coordinate work across distributed components and not used as a way to trigger event notifications from S3. You have to use SNS, SQS or Lambda.

Here's what you need to do in order to start using this new feature with your application:

1. Create the queue, topic, or Lambda function (which I'll call the target for brevity) if necessary.
2. Grant S3 permission to publish to the target or invoke the Lambda function. For SNS or SQS, you do this by applying an appropriate policy to the topic or the queue. For Lambda, you must create and supply an IAM role, then associate it with the Lambda function.
3. Arrange for your application to be invoked in response to activity on the target. As you will see in a moment, you have several options here.
4. Set the bucket's Notification Configuration to point to the target.



Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

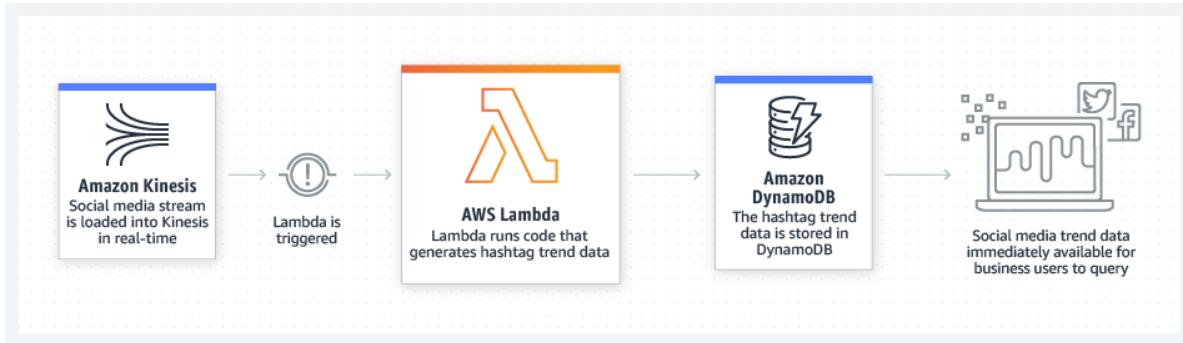
A company is building an internal application that serves as a repository for images uploaded by a couple of users. Whenever a user uploads an image, it would be sent to Kinesis Data Streams for processing before it is stored in an S3 bucket. If the upload was successful, the application will return a prompt informing the user that the operation was successful. The entire processing typically takes about 5 minutes to finish.

Which of the following options will allow you to asynchronously process the request to the application from upload request to Kinesis, S3, and return a reply in the most cost-effective manner?

- Use a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests.
- Use a combination of SQS to queue the requests and then asynchronously process them using On-Demand EC2 Instances.
- Use a combination of SNS to buffer the requests and then asynchronously process them using On-Demand EC2 Instances.
- Replace the Kinesis Data Streams with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.

Correct

AWS Lambda supports the synchronous and asynchronous invocation of a Lambda function. You can control the invocation type only when you invoke a Lambda function. When you use an AWS service as a trigger, the invocation type is predetermined for each service. You have no control over the invocation type that these event sources use when they invoke your Lambda function. Since processing only takes 5 minutes, Lambda is also a cost-effective choice.



You can use an AWS Lambda function to process messages in an Amazon Simple Queue Service (Amazon SQS) queue. Lambda event source mappings support standard queues and first-in, first-out (FIFO) queues. With Amazon SQS, you can offload tasks from one component of your application by sending them to a queue and processing them asynchronously.

Kinesis Data Streams is a real-time data streaming service that requires the provisioning of shards. Amazon SQS is a cheaper option because you only pay for what you use. Since there is no requirement for real-time processing in the scenario given, replacing Kinesis Data Streams with Amazon SQS would save more costs.

Hence, the correct answer is: ***Replace the Kinesis stream with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.***

Using a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests is incorrect. The AWS Step Functions service lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Although this can be a valid solution, it is not cost-effective since the application does not have a lot of components to orchestrate. Lambda functions can effectively meet the requirements in this scenario without using Step Functions. This service is not as cost-effective as Lambda.

Using a combination of SQS to queue the requests and then asynchronously processing them using On-Demand EC2 Instances and ***Using a combination of SNS to buffer the requests and then asynchronously processing them using On-Demand EC2 Instances*** are both incorrect as using On-Demand EC2 instances is not cost-effective. It is better to use a Lambda function instead.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-invocation.html>

<https://aws.amazon.com/blogs/compute/new-aws-lambda-controls-for-stream-processing-and-asynchronous-invocations/>

AWS Lambda Overview – Serverless Computing in AWS:

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A popular social media website uses a CloudFront web distribution to serve their static contents to their millions of users around the globe. They are receiving a number of complaints recently that their users take a lot of time to log into their website. There are also occasions when their users are getting HTTP 504 errors. You are instructed by your manager to significantly reduce the user's login time to further optimize the system.

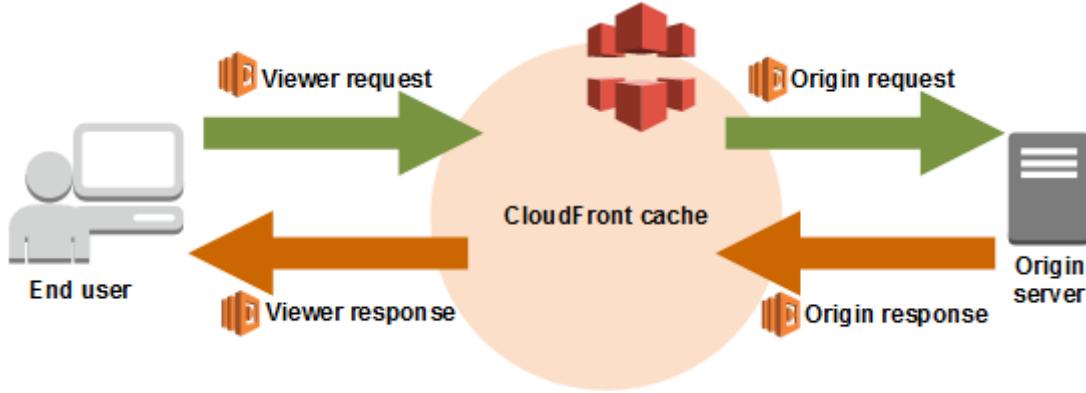
Which of the following options should you use together to set up a cost-effective solution that can improve your application's performance? (Select TWO.)

- Customize the content that the CloudFront web distribution delivers to your users using Lambda@Edge, which allows your Lambda functions to execute the authentication process in AWS locations closer to the users.
- Set up an origin failover by creating an origin group with two origins. Specify one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- Use multiple and geographically disperse VPCs to various AWS regions then create a transit VPC to connect all of your resources. In order to handle the requests faster, set up Lambda functions in each region using the AWS Serverless Application Model (SAM) service.
- Configure your origin to add a `Cache-Control max-age` directive to your objects, and specify the longest practical value for `max-age` to increase the cache hit ratio of your CloudFront distribution.
- Deploy your application to multiple AWS regions to accommodate your users around the world. Set up a Route 53 record with latency routing policy to route incoming traffic to the region that provides the best latency to the user.

Correct

Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests and responses at the following points:

- After CloudFront receives a request from a viewer (viewer request)
- Before CloudFront forwards the request to the origin (origin request)
- After CloudFront receives the response from the origin (origin response)
- Before CloudFront forwards the response to the viewer (viewer response)



In the given scenario, you can use Lambda@Edge to allow your Lambda functions to customize the content that CloudFront delivers and to execute the authentication process in AWS locations closer to the users. In addition, you can set up an origin failover by creating an origin group with two origins with one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin fails. This will alleviate the occasional HTTP 504 errors that users are experiencing. Therefore, the correct answers are:

- Customize the content that the CloudFront web distribution delivers to your users using Lambda@Edge, which allows your Lambda functions to execute the authentication process in AWS locations closer to the users.

- Set up an origin failover by creating an origin group with two origins. Specify one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.

The option that says: ***Use multiple and geographically disperse VPCs to various AWS regions then create a transit VPC to connect all of your resources. In order to handle the requests faster, set up Lambda functions in each region using the AWS Serverless Application Model (SAM) service*** is incorrect because of the same reason provided above. Although setting up multiple VPCs across various regions which are connected with a transit VPC is valid, this solution still entails higher setup and maintenance costs. A more cost-effective option would be to use Lambda@Edge instead.

The option that says: ***Configure your origin to add a Cache-Control max-age directive to your objects, and specify the longest practical value for max-age to increase the cache hit ratio of your CloudFront distribution*** is incorrect because improving the cache hit ratio for the CloudFront distribution is irrelevant in this scenario. You can improve your cache performance by increasing the proportion of your viewer requests that are served from CloudFront edge caches instead of going to your origin servers for content. However, take note that the problem in the scenario is the sluggish authentication process of your global users and not just the caching of the static objects.

The option that says: ***Deploy your application to multiple AWS regions to accommodate your users around the world. Set up a Route 53 record with latency routing policy to route incoming traffic to the region that provides the best latency to the user*** is incorrect because although this may resolve the performance issue, this solution entails a significant implementation cost since you have to deploy your application to multiple AWS regions. Remember that the scenario asks for a solution that will improve the performance of the application with **minimal cost**.

References:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/high_availability_origin_failover.html

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-edge.html>

Check out these Amazon CloudFront and AWS Lambda Cheat Sheets:

<https://tutorialsdojo.com/amazon-cloudfront/>

<https://tutorialsdojo.com/aws-lambda/>

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A software development company is using serverless computing with AWS Lambda to build and run applications without having to set up or manage servers. They have a Lambda function that connects to a MongoDB Atlas, which is a popular Database as a Service (DBaaS) platform and also uses a third party API to fetch certain data for their application. One of the developers was instructed to create the environment variables for the MongoDB database hostname, username, and password as well as the API credentials that will be used by the Lambda function for DEV, SIT, UAT, and PROD environments.

Considering that the Lambda function is storing sensitive database and API credentials, how can this information be secured to prevent other developers in the team, or anyone, from seeing these credentials in plain text? Select the best option that provides maximum security.

- There is no need to do anything because, by default, AWS Lambda already encrypts the environment variables using the AWS Key Management Service.
- AWS Lambda does not provide encryption for the environment variables. Deploy your code to an EC2 instance instead.
- Enable SSL encryption that leverages on AWS CloudHSM to store and encrypt the sensitive information.
- **Create a new KMS key and use it to enable encryption helpers that leverage on AWS Key Management Service to store and encrypt the sensitive information.**

Incorrect

When you create or update Lambda functions that use environment variables, AWS Lambda encrypts them using the AWS Key Management Service. When your Lambda function is invoked, those values are decrypted and made available to the Lambda code.

The first time you create or update Lambda functions that use environment variables in a region, a default service key is created for you automatically within AWS KMS. This key is used to encrypt environment variables. However, if you wish to use encryption helpers and use KMS to encrypt environment variables after your Lambda function is created, you must create your own AWS KMS key and choose it instead of the default key. The default key will give errors when chosen. Creating your own key gives you more flexibility, including the ability to create, rotate, disable, and define access controls, and to audit the encryption keys used to protect your data.

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. [Learn more](#)

password	AQICAHgdCwJ7eNzGOCBk9Q6nDD21wmtICsvWz2AsE75No	Encrypt	Code	Remove
Key	Value	Encrypt	Code	Remove

Encryption configuration

Enable helpers for encryption in transit [Info](#)

AWS KMS key to encrypt in transit
 [?key/2defc6c2-ab8a-499f-87de-](#)
⚠ AWS KMS call failed for reason: User: arn:aws:iam::84205 7:user/koko is not authorized to perform: kms:Encrypt on resource: arn:aws:kms:us-east-1:84205 2defc6c2-ab8a-499f-87de-

AWS KMS key to encrypt at rest [Info](#)
 Choose an AWS KMS key to encrypt the environment variables at rest, or simply let Lambda manage the encryption.
 (default) aws/lambda
 Use a customer master key

The option that says: ***There is no need to do anything because, by default, AWS Lambda already encrypts the environment variables using the AWS Key Management Service*** is incorrect. Although Lambda encrypts the environment variables in your function by default, the sensitive information would still be visible to other users who have access to the Lambda console. This is because Lambda uses a default KMS key to encrypt the variables, which is usually accessible by other users. The best option in this scenario is to use encryption helpers to secure your environment variables.

The option that says: ***Enable SSL encryption that leverages on AWS CloudHSM to store and encrypt the sensitive information*** is also incorrect since enabling SSL would encrypt data only when in-transit. Your other teams would still be able to view the plaintext at-rest. Use AWS KMS instead.

The option that says: ***AWS Lambda does not provide encryption for the environment variables. Deploy your code to an EC2 instance instead*** is incorrect since, as mentioned, Lambda does provide encryption functionality of environment variables.

References:

https://docs.aws.amazon.com/lambda/latest/dg/env_variables.html#env_encrypt

https://docs.aws.amazon.com/lambda/latest/dg/tutorial-env_console.html

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

AWS Lambda Overview – Serverless Computing in AWS: <https://youtu.be/bPVX1zHwAnY>

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A company needs to implement a solution that will process real-time streaming data of its users across the globe. This will enable them to track and analyze globally-distributed user activity on their website and mobile applications, including clickstream analysis. The solution should process the data in close geographical proximity to their users and respond to user requests at low latencies.

Which of the following is the most suitable solution for this scenario?

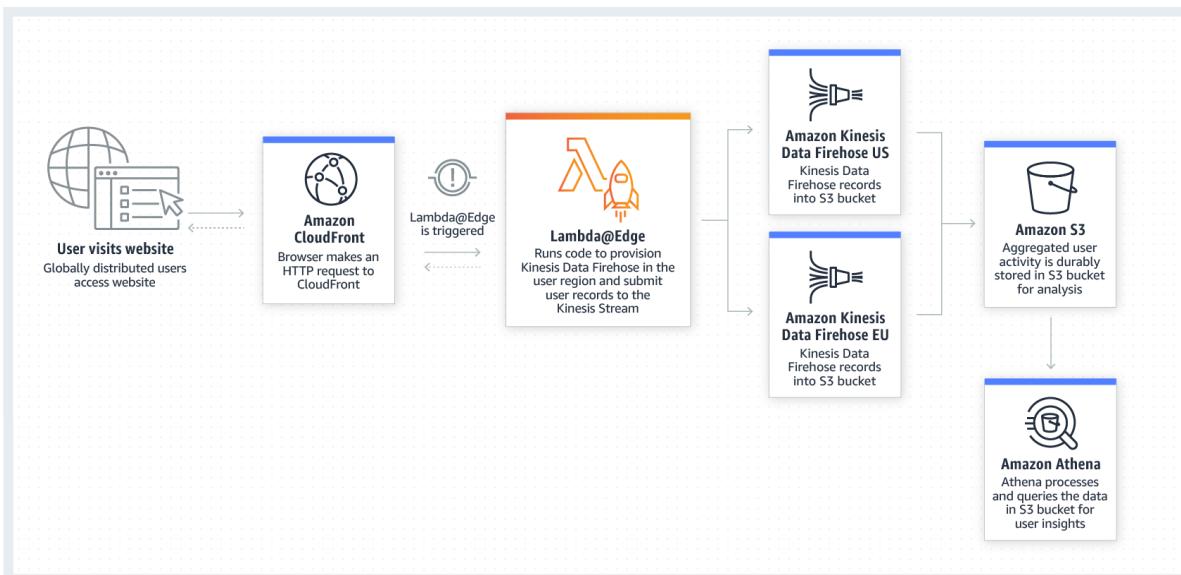
- Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.
- Use a CloudFront web distribution and Route 53 with a Geoproximity routing policy in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.

- Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Amazon Athena and durably store the results to an Amazon S3 bucket.
- Use a CloudFront web distribution and Route 53 with a latency-based routing policy, in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.

Correct

Lambda@Edge is a feature of Amazon CloudFront that lets you run code closer to users of your application, which improves performance and reduces latency. With Lambda@Edge, you don't have to provision or manage infrastructure in multiple locations around the world. You pay only for the compute time you consume – there is no charge when your code is not running.

With Lambda@Edge, you can enrich your web applications by making them globally distributed and improving their performance — all with zero server administration. Lambda@Edge runs your code in response to events generated by the Amazon CloudFront content delivery network (CDN). Just upload your code to AWS Lambda, which takes care of everything required to run and scale your code with high availability at an AWS location closest to your end user.



By using Lambda@Edge and Kinesis together, you can process real-time streaming data so that you can track and analyze globally-distributed user activity on your website and mobile applications, including clickstream analysis. Hence, the correct answer in this scenario is the option that says: ***Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.***

The options that say: ***Use a CloudFront web distribution and Route 53 with a latency-based routing policy, in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket*** and ***Use a CloudFront web distribution and Route 53 with a Geoproximity routing policy in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket*** are both incorrect because you can only route traffic using Route 53 since it does not have any computing capability. This solution would not be able to process and return the data in close geographical proximity to your users since it is not using Lambda@Edge.

The option that says: ***Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Amazon Athena and durably store the results to an Amazon S3 bucket*** is incorrect because although using Lambda@Edge is correct, Amazon Athena is just an interactive query service that

enables you to easily analyze data in Amazon S3 using standard SQL. Kinesis should be used to process the streaming data in real-time.

References:

<https://aws.amazon.com/lambda/edge/>

<https://aws.amazon.com/blogs/networking-and-content-delivery/global-data-ingestion-with-amazon-cloudfront-and-lambdaedge/>

7. QUESTION

Category: CSAA – Design Resilient Architectures

A company is using Amazon VPC that has a CIDR block of `10.31.0.0/27` that is connected to the on-premises data center. There was a requirement to create a Lambda function that will process massive amounts of cryptocurrency transactions every minute and then store the results to EFS. After setting up the serverless architecture and connecting the Lambda function to the VPC, the Solutions Architect noticed an increase in invocation errors with EC2 error types such as `EC2ThrottledException` at certain times of the day.

Which of the following are the possible causes of this issue? (Select TWO.)

- The associated security group of your function does not allow outbound connections.
- The attached IAM execution role of your function does not have the necessary permissions to access the resources of your VPC.
- Your VPC does not have sufficient subnet ENIs or subnet IPs.
- Your VPC does not have a NAT gateway.
- You only specified one subnet in your Lambda function configuration. That single subnet runs out of available IP addresses and there is no other subnet or Availability Zone which can handle the peak load.

Correct

You can configure a function to connect to a virtual private cloud (VPC) in your account. Use Amazon Virtual Private Cloud (Amazon VPC) to create a private network for resources such as databases, cache instances, or internal services. Connect your function to the VPC to access private resources during execution.

AWS Lambda runs your function code securely within a VPC by default. However, to enable your Lambda function to access resources inside your private VPC, you must provide additional VPC-specific configuration information that includes VPC subnet IDs and security group IDs. AWS Lambda uses this information to set up elastic network interfaces (ENIs) that enable your function to connect securely to other resources within your private VPC.

Lambda functions cannot connect directly to a VPC with dedicated instance tenancy. To connect to resources in a dedicated VPC, peer it to a second VPC with default tenancy.

Your Lambda function automatically scales based on the number of events it processes. If your Lambda function accesses a VPC, you must make sure that your VPC has sufficient ENI capacity to support the scale requirements of your Lambda function. It is also recommended that you specify at least one subnet in each Availability Zone in your Lambda function configuration.

By specifying subnets in each of the Availability Zones, your Lambda function can run in another Availability Zone if one goes down or runs out of IP addresses. If your VPC does not have sufficient ENIs or subnet IPs, your Lambda function will not scale as requests increase, and you will see an increase in invocation errors with EC2 error types like `EC2ThrottledException`. For asynchronous invocation, if you see an increase in errors without corresponding CloudWatch Logs, invoke the Lambda function synchronously in the console to get the error responses.

Hence, the correct answers for this scenario are:

- You only specified one subnet in your Lambda function configuration. That single subnet runs out of available IP addresses and there is no other subnet or Availability Zone which can handle the peak load.

- Your VPC does not have sufficient subnet ENIs or subnet IPs.

The screenshot shows the AWS Lambda function configuration interface. It consists of three main sections:

- Execution role**: A dropdown menu titled "Use an existing role" is open, showing the selected option "service-role/tutorialsdojo-lambda-vpc-role-xd5u9vhy". Below the dropdown, there is a link to "View the tutorialsdojo-lambda-vpc-role-xd5u9vhy role on the IAM console." There is also a small "C" icon.
- Network**: A dropdown menu titled "No VPC" is selected. Above it, there is a "Virtual Private Cloud (VPC) Info" section with a link to "Choose a VPC for your function to access." A cursor arrow points towards this link.
- Concurrency**: Shows "Unreserved account concurrency 1000". It has two radio button options: "Use unreserved account concurrency" (selected) and "Reserve concurrency". A slider bar is visible next to the reserve concurrency option.

The option that says: ***Your VPC does not have a NAT gateway*** is incorrect because an issue in the NAT Gateway is unlikely to cause a request throttling issue or produce an `EC2ThrottledException` error in Lambda. As per the scenario, the issue is happening only at certain times of the day, which means that the issue is only intermittent and the function works at other times. We can also conclude that an availability issue is not an issue since the application is already using a highly available NAT Gateway and not just a NAT instance.

The option that says: ***The associated security group of your function does not allow outbound connections*** is incorrect because if the associated security group does not allow outbound connections then the Lambda function will not work at all in the first place. Remember that as per the scenario, the issue only happens intermittently. In addition, Internet traffic restrictions do not usually produce `EC2ThrottledException` errors.

The option that says: ***The attached IAM execution role of your function does not have the necessary permissions to access the resources of your VPC*** is incorrect because just as what is explained above, the issue is intermittent and thus, the IAM execution role of the function does have the necessary permissions to access the resources of the VPC since it works at those specific times. In case the issue is indeed caused by a permission problem then an `EC2AccessDeniedException` the error would most likely be returned and not an `EC2ThrottledException` error.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/vpc.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/internet-access-lambda-function/>

<https://aws.amazon.com/premiumsupport/knowledge-center/lambda-troubleshoot-invoke-error-502-500/>

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

8. QUESTION

Category: CSAA – Design Resilient Architectures

An application is using a Lambda function to process complex financial data that run for 15 minutes on average. Most invocations were successfully processed. However, you noticed that there are a few terminated invocations throughout the day, which caused data discrepancy in the application.

Which of the following is the most likely cause of this issue?

- The failed Lambda Invocations contain a `ServiceException` error which means that the AWS Lambda service encountered an internal error.
- The concurrent execution limit has been reached.
- The Lambda function contains a recursive code and has been running for over 15 minutes.
- The failed Lambda functions have been running for over 15 minutes and reached the maximum execution time.**

Correct

A **Lambda function** consists of code and any associated dependencies. In addition, a Lambda function also has configuration information associated with it. Initially, you specify the configuration information when you create a Lambda function. Lambda provides an API for you to update some of the configuration data.

You pay for the AWS resources that are used to run your Lambda function. To prevent your Lambda function from running indefinitely, you specify a **timeout**. When the specified timeout is reached, AWS Lambda terminates execution of your Lambda function. It is recommended that you set this value based on your expected execution time. The default timeout is 3 seconds and the maximum execution duration per request in AWS Lambda is 900 seconds, which is equivalent to 15 minutes.

Hence, the correct answer is the option that says: ***The failed Lambda functions have been running for over 15 minutes and reached the maximum execution time***.

The screenshot shows the AWS Lambda 'Basic settings' configuration page. At the top, there are tabs for 'Throttle', 'Qualifiers ▾', 'Actions ▾', and 'Select a t'. Below these are sections for 'Description' (with an empty text area) and 'Memory (MB) Info' (with a slider set to 128 MB). A green box highlights the 'Timeout' section, which displays '15 min 0 sec'. The entire configuration page is enclosed in a light gray border.

Take note that you can invoke a Lambda function synchronously either by calling the `Invoke` operation or by using an AWS SDK in your preferred runtime. If you anticipate a long-running Lambda function, your client may time out before function execution completes. To avoid this, update the client timeout or your SDK configuration.

The option that says: ***The concurrent execution limit has been reached*** is incorrect because, by default, the AWS Lambda limits the total concurrent executions across all functions within a given region to 1000. By setting a concurrency limit on a function, Lambda guarantees that allocation will be applied specifically to that function, regardless of the amount of traffic processing the remaining functions. If that limit is exceeded, the function will be throttled but not terminated, which is in contrast with what is happening in the scenario.

The option that says: ***The Lambda function contains a recursive code and has been running for over 15 minutes*** is incorrect because having a recursive code in your Lambda function does not directly result to an abrupt termination of the function execution. This is a scenario wherein the function automatically calls itself until some arbitrary criteria is met. This could lead to an unintended volume of function invocations and escalated costs, but not an abrupt termination because Lambda will throttle all invocations to the function.

The option that says: ***The failed Lambda Invocations contain a ServiceException error which means that the AWS Lambda service encountered an internal error*** is incorrect because although this is a valid root cause, it is unlikely to have several **ServiceException** errors throughout the day unless there is an outage or disruption in AWS. Since the scenario says that the Lambda function runs for about 10 to 15 minutes, the maximum execution duration is the most likely cause of the issue and not the AWS Lambda service encountering an internal error.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/limits.html>

<https://docs.aws.amazon.com/lambda/latest/dg/resource-model.html>

AWS Lambda Overview – Serverless Computing in AWS:

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application is using CloudFront to distribute their images, videos, and other static contents stored in their S3 bucket to its users around the world. The company has recently introduced a new member-only access to some of its high quality media files. There is a requirement to provide access to multiple private media files only to their paying subscribers without having to change their current URLs.

Which of the following is the most suitable solution that you should implement to satisfy this requirement?

- Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member.
- Create a Signed URL with a custom policy which only allows the members to see the private files.
- Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members.
- Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required `Set-Cookie` headers to the viewer which will unlock the content only to them.

Incorrect

CloudFront signed URLs and signed cookies provide the same basic functionality: they allow you to control who can access your content. If you want to serve private content through CloudFront and you're trying to decide whether to use signed URLs or signed cookies, consider the following:

Use **signed URLs** for the following cases:

- You want to use an RTMP distribution. Signed cookies aren't supported for RTMP distributions.
- You want to restrict access to individual files, for example, an installation download for your application.
- Your users are using a client (for example, a custom HTTP client) that doesn't support cookies.

Use **signed cookies** for the following cases:

- You want to provide access to multiple restricted files, for example, all of the files for a video in HLS format or all of the files in the subscribers' area of a website.
- You don't want to change your current URLs.

Hence, the correct answer for this scenario is the option that says: ***Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required Set-Cookie headers to the viewer which will unlock the content only to them.***

The option that says: ***Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member*** is incorrect because a Match Viewer is an Origin Protocol Policy which configures CloudFront to communicate with your origin using HTTP or HTTPS, depending on the protocol of the viewer request. CloudFront caches the object only once even if viewers make requests using both HTTP and HTTPS protocols.

The option that says: ***Create a Signed URL with a custom policy which only allows the members to see the private files*** is incorrect because Signed URLs are primarily used for providing access to individual files, as shown on the above explanation. In addition, the scenario explicitly says that they don't want to change their current URLs which is why implementing Signed Cookies is more suitable than Signed URL.

The option that says: ***Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members*** is incorrect because Field-Level Encryption only allows you to securely upload user-submitted sensitive information to your web servers. It does not provide access to download multiple private files.

Reference:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-choosing-signed-urls-cookies.html>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-signed-cookies.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A travel photo sharing website is using Amazon S3 to serve high-quality photos to visitors of your website. After a few days, you found out that there are other travel websites linking and using your photos. This resulted in financial losses for your business.

What is the MOST effective method to mitigate this issue?

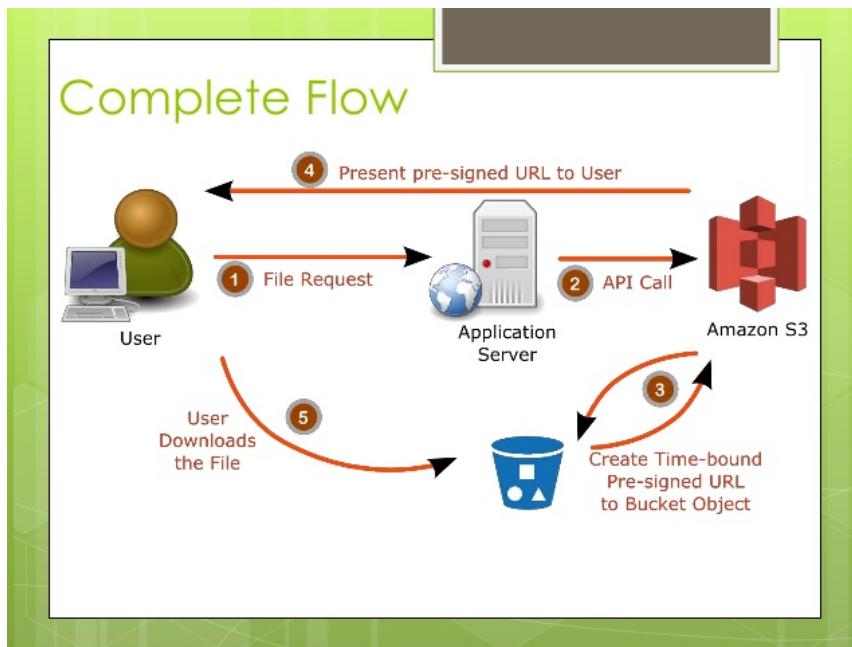
- Store and privately serve the high-quality photos on Amazon WorkDocs instead.
- **Configure your S3 bucket to remove public read access and use pre-signed URLs with expiry dates.**
- Use CloudFront distributions for your photos.
- Block the IP addresses of the offending websites using NACL.

Incorrect

In Amazon S3, all objects are private by default. Only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a pre-signed URL, using their own security credentials, to grant time-limited permission to download the objects.

When you create a pre-signed URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The pre-signed URLs are valid only for the specified duration.

Anyone who receives the pre-signed URL can then access the object. For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a pre-signed URL.



Using CloudFront distributions for your photos is incorrect. CloudFront is a content delivery network service that speeds up delivery of content to your customers.

Blocking the IP addresses of the offending websites using NACL is also incorrect. Blocking IP address using NACLs is not a very efficient method because a quick change in IP address would easily bypass this configuration.

Storing and privately serving the high-quality photos on Amazon WorkDocs instead is incorrect as WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service. It is not a suitable service for storing static content. Amazon WorkDocs is more often used to easily create, edit, and share documents for collaboration and not for serving object data like Amazon S3.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectOperations.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

S3 Pre-signed URLs vs CloudFront Signed URLs vs Origin Access Identity (OAI)

<https://tutorialsdojo.com/s3-pre-signed-urls-vs-cloudfront-signed-urls-vs-origin-access-identity-oai/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A website that consists of HTML, CSS, and other client-side Javascript will be hosted on the AWS environment. Several high-resolution images will be displayed on the webpage. The website and the photos should have the optimal loading response times as possible, and should also be able to scale to high request rates.

Which of the following architectures can provide the most cost-effective and fastest loading experience?

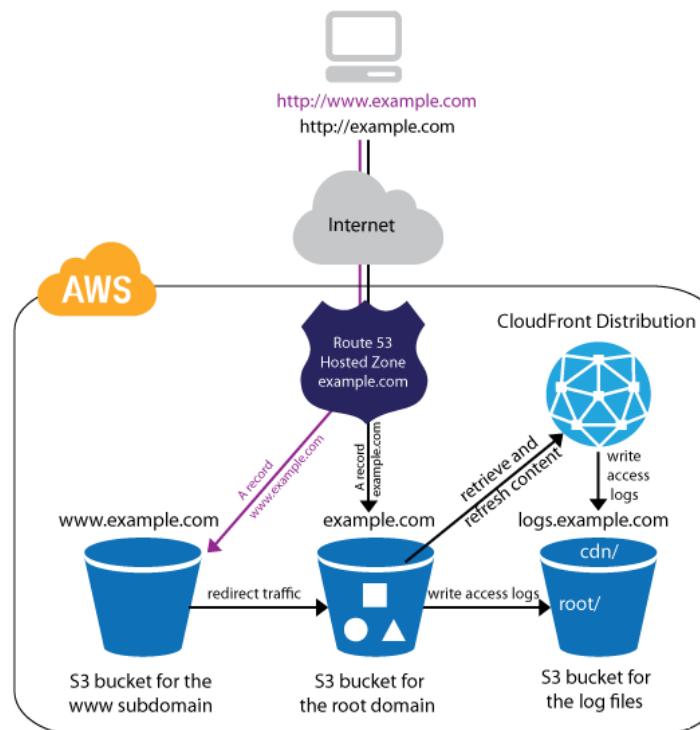
- Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator.

- Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.
- Create a Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.
- Create a Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.

Correct

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Additionally, You can use Amazon S3 to host a static website. On a static website, individual webpages include static content. Amazon S3 is **highly scalable and you only pay for what you use**, you can start small and grow your application as you wish, with no compromise on performance or reliability.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. CloudFront can be integrated with Amazon S3 for fast delivery of data originating from an S3 bucket to your end-users. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.



The scenario given is about storing and hosting images and a static website respectively. Since we are just dealing with static content, we can leverage the web hosting feature of S3. Then we can improve the architecture further by integrating it with CloudFront. This way, users will be able to load both the web pages and images faster than if we are serving them from a standard webserver.

Hence, the correct answer is: ***Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.***

The option that says: ***Create an Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in a S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users*** is incorrect. Creating your own web server just to host a static website in AWS is a costly solution. Web Servers on an EC2 instance is usually used for hosting

dynamic web applications. Since static websites contain web pages with fixed content, we should use S3 website hosting instead.

The option that says: ***Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator*** is incorrect. This is how we serve static websites in the old days. Now, with the help of S3 website hosting, we can host our static contents from a durable, high-availability, and highly scalable environment without managing any servers. Hosting static websites in S3 is cheaper than hosting it in an EC2 instance. In addition, Using ASG for scaling instances that host a static website is an over-engineered solution that carries unnecessary costs. S3 automatically scales to high requests and you only pay for what you use.

The option that says: ***Create an Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users*** is incorrect because although LightSail is cheaper than EC2, creating your own LightSail web server for hosting static websites is still a relatively expensive solution when compared to hosting it on S3. In addition, S3 automatically scales to high request rates.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>

<https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/>

Check out these Amazon S3 and CloudFront Cheat Sheets:

<https://tutorialsdojo.com/amazon-s3/>

<https://tutorialsdojo.com/amazon-cloudfront/>

4. QUESTION

Category: CSAA – Design High-Performing Architectures

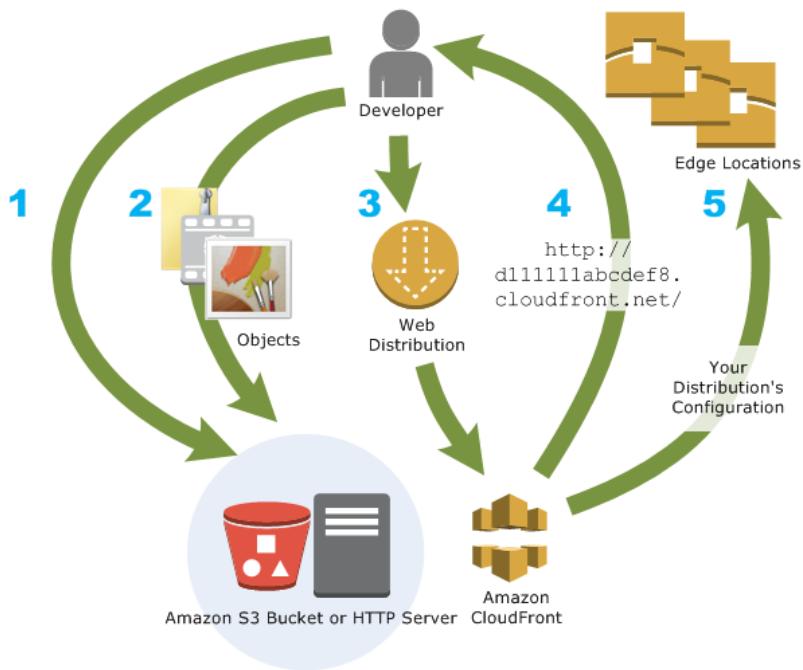
A company has a global news website hosted in a fleet of EC2 Instances. Lately, the load on the website has increased which resulted in slower response time for the site visitors. This issue impacts the revenue of the company as some readers tend to leave the site if it does not load after 10 seconds.

Which of the below services in AWS can be used to solve this problem? (Select TWO.)

- Deploy the website to all regions in different VPCs for faster processing.
- Use Amazon ElastiCache for the website's in-memory data store or cache.
- For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions.
- Use Amazon CloudFront with website as the custom origin.

Correct

The global news website has a problem with latency considering that there are a lot of readers of the site from all parts of the globe. In this scenario, you can use a content delivery network (CDN) which is a geographically distributed group of servers that work together to provide fast delivery of Internet content. And since this is a news website, most of its data are read-only, which can be cached to improve the read throughput and avoid repetitive requests from the server.



In AWS, Amazon CloudFront is the global content delivery network (CDN) service that you can use and for web caching, Amazon ElastiCache is the suitable service.

Hence, the correct answers are:

- Use Amazon CloudFront with website as the custom origin.

- Use Amazon ElastiCache for the website's in-memory data store or cache.

The option that says: ***For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions*** is incorrect as AWS Storage Gateway is used for storage.

Deploying the website to all regions in different VPCs for faster processing is incorrect as this would be costly and totally unnecessary considering that you can use Amazon CloudFront and ElastiCache to improve the performance of the website.

References:

<https://aws.amazon.com/elasticsearch/>

<http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application, which is used by your clients around the world, is hosted in an Auto Scaling group of EC2 instances behind a Classic Load Balancer. You need to secure your application by allowing multiple domains to serve SSL traffic over the same IP address.

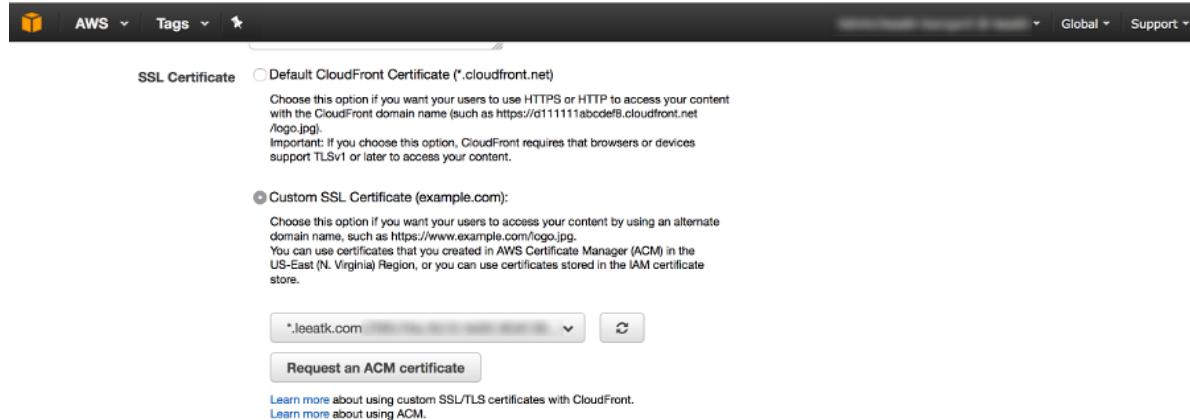
Which of the following should you do to meet the above requirement?

- Generate an SSL certificate with AWS Certificate Manager and create a CloudFront web distribution. Associate the certificate with your web distribution and enable the support for Server Name Indication (SNI).
- It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS.
- Use Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic.
- Use an Elastic IP and upload multiple 3rd party certificates in your Classic Load Balancer using the AWS Certificate Manager.

Incorrect

SNI Custom SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname which the viewers are trying to connect to.

Amazon CloudFront delivers your content from each edge location and offers the same security as the Dedicated IP Custom SSL feature. SNI Custom SSL works with most modern browsers, including Chrome version 6 and later (running on Windows XP and later or OS X 10.5.7 and later), Safari version 3 and later (running on Windows Vista and later or Mac OS X 10.5.6. and later), Firefox 2.0 and later, and Internet Explorer 7 and later (running on Windows Vista and later).



The screenshot shows the AWS CloudFront SSL Certificate configuration page. Under 'SSL Certificate', the 'Custom SSL Certificate (example.com)' option is selected. A dropdown menu shows '*.leettk.com' as the alternate domain name. Below it is a button to 'Request an ACM certificate'.

Custom SSL Client Support

- Only Clients that Support Server Name Indication (SNI)
CloudFront serves your content over HTTPS only to clients that support SNI. Older browsers and other clients that do not support SNI can not access your content over HTTPS.
[Learn More](#)
- All Clients (\$600/month prorated charge applies). [Learn about pricing.](#)
CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over HTTPS. Any client can access your content.
[Learn More](#)

Some users may not be able to access your content because some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content. If you need to support non-SNI compliant browsers for HTTPS content, it is recommended to use the Dedicated IP Custom SSL feature.

Using Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic is incorrect because a Classic Load Balancer does not support Server Name Indication (SNI). You have to use an Application Load Balancer instead or a CloudFront web distribution to allow the SNI feature.

Using an Elastic IP and uploading multiple 3rd party certificates in your Classic Load Balancer using the AWS Certificate Manager is incorrect because just like in the above, a Classic Load Balancer does not support Server Name Indication (SNI) and the use of an Elastic IP is not a suitable solution to allow multiple domains to serve SSL traffic. You have to use Server Name Indication (SNI).

The option that says: ***It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS*** is incorrect because AWS does support the use of Server Name Indication (SNI).

References:

<https://aws.amazon.com/about-aws/whats-new/2014/03/05/amazon-cloudfront-announces-sni-custom-ssl/>

<https://aws.amazon.com/blogs/security/how-to-help-achieve-mobile-app-transport-security-compliance-by-using-amazon-cloudfront-and-aws-certificate-manager/>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

SNI Custom SSL vs Dedicated IP Custom SSL:

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A global news network created a CloudFront distribution for their web application. However, you noticed that the application's origin server is being hit for each request instead of the AWS Edge locations, which serve the cached objects. The issue occurs even for the commonly requested objects.

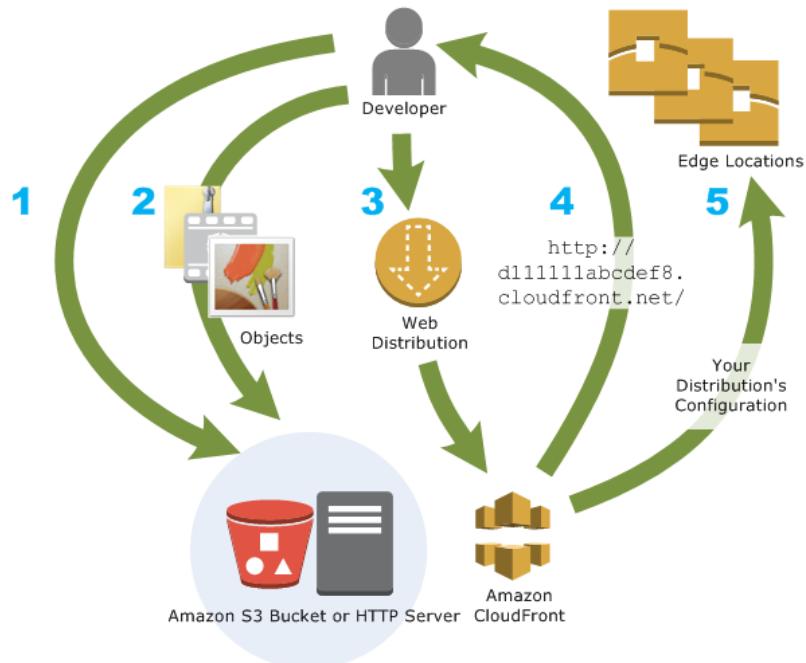
What could be a possible cause of this issue?

- The file sizes of the cached objects are too large for CloudFront to handle.
- You did not add an SSL certificate.
- An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server.
- The Cache-Control max-age directive is set to zero.**

Correct

You can control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin. Reducing the duration allows you to serve dynamic content. Increasing the duration means your users get better performance because your objects are more likely to be served directly from the edge cache. A longer duration also reduces the load on your origin.

Typically, CloudFront serves an object from an edge location until the cache duration that you specified passes — that is, until the object expires. After it expires, the next time the edge location gets a user request for the object, CloudFront forwards the request to the origin server to verify that the cache contains the latest version of the object.



The `Cache-Control` and `Expires` headers control how long objects stay in the cache. The `Cache-Control max-age` directive lets you specify how long (in seconds) you want an object to remain in the cache before CloudFront gets the object again from the origin server. The minimum expiration time CloudFront supports is 0 seconds for web distributions and 3600 seconds for RTMP distributions.

In this scenario, the main culprit is that the Cache-Control max-age directive is set to a low value, which is why the request is always directed to your origin server.

Hence, the correct answer is: ***The Cache-Control max-age directive is set to zero.***

The option that says: ***An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server*** is incorrect because the issue also occurs even for the commonly requested objects. This means that these objects were successfully requested before but due to a zero Cache-Control max-age directive value, it causes this issue in CloudFront.

The options that say: ***The file sizes of the cached objects are too large for CloudFront to handle*** and ***You did not add an SSL certificate*** are incorrect because they are not related to the issue in caching.

Reference:

<http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Expiration.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

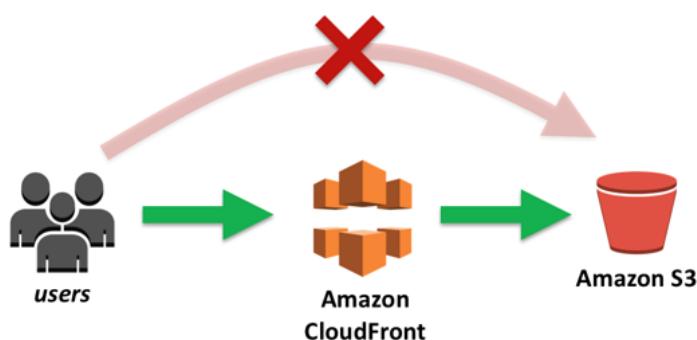
A Solutions Architect is working for a large global media company with multiple office locations all around the world. The Architect is instructed to build a system to distribute training videos to all employees.

Using CloudFront, what method would be used to serve content that is stored in S3, but not publicly accessible from S3 directly?

- Create an Identity and Access Management (IAM) user for CloudFront and grant access to the objects in your S3 bucket to that IAM user.
- Add the CloudFront account security group.
- Create an S3 bucket policy that lists the CloudFront distribution ID as the principal and the target bucket as the Amazon Resource Name (ARN).
- **Create an Origin Access Identity (OAI) for CloudFront and grant access to the objects in your S3 bucket to that OAI.**

Correct

When you create or update a distribution in CloudFront, you can add an origin access identity (OAI) and automatically update the bucket policy to give the origin access identity permission to access your bucket. Alternatively, you can choose to manually change the bucket policy or change ACLs, which control permissions on individual objects in your bucket.



You can update the Amazon S3 bucket policy using either the AWS Management Console or the Amazon S3 API:

- Grant the CloudFront origin access identity the applicable permissions on the bucket.
- Deny access to anyone that you don't want to have access using Amazon S3 URLs.

Reference:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-restricting-access-to-s3.html#private-content-granting-permissions-to-oai>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

S3 Pre-signed URLs vs CloudFront Signed URLs vs Origin Access Identity (OAI)

<https://tutorialsdojo.com/s3-pre-signed-urls-vs-cloudfront-signed-urls-vs-origin-access-identity-oai/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

1. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A Solutions Architect is working for a financial company. The manager wants to have the ability to automatically transfer obsolete data from their S3 bucket to a low-cost storage system in AWS.

What is the best solution that the Architect can provide to them?

- Use Amazon SQS.
- Use CloudEndure Migration.
- Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier.
- **Use Lifecycle Policies in S3 to move obsolete data to Glacier.**

Correct

In this scenario, you can use lifecycle policies in S3 to automatically move obsolete data to Glacier.

Lifecycle configuration in Amazon S3 enables you to specify the lifecycle management of objects in a bucket. The configuration is a set of one or more rules, where each rule defines an action for Amazon S3 to apply to a group of objects.

The screenshot shows the AWS S3 console with the 'Lifecycle rule actions' configuration. On the left, there's a sidebar with 'Amazon S3' selected. The main area has a heading 'Lifecycle rule actions' with a note about fees and links to 'Learn more' and 'Amazon S3 pricing'. It lists several actions:

- Transition *current* versions of objects between storage classes
- Transition *previous* versions of objects between storage classes
- Expire *current* versions of objects
- Permanently delete *previous* versions of objects
- Delete expired delete markers or incomplete multipart uploads

A note states: 'When a lifecycle rule is scoped with tags, these actions are unavailable.' Below this is a section titled 'Transition current versions of objects between storage classes' with fields for 'Storage class transitions' (set to 'Glacier') and 'Days after object creation' (set to '30'). Buttons for 'Add transition' and 'Remove transition' are present. A 'Tutorials Dojo' link is at the bottom right.

These actions can be classified as follows:

Transition actions – In which you define when objects transition to another storage class. For example, you may choose to transition objects to the STANDARD_IA (IA, for infrequent access) storage class 30 days after creation, or archive objects to the GLACIER storage class one year after creation.

Expiration actions – In which you specify when the objects expire. Then Amazon S3 deletes the expired objects on your behalf.

The option that says: ***Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier*** is incorrect because you don't need to create a scheduled job in EC2 as you can simply use the lifecycle policy in S3.

The option that says: ***Use Amazon SQS*** is incorrect as SQS is not a storage service. Amazon SQS is primarily used to decouple your applications by queueing the incoming requests of your application.

The option that says: ***Use CloudEndure Migration*** is incorrect because this service is just a highly automated lift-and-shift (rehost) solution that simplifies, expedites, and reduces the cost of migrating applications to AWS. You cannot use this to automatically transition your S3 objects to a cheaper storage class.

References:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

<https://aws.amazon.com/blogs/aws/archive-s3-to-glacier/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

2. QUESTION

Category: CSAA – Design Resilient Architectures

There was an incident in your production environment where the user data stored in the S3 bucket has been accidentally deleted by one of the Junior DevOps Engineers. The issue was escalated to your manager and after a few days, you were instructed to improve the security and protection of your AWS resources.

What combination of the following options will protect the S3 objects in your bucket from both accidental deletion and overwriting? (Select TWO.)

- Provide access to S3 data strictly through pre-signed URL only
- Disallow S3 Delete using an IAM bucket policy
- **Enable Versioning**
- **Enable Multi-Factor Authentication Delete**
- Enable Amazon S3 Intelligent-Tiering

Correct

By using Versioning and enabling MFA (Multi-Factor Authentication) Delete, you can secure and recover your S3 objects from accidental deletion or overwrite.

Versioning is a means of keeping multiple variants of an object in the same bucket. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

You can also optionally add another layer of security by configuring a bucket to enable MFA (Multi-Factor Authentication) Delete, which requires additional authentication for either of the following operations:

- Change the versioning state of your bucket
- Permanently delete an object version

MFA Delete requires two forms of authentication together:

- Your security credentials
- The concatenation of a valid serial number, a space, and the six-digit code displayed on an approved authentication device

Providing access to S3 data strictly through pre-signed URL only is incorrect since a pre-signed URL gives access to the object identified in the URL. Pre-signed URLs are useful when customers perform an object upload to your S3 bucket, but does not help in preventing accidental deletes.

Disallowing S3 Delete using an IAM bucket policy is incorrect since you still want users to be able to delete objects in the bucket, and you just want to prevent accidental deletions. Disallowing S3 Delete using an IAM bucket policy will restrict all delete operations to your bucket.

Enabling Amazon S3 Intelligent-Tiering is incorrect since S3 intelligent tiering does not help in this situation.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A start-up company that offers an intuitive financial data analytics service has consulted you about their AWS architecture. They have a fleet of Amazon EC2 worker instances that process financial data and then outputs reports which are used by their clients. You must store the generated report files in a durable storage. The number of files to be stored can grow over time as the start-up company is expanding rapidly overseas and hence, they also need a way to distribute the reports faster to clients located across the globe.

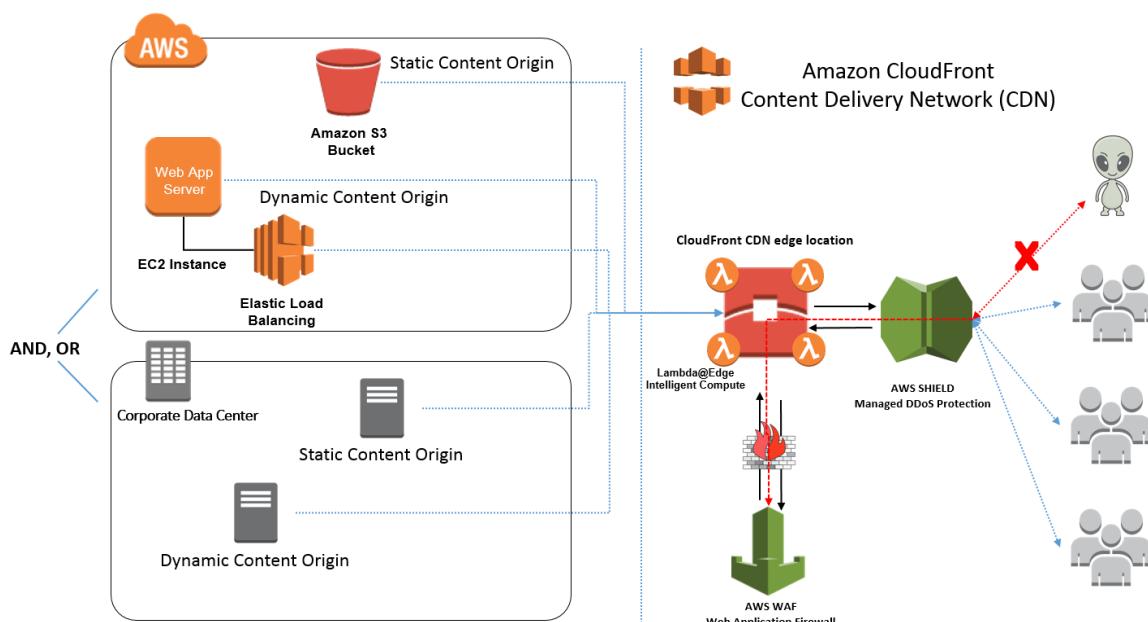
Which of the following is a cost-efficient and scalable storage option that you should use for this scenario?

- Use multiple EC2 instance stores for data storage and ElastiCache as the CDN.
- Use Amazon S3 Glacier as the data storage and ElastiCache as the CDN.
- Use Amazon Redshift as the data storage and CloudFront as the CDN.
- **Use Amazon S3 as the data storage and CloudFront as the CDN.**

Correct

A Content Delivery Network (CDN) is a critical component of nearly any modern web application. It used to be that CDN merely improved the delivery of content by replicating commonly requested files (static content) across a globally distributed set of caching servers. However, CDNs have become much more useful over time.

For caching, a CDN will reduce the load on an application origin and improve the experience of the requestor by delivering a local copy of the content from a nearby cache edge, or Point of Presence (PoP). The application origin is off the hook for opening the connection and delivering the content directly as the CDN takes care of the heavy lifting. The end result is that the application origins don't need to scale to meet demands for static content.



Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.

Amazon S3 offers a highly durable, scalable, and secure destination for backing up and archiving your critical data. This is the correct option as the start-up company is looking for a durable storage to store the audio and text files. In addition, ElastiCache is only used for caching and not specifically as a Global Content Delivery Network (CDN).

Using Amazon Redshift as the data storage and CloudFront as the CDN is incorrect as Amazon Redshift is usually used as a Data Warehouse.

Using Amazon S3 Glacier as the data storage and ElastiCache as the CDN is incorrect as Amazon S3 Glacier is usually used for data archives.

Using multiple EC2 instance stores for data storage and ElastiCache as the CDN is incorrect as data stored in an instance store is not durable.

References:

<https://aws.amazon.com/s3/>

<https://aws.amazon.com/caching/cdn/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A Solutions Architect created a new Standard-class S3 bucket to store financial reports that are not frequently accessed but should immediately be available when an auditor requests them. To save costs, the Architect changed the storage class of the S3 bucket from Standard to Infrequent Access storage class.

In Amazon S3 Standard – Infrequent Access storage class, which of the following statements are true?
(Select TWO.)

- It provides high latency and low throughput performance.
- It automatically moves data to the most cost-effective access tier without any operational overhead.
- **It is designed for data that requires rapid access when needed.**
- **It is designed for data that is accessed less frequently.**
- Ideal to use for data archiving.

Correct

Amazon S3 Standard – Infrequent Access (Standard – IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard – IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee.

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads



This combination of low cost and high performance make Standard – IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard – IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of Standard
- Designed for durability of 99.999999999% of objects
- Designed for 99.9% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL encryption of data in transit and at rest
- Lifecycle management for automatic migration of objects

Hence, the correct answers are:

- ***It is designed for data that is accessed less frequently.***
- ***It is designed for data that requires rapid access when needed.***

The option that says: ***It automatically moves data to the most cost-effective access tier without any operational overhead*** is incorrect as it actually refers to Amazon S3 – Intelligent Tiering, which is the only cloud storage class that delivers automatic cost savings by moving objects between different access tiers when access patterns change.

The option that says: ***It provides high latency and low throughput performance*** is incorrect as it should be “low latency” and “high throughput” instead. S3 automatically scales performance to meet user demands.

The option that says: ***Ideal to use for data archiving*** is incorrect because this statement refers to Amazon S3 Glacier. Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.

References:

<https://aws.amazon.com/s3/storage-classes/>

<https://aws.amazon.com/s3/faqs>

Check out this Amazon S3 Cheat Sheet:

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An online medical system hosted in AWS stores sensitive Personally Identifiable Information (PII) of the users in an Amazon S3 bucket. Both the master keys and the unencrypted data should never be sent to AWS to comply with the strict compliance and regulatory requirements of the company.

Which S3 encryption technique should the Architect use?

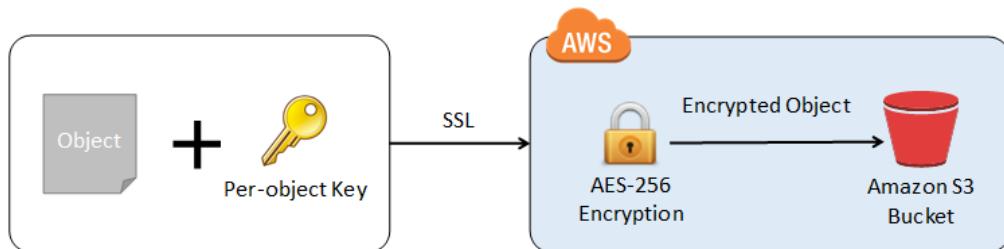
- Use S3 client-side encryption with a client-side master key.
- Use S3 client-side encryption with a KMS-managed customer master key.
- Use S3 server-side encryption with a KMS managed key.
- Use S3 server-side encryption with customer provided key.

Correct

Client-side encryption is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options:

- Use an AWS KMS-managed customer master key.
- Use a client-side master key.

When using an AWS KMS-managed customer master key to enable client-side data encryption, you provide an AWS KMS customer master key ID (CMK ID) to AWS. On the other hand, when you use client-side master key for client-side data encryption, **your client-side master keys and your unencrypted data are never sent to AWS**. It's important that you safely manage your encryption keys because if you lose them, you can't decrypt your data.



This is how client-side encryption using client-side master key works:

When uploading an object – You provide a client-side master key to the Amazon S3 encryption client. The client uses the master key only to encrypt the data encryption key that it generates randomly. The process works like this:

- \1. The Amazon S3 encryption client generates a one-time-use symmetric key (also known as a data encryption key or data key) locally. It uses the data key to encrypt the data of a single Amazon S3 object. The client generates a separate data key for each object.
- \2. The client encrypts the data encryption key using the master key that you provide. The client uploads the encrypted data key and its material description as part of the object metadata. The client uses the material description to determine which client-side master key to use for decryption.
- \3. The client uploads the encrypted data to Amazon S3 and saves the encrypted data key as object metadata (`x-amz-meta-x-amz-key`) in Amazon S3.

When downloading an object – The client downloads the encrypted object from Amazon S3. Using the material description from the object's metadata, the client determines which master key to use to decrypt the data key. The client uses that master key to decrypt the data key and then uses the data key to decrypt the object.

Hence, the correct answer is to ***use S3 client-side encryption with a client-side master key***.

Using S3 client-side encryption with a KMS-managed customer master key is incorrect because in client-side encryption with a KMS-managed customer master key, you provide an AWS KMS customer master key ID (CMK ID) to AWS. The scenario clearly indicates that both the master keys and the unencrypted data should never be sent to AWS.

Using S3 server-side encryption with a KMS managed key is incorrect because the scenario mentioned that the unencrypted data should never be sent to AWS, which means that you have to use client-side encryption in order to encrypt the data first before sending to AWS. In this way, you can ensure that there is no unencrypted data being uploaded to AWS. In addition, the master key used by Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS) is uploaded and managed by AWS, which directly violates the requirement of not uploading the master key.

Using S3 server-side encryption with customer provided key is incorrect because just as mentioned above, you have to use client-side encryption in this scenario instead of server-side encryption. For the S3 server-side encryption with customer-provided key (SSE-C), you actually provide the encryption key as part of your request to upload the object to S3. Using this key, Amazon S3 manages both the encryption (as it writes to disks) and decryption (when you access your objects).

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A company collects atmospheric data such as temperature, air pressure, and humidity from different countries. Each site location is equipped with various weather instruments and a high-speed Internet connection. The average collected data in each location is around 500 GB and will be analyzed by a weather forecasting application hosted in Northern Virginia. As the Solutions Architect, you need to aggregate all the data in the fastest way.

Which of the following options can satisfy the given requirement?

- Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.
- Use AWS Snowball Edge to transfer large amounts of data.
- Set up a Site-to-Site VPN connection.
- Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket.

Incorrect

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers industry-leading durability, availability, performance, security, and virtually unlimited scalability at very low costs. Amazon S3 is also designed to be highly flexible. Store any type and amount of data that you want; read the same piece of data a million times or only for emergency disaster recovery; build a simple FTP application or a sophisticated web application.



Amazon S3 Transfer Acceleration

Speed Comparison

Upload speed comparison in the selected region
(Based on the location of bucket: jbarr-public)

N. Virginia
(US-EAST-1)

539% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

This speed checker uses multipart uploads to transfer a file from your browser to various Amazon S3 regions with and without Amazon S3 Transfer Acceleration. It compares the speed results and shows the percentage difference for every region.

Note: In general, the farther away you are from an Amazon S3 region, the higher the speed improvement you can expect from using Amazon S3 Transfer Acceleration. If you see similar speed results with and without the acceleration, your upload bandwidth or a system constraint might be limiting your speed.

Upload speed comparison in other regions

N. California
(US-WEST-1)

73% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Oregon
(US-WEST-2)

17% slower

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Ireland
(EU-WEST-1)

919% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Frankfurt
(EU-CENTRAL-1)

928% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Tokyo
(AP-NORTHEAST-1)

680% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Seoul
(AP-NORTHEAST-2)

822% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Singapore
(AP-SOUTHEAST-1)

1261% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Sydney
(AP-SOUTHEAST-2)

1226% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

São Paulo
(SA-EAST-1)

1000% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Since the weather forecasting application is located in N.Virginia, you need to transfer all the data in the same AWS Region. With Amazon S3 Transfer Acceleration, you can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Multipart upload allows you to upload a single object as a set of parts. After all the parts of your object are uploaded, Amazon S3 then presents the data as a single object. This approach is the fastest way to aggregate all the data.

Hence, the correct answer is: ***Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.***

The option that says: ***Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket*** is incorrect because replicating the objects to the destination bucket takes about 15 minutes. Take note that the requirement in the scenario is to aggregate the data in the fastest way.

The option that says: ***Use AWS Snowball Edge to transfer large amounts of data*** is incorrect because the end-to-end time to transfer up to 80 TB of data into AWS Snowball Edge is approximately one week.

The option that says: ***Set up a Site-to-Site VPN connection*** is incorrect because setting up a VPN connection is not needed in this scenario. Site-to-Site VPN is just used for establishing secure connections between an on-premises network and Amazon VPC. Also, this approach is not the fastest way to transfer your data. You must use Amazon S3 Transfer Acceleration.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/replication.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

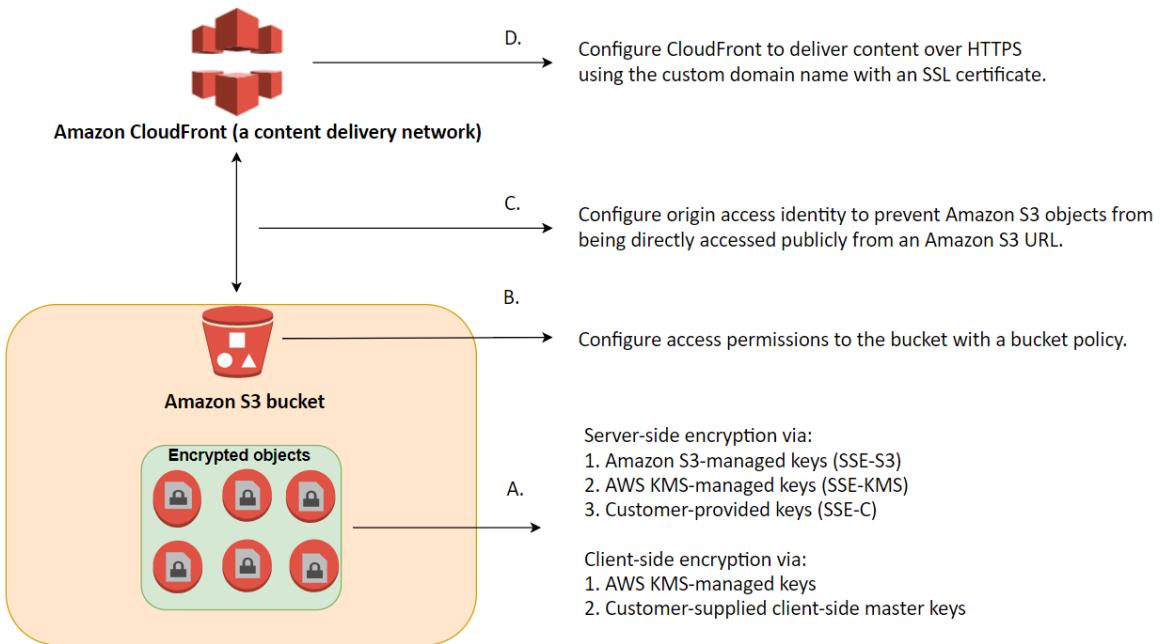
For data privacy, a healthcare company has been asked to comply with the Health Insurance Portability and Accountability Act (HIPAA). The company stores all its backups on an Amazon S3 bucket. It is required that data stored on the S3 bucket must be encrypted.

What is the best option to do this? (Select TWO.)

- Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption.
- Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys.
- Enable Server-Side Encryption on an S3 bucket to make use of AES-128 encryption.
- Store the data on EBS volumes with encryption enabled instead of using Amazon S3.
- Store the data in encrypted EBS snapshots.

Correct

Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a pre-signed URL, that URL works the same way for both encrypted and unencrypted objects.



You have three mutually exclusive options depending on how you choose to manage the encryption keys:

1. Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
2. Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
3. Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

The options that say: ***Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys*** and ***Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption*** are correct because these options are using client-side encryption and Amazon S3-Managed Keys (SSE-S3) respectively. **Client-side encryption** is the act of encrypting data before sending it to Amazon S3 while SSE-S3 uses AES-256 encryption.

Storing the data on EBS volumes with encryption enabled instead of using Amazon S3 and ***storing the data in encrypted EBS snapshots*** are incorrect because both options use EBS encryption and not S3.

Enabling Server-Side Encryption on an S3 bucket to make use of AES-128 encryption is incorrect as S3 doesn't provide AES-128 encryption, only AES-256.

References:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

8. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

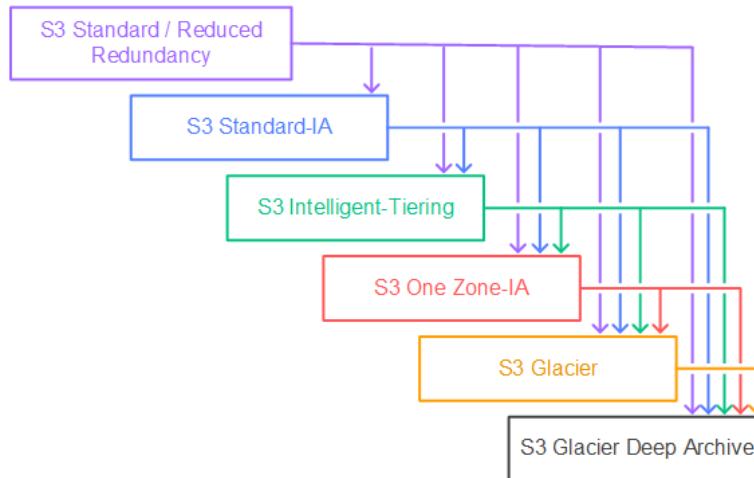
There are a few, easily reproducible but confidential files that your client wants to store in AWS without worrying about storage capacity. For the first month, all of these files will be accessed frequently but after that, they will rarely be accessed at all. The old files will only be accessed by developers so there is no set retrieval time requirement. However, the files under a specific `tdojo-finance` prefix in the S3 bucket will be used for post-processing that requires millisecond retrieval time.

Given these conditions, which of the following options would be the most cost-effective solution for your client's storage needs?

- Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the bucket to Intelligent-Tiering using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to S3-IA while the remaining go to Glacier using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the bucket to S3-IA using lifecycle policy.

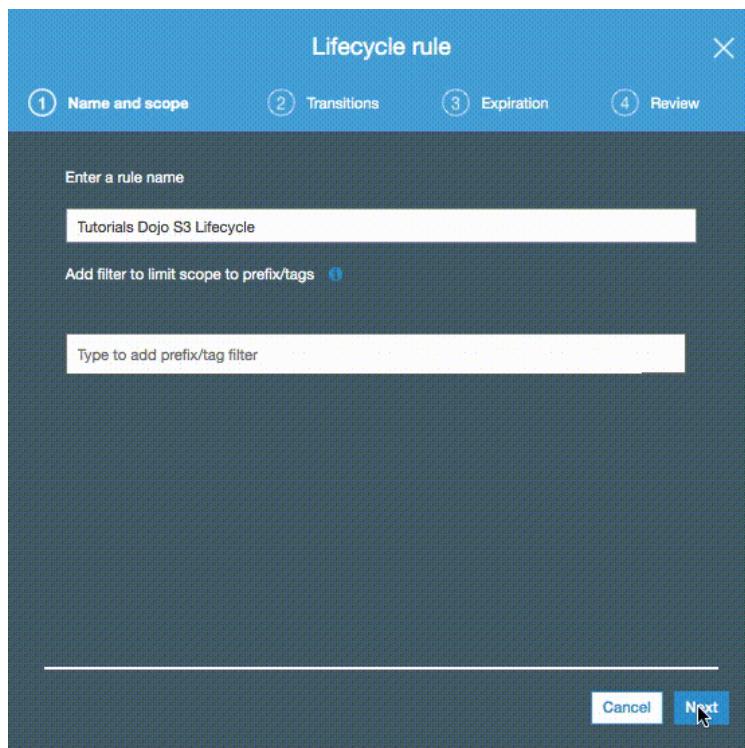
Incorrect

Initially, the files will be accessed frequently, and S3 is a durable and highly available storage solution for that. After a month has passed, the files won't be accessed frequently anymore, so it is a good idea to use lifecycle policies to move them to a storage class that would have a lower cost for storing them.



Since the files are easily reproducible and some of them are needed to be retrieved quickly based on a specific prefix filter (`tdojo-finance`), S3-One Zone IA would be a good choice for storing them. The other files that do not contain such prefix would then be moved to Glacier for low-cost archival. This setup would also be the most cost-effective for the client.

Hence, the correct answer is: ***Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy***.



The option that says: ***Storing the files in S3 then after a month, changing the storage class of the bucket to S3-IA using lifecycle policy*** is incorrect. Although it is valid to move the files to S3-IA, [this solution still](#) costs more compared with using a combination of S3-One Zone IA and Glacier.

The option that says: ***Storing the files in S3 then after a month, changing the storage class of the bucket to Intelligent-Tiering using lifecycle policy*** is incorrect. While S3 Intelligent-Tiering can automatically move data between two access tiers (frequent access and infrequent access) when access patterns change, it is more suitable for scenarios where you don't know the access patterns of your data. It may take some time for S3 Intelligent-Tiering to analyze the access patterns before it moves the data to a cheaper storage class like S3-IA which means you may still end up paying more in the beginning. In addition, you already know the access patterns of the files which means you can directly change the storage class immediately and save cost right away.

The option that says: ***Storing the files in S3 then after a month, changing the storage class of the `tdojo-finance` prefix to S3-IA while the remaining go to Glacier using lifecycle policy*** is incorrect. Even though S3-IA costs less than the S3 Standard storage class, it is still more expensive than S3-One Zone IA. Remember that the files are easily reproducible so you can safely move the data to S3-One Zone IA and in case there is an outage, you can simply generate the missing data again.

References:

<https://aws.amazon.com/blogs/compute/amazon-s3-adds-prefix-and-suffix-filters-for-lambda-function-triggering>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-configuration-examples.html>

<https://aws.amazon.com/s3/pricing>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A company has a web-based ticketing service that utilizes Amazon SQS and a fleet of EC2 instances. The EC2 instances that consume messages from the SQS queue are configured to poll the queue as often as possible to keep end-to-end throughput as high as possible. The Solutions Architect noticed that polling the queue in tight loops is using unnecessary CPU cycles, resulting in increased operational costs due to empty responses.

In this scenario, what should the Solutions Architect do to make the system more cost-effective?

- Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to zero.
- Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero.**
- Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to zero.
- Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero.

Correct

In this scenario, the application is deployed in a fleet of EC2 instances that are polling messages from a single SQS queue. **Amazon SQS uses short polling by default, querying only a subset of the servers (based on a weighted random distribution) to determine whether any messages are available for inclusion in the response. Short polling works for scenarios that require higher throughput. However, you can also configure the queue to use Long polling instead, to reduce cost.**

The ReceiveMessageWaitTimeSeconds is the queue attribute that determines whether you are using Short or Long polling. By default, its value is zero which means it is using Short polling. If it is set to a value greater than zero, then it is Long polling.

Hence, ***configuring Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero is the correct answer.***

Quick facts about SQS Long Polling:

- Long polling helps reduce your cost of using Amazon SQS by reducing 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 and eliminating false empty responses when messages are available in the queue but aren't included in the response.
- **Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the response to the `ReceiveMessage` request contains at least one of the available messages, up to the maximum number of messages specified in the `ReceiveMessage` action.**
- **Long polling eliminates false empty responses by querying all (rather than a limited number) of the servers. Long polling returns messages as soon any message becomes available.**

Reference:

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

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

The start-up company that you are working for has a batch job application that is currently hosted on an EC2 instance. It is set to process messages from a queue created in SQS with default settings. You configured the application to process the messages once a week. After 2 weeks, you noticed that not all messages are being processed by the application.

What is the root cause of this issue?

- Missing permissions in SQS.
- The SQS queue is set to short-polling.
- Amazon SQS has automatically deleted the messages that have been in a queue for more than the maximum message retention period.
- The batch job application is configured to long polling.

Incorrect

Amazon SQS automatically deletes messages that have been in a queue for more than the maximum message retention period. The default message retention period is 4 days. Since the queue is configured to the default settings and the batch job application only processes the messages once a week, the messages that are in the queue for more than 4 days are deleted. This is the root cause of the issue.

To fix this, you can increase the message retention period to a maximum of 14 days using the [SetQueueAttributes](#) action.

References:

<https://aws.amazon.com/sqs/faqs/>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-message-lifecycle.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

An e-commerce application is using a fanout messaging pattern for its order management system. For every order, it sends an Amazon SNS message to an SNS topic, and the message is replicated and pushed to multiple Amazon SQS queues for parallel asynchronous processing. A Spot EC2 instance retrieves the message from each SQS queue and processes the message. There was an incident that while an EC2 instance is currently processing a message, the instance was abruptly terminated, and the processing was not completed in time.

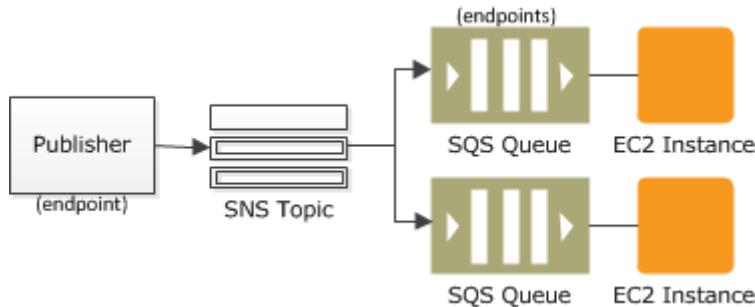
In this scenario, what happens to the SQS message?

- The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online.
- The message will be sent to a Dead Letter Queue in AWS DataSync.
- When the message visibility timeout expires, the message becomes available for processing by other EC2 instances
- The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout.

Correct

A “fanout” pattern is when an Amazon SNS message is sent to a topic and then replicated and pushed to multiple Amazon SQS queues, HTTP endpoints, or email addresses. This allows for parallel asynchronous processing. For example, you could develop an application that sends an Amazon SNS message to a topic whenever an order is placed for a product. Then, the Amazon SQS queues that are subscribed to that

topic would receive identical notifications for the new order. The Amazon EC2 server instance attached to one of the queues could handle the processing or fulfillment of the order, while the other server instance could be attached to a data warehouse for analysis of all orders received.



When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a *visibility timeout*, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

The option that says: *The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout* is incorrect because the message will not be automatically assigned to the same EC2 instance once it is abruptly terminated. When the message visibility timeout expires, the message becomes available for processing by other EC2 instances.

The option that says: *The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online* is incorrect because the message will not be deleted and won't be duplicated in the SQS queue when the EC2 instance comes online.

The option that says: *The message will be sent to a Dead Letter Queue in AWS DataSync* is incorrect because although the message could be programmatically sent to a Dead Letter Queue (DLQ), it won't be handled by AWS DataSync but by Amazon SQS instead. AWS DataSync is primarily used to simplify your migration with AWS. It makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3 or Amazon Elastic File System (Amazon EFS).

References:

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

<https://docs.aws.amazon.com/sns/latest/dg/sns-common-scenarios.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs>

4. QUESTION

Category: CSAA – Design Resilient Architectures

An investment bank has a distributed batch processing application which is hosted in an Auto Scaling group of Spot EC2 instances with an SQS queue. You configured your components to use client-side buffering so that the calls made from the client will be buffered first and then sent as a batch request to SQS.

What is a period of time during which the SQS queue prevents other consuming components from receiving and processing a message?

- **Visibility Timeout**
- Component Timeout
- Receiving Timeout
- Processing Timeout

Correct

The visibility timeout is a period of time during which Amazon SQS prevents other consuming components from receiving and processing a message.

When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a ***visibility timeout***, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

References:

<https://aws.amazon.com/sqs/faqs/>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company needs to integrate the Lightweight Directory Access Protocol (LDAP) directory service from the on-premises data center to the AWS VPC using IAM. The identity store which is currently being used is not compatible with SAML.

Which of the following provides the most valid approach to implement the integration?

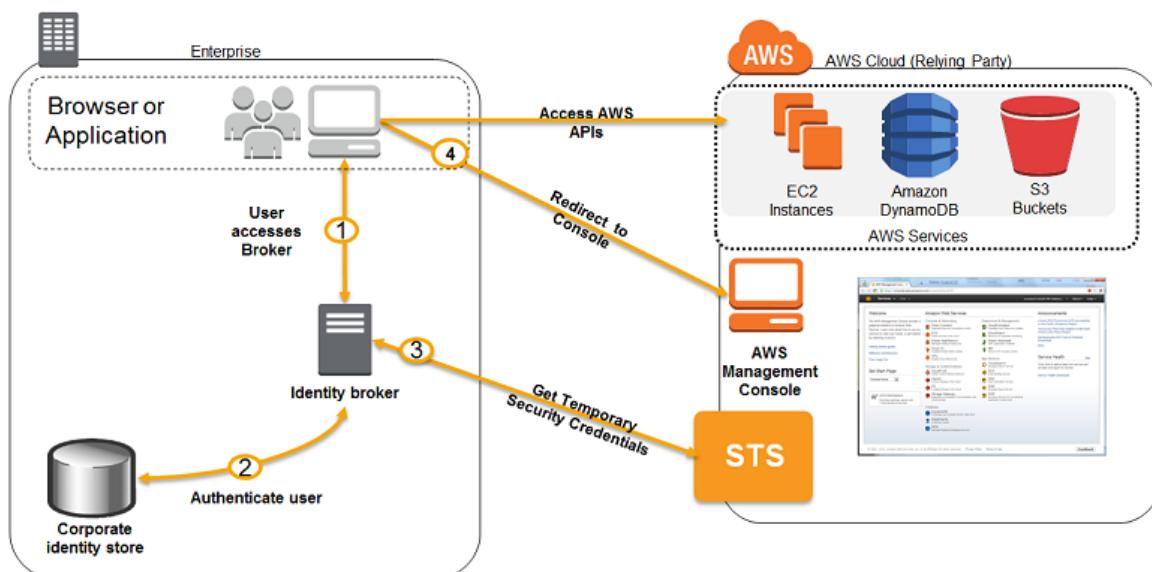
- Use AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP.
- Use an IAM policy that references the LDAP identifiers and AWS credentials.
- Use IAM roles to rotate the IAM credentials whenever LDAP credentials are updated.
- Develop an on-premises custom identity broker application and use STS to issue short-lived AWS credentials.

Incorrect

If your identity store is not compatible with SAML 2.0 then you can build a custom identity broker application to perform a similar function. The broker application authenticates users, requests temporary credentials for users from AWS, and then provides them to the user to access AWS resources.

The application verifies that employees are signed into the existing corporate network's identity and authentication system, which might use LDAP, Active Directory, or another system. The identity broker application then obtains temporary security credentials for the employees.

To get temporary security credentials, the identity broker application calls either `AssumeRole` or `GetFederationToken` to obtain temporary security credentials, depending on how you want to manage the policies for users and when the temporary credentials should expire. The call returns temporary security credentials consisting of an AWS access key ID, a secret access key, and a session token. The identity broker application makes these temporary security credentials available to the internal company application. The app can then use the temporary credentials to make calls to AWS directly. The app caches the credentials until they expire, and then requests a new set of temporary credentials.



Using an IAM policy that references the LDAP identifiers and AWS credentials is incorrect because using an IAM policy is not enough to integrate your LDAP service to IAM. You need to use SAML, STS, or a custom identity broker.

Using AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP is incorrect because the scenario did not require SSO and in addition, the identity store that you are using is not SAML-compatible.

Using IAM roles to rotate the IAM credentials whenever LDAP credentials are updated is incorrect because manually rotating the IAM credentials is not an optimal solution to integrate your on-premises and VPC network. You need to use SAML, STS, or a custom identity broker.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_federated-users.html

<https://aws.amazon.com/blogs/aws/aws-identity-and-access-management-now-with-identity-federation/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An Intelligence Agency developed a missile tracking application that is hosted on both development and production AWS accounts. The Intelligence agency's junior developer only has access to the development account. She has received security clearance to access the agency's production account but the access is only temporary and only write access to EC2 and S3 is allowed.

Which of the following allows you to issue short-lived access tokens that act as temporary security credentials to allow access to your AWS resources?

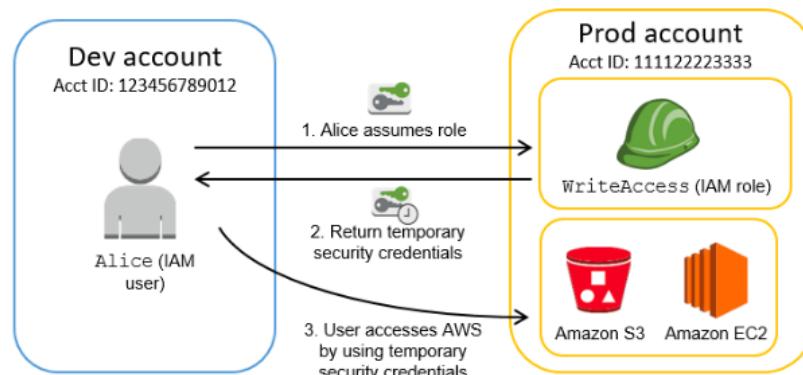
- Use AWS Cognito to issue JSON Web Tokens (JWT)
- All of the given options are correct.
- **Use AWS STS**
- Use AWS SSO

Correct

AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

1. Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
2. STS returns a set of temporary security credentials.
3. Alice uses the temporary security credentials to access services and resources in the Prod account. Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.



Using AWS Cognito to issue JSON Web Tokens (JWT) is incorrect because the Amazon Cognito service is primarily used for user authentication and not for providing access to your AWS resources. A JSON Web Token (JWT) is meant to be used for user authentication and session management.

Using AWS SSO is incorrect. Although the AWS SSO service uses STS, it does not issue short-lived credentials by itself. **AWS Single Sign-On (SSO)** is a cloud SSO service that makes it easy to centrally manage SSO access to multiple AWS accounts and business applications.

The option that says ***All of the above*** is incorrect as only STS has the ability to provide temporary security credentials.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

3. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A tech company that you are working for has undertaken a Total Cost Of Ownership (TCO) analysis evaluating the use of Amazon S3 versus acquiring more storage hardware. The result was that all 1200 employees would be granted access to use Amazon S3 for storage of their personal documents.

Which of the following will you need to consider so you can set up a solution that incorporates single sign-on feature from your corporate AD or LDAP directory and also restricts access for each individual user to a designated user folder in an S3 bucket? (Select TWO.)

- Configure an IAM role and an IAM Policy to access the bucket.
- Set up a Federation proxy or an Identity provider, and use AWS Security Token Service to generate temporary tokens.
- Set up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket.
- Use 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others.
- Map each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents.

Incorrect

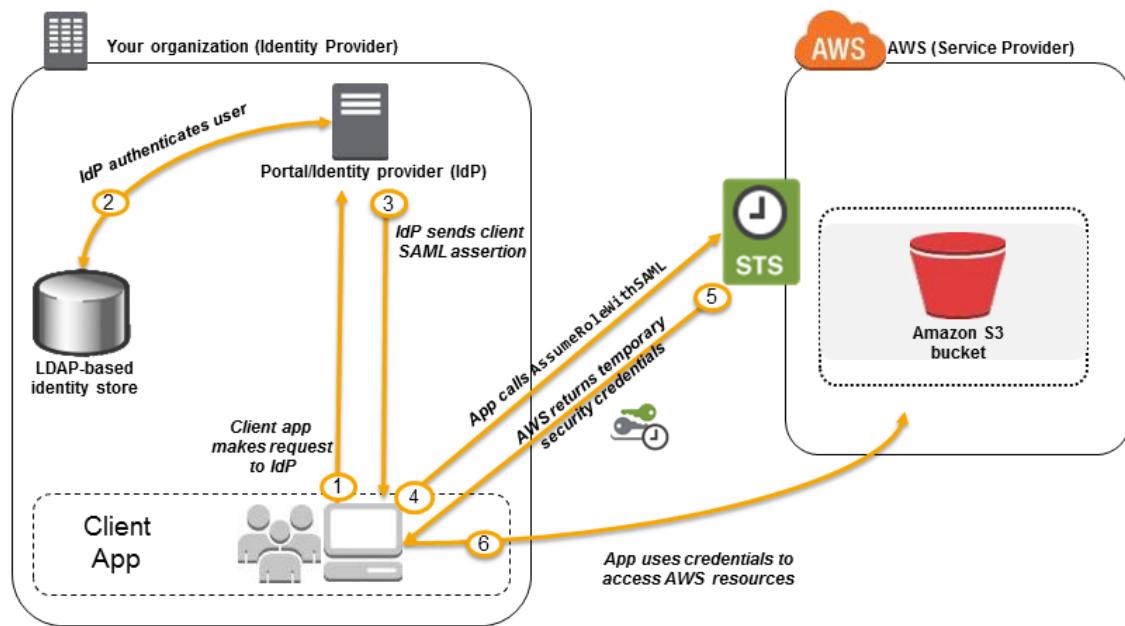
The question refers to one of the common scenarios for temporary credentials in AWS. Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles. In this example, it is called **enterprise identity federation** considering that you also need to set up a single sign-on (SSO) capability.

The correct answers are:

– Setup a Federation proxy or an Identity provider

– Setup an AWS Security Token Service to generate temporary tokens

– Configure an IAM role and an IAM Policy to access the bucket.



In an enterprise identity federation, you can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with a separate user name and password. This is known as the *single sign-on* (SSO) approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your own solution for federating user identities.

Using 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others is incorrect since you don't have to use 3rd party solutions to provide the access. AWS already provides the necessary tools that you can use in this situation.

Mapping each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents is incorrect as there is no direct way of integrating Amazon S3 with Amazon WorkDocs for this particular scenario. Amazon WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service. With Amazon WorkDocs, you can easily create, edit, and share content. And because it's stored centrally on AWS, you can access it from anywhere on any device.

Setting up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket is incorrect since creating that many IAM users would be unnecessary. Also, you want the account to integrate with your AD or LDAP directory, hence, IAM Users does not fit these criteria.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc.html

<https://aws.amazon.com/blogs/security/writing-iam-policies-grant-access-to-user-specific-folders-in-an-amazon-s3-bucket/>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A mobile application stores pictures in Amazon Simple Storage Service (S3) and allows application sign-in using an OpenID Connect-compatible identity provider.

Which AWS Security Token Service approach to temporary access should you use for this scenario?

- Web Identity Federation
- SAML-based Identity Federation
- Cross-Account Access
- AWS Identity and Access Management roles

Incorrect

With web identity federation, you don't need to create custom sign-in code or manage your own user identities. Instead, users of your app can sign in using a well-known identity provider (IdP) —such as Login with Amazon, Facebook, Google, or any other OpenID Connect (OIDC)-compatible IdP, receive an authentication token, and then exchange that token for temporary security credentials in AWS that map to an IAM role with permissions to use the resources in your AWS account. Using an IdP helps you keep your AWS account secure because you don't have to embed and distribute long-term security credentials with your application.

Reference:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect is managing a company's AWS account of approximately 300 IAM users. They have a new company policy that requires changing the associated permissions of all 100 IAM users that control the access to Amazon S3 buckets.

What will the Solutions Architect do to avoid the time-consuming task of applying the policy to each user?

- Create a new S3 bucket access policy with unlimited access for each IAM user.
- **Create a new IAM group and then add the users that require access to the S3 bucket. Afterwards, apply the policy to IAM group.**
- Create a new IAM role and add each user to the IAM role.
- Create a new policy and apply it to multiple IAM users using a shell script.

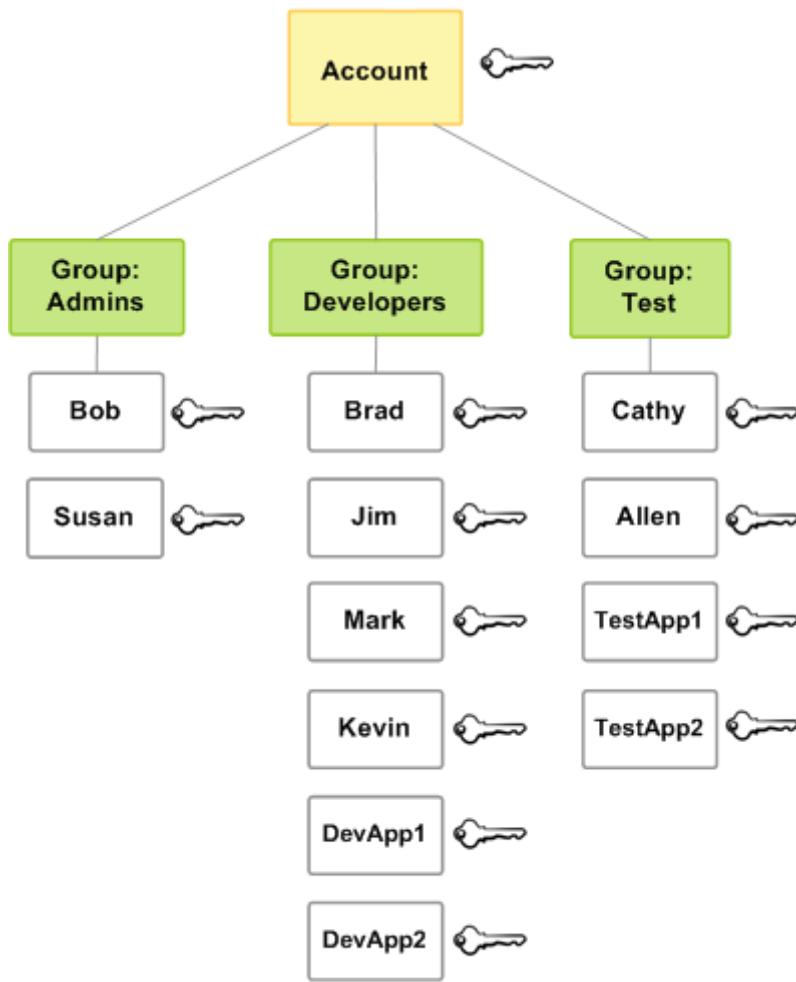
Correct

In this scenario, the best option is to ***group the set of users in an IAM Group and then apply a policy with the required access to the Amazon S3 bucket***. This will enable you to easily add, remove, and manage the users instead of manually adding a policy to each and every 100 IAM users.

Creating a new policy and applying it to multiple IAM users using a shell script is incorrect because you need a new IAM Group for this scenario and not assign a policy to each user via a shell script. This method can save you time but afterward, it will be difficult to manage all 100 users that are not contained in an IAM Group.

Creating a new S3 bucket access policy with unlimited access for each IAM user is incorrect because you need a new IAM Group and the method is also time-consuming.

Creating a new IAM role and adding each user to the IAM role is incorrect because you need to use an IAM Group and not an IAM role.



Reference:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

6. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect created a brand new IAM User with a default setting using AWS CLI. This is intended to be used to send API requests to Amazon S3, DynamoDB, Lambda, and other AWS resources of the company's cloud infrastructure.

Which of the following must be done to allow the user to make API calls to the AWS resources?

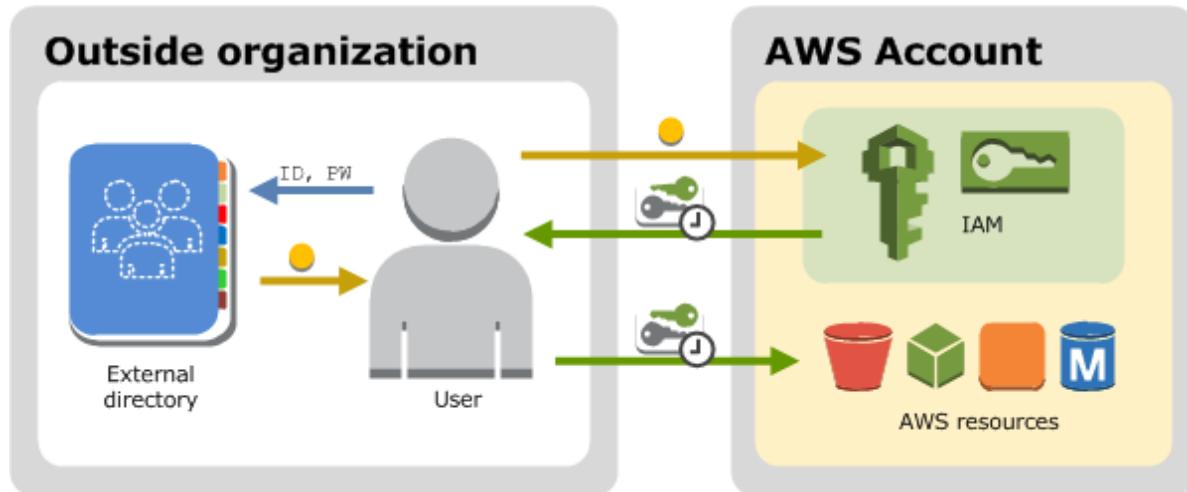
- Do nothing as the IAM User is already capable of sending API calls to your AWS resources.
- **Create a set of Access Keys for the user and attach the necessary permissions.**
- Enable Multi-Factor Authentication for the user.
- Assign an IAM Policy to the user to allow it to send API calls.

Incorrect

You can choose the credentials that are right for your IAM user. When you use the AWS Management Console to create a user, you must choose to at least include a console password or access keys. By default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. You must create the type of credentials for an IAM user based on the needs of your user.

Access keys are long-term credentials for an IAM user or the AWS account root user. You can use access keys to sign programmatic requests to the AWS CLI or AWS API (directly or using the AWS SDK). Users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.

To fill this need, you can create, modify, view, or rotate access keys (access key IDs and secret access keys) for IAM users. When you create an access key, IAM returns the access key ID and secret access key. You should save these in a secure location and give them to the user.



The option that says: ***Do nothing as the IAM User is already capable of sending API calls to your AWS resources*** is incorrect because by default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. Take note that in the scenario, you created the new IAM user using the AWS CLI and not via the AWS Management Console, where you must choose to at least include a console password or access keys when creating a new IAM user.

Enabling Multi-Factor Authentication for the user is incorrect because this will still not provide the required Access Keys needed to send API calls to your AWS resources. You have to grant the IAM user with Access Keys to meet the requirement.

Assigning an IAM Policy to the user to allow it to send API calls is incorrect because adding a new IAM policy to the new user will not grant the needed Access Keys needed to make API calls to the AWS resources.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html#id_users_creds

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

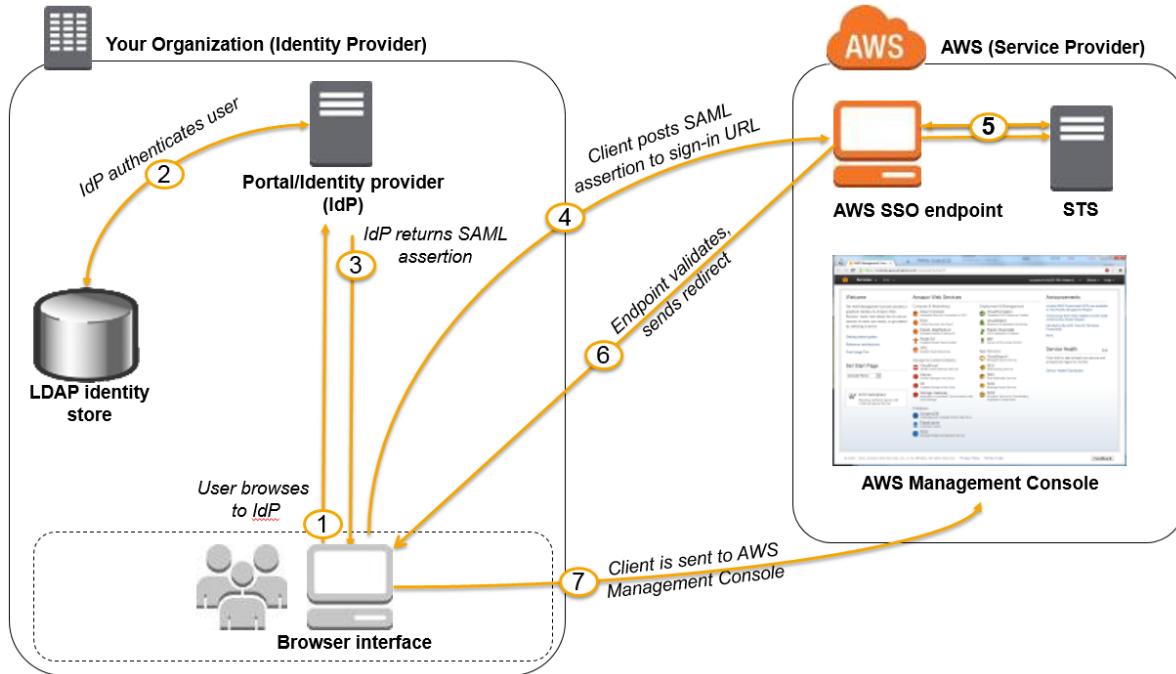
A pharmaceutical company has resources hosted on both their on-premises network and in AWS cloud. They want all of their Software Architects to access resources on both environments using their on-premises credentials, which is stored in Active Directory.

In this scenario, which of the following can be used to fulfill this requirement?

- Set up SAML 2.0-Based Federation by using a Web Identity Federation.
- **Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).**
- Use IAM users
- Use Amazon VPC

Incorrect

Since the company is using Microsoft Active Directory which implements Security Assertion Markup Language (SAML), you can set up a SAML-Based Federation for API Access to your AWS cloud. In this way, you can easily connect to AWS using the login credentials of your on-premises network.



AWS supports identity federation with SAML 2.0, an open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS APIs without you having to create an IAM user for everyone in your organization. By using SAML, you can simplify the process of configuring federation with AWS, because you can use the IdP's service instead of writing custom identity proxy code.

Before you can use SAML 2.0-based federation as described in the preceding scenario and diagram, you must configure your organization's IdP and your AWS account to trust each other. The general process for configuring this trust is described in the following steps. Inside your organization, you must have an IdP that supports SAML 2.0, like Microsoft Active Directory Federation Service (AD FS, part of Windows Server), Shibboleth, or another compatible SAML 2.0 provider.

Hence, the correct answer is: ***Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).***

Setting up SAML 2.0-Based Federation by using a Web Identity Federation is incorrect because this is primarily used to let users sign in via a well-known external identity provider (IdP), such as Login with Amazon, Facebook, Google. It does not utilize Active Directory.

Using IAM users is incorrect because the situation requires you to use the existing credentials stored in their Active Directory, and not user accounts that will be generated by IAM.

Using Amazon VPC is incorrect because this only lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. This has nothing to do with user authentication or Active Directory.

References:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

8. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company recently adopted a hybrid architecture that integrates its on-premises data center to AWS cloud. You are assigned to configure the VPC and implement the required IAM users, IAM roles, IAM groups, and IAM policies.

In this scenario, what is the best practice when creating IAM policies?

- Use the principle of least privilege which means granting only the permissions required to perform a task.
- Determine what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations.
- Grant all permissions to any EC2 user.
- Use the principle of least privilege which means granting only the least number of people with full root access.

Correct

One of the best practices in AWS IAM is to **grant least privilege**.

When you create IAM policies, follow the standard security advice of granting *least privilege*—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform *only* those tasks.

Therefore, ***using the principle of least privilege which means granting only the permissions required to perform a task*** is the correct answer.

Start with a minimum set of permissions and grant additional permissions as necessary. Defining the right set of permissions requires some understanding of the user's objectives. Determine what is required for the specific task, what actions a particular service supports, and what permissions are required in order to perform those actions.

Granting all permissions to any EC2 user is incorrect since you don't want your users to gain access to everything and perform unnecessary actions. Doing so is not a good security practice.

Using the principle of least privilege which means granting only the least number of people with full root access is incorrect because this is not the correct definition of what the principle of least privilege is.

Determining what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations is incorrect since there are some users who you should not give administrative access to. You should follow the principle of least privilege when providing permissions and accesses to your resources.

Reference:

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#use-groups-for-permissions>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

Service Control Policies (SCP) vs IAM Policies:

<https://tutorialsdojo.com/service-control-policies-scp-vs-iam-policies/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company hosted an e-commerce website on an Auto Scaling group of EC2 instances behind an Application Load Balancer. The Solutions Architect noticed that the website is receiving a large number of illegitimate external requests from multiple systems with IP addresses that constantly change. To resolve the performance issues, the Solutions Architect must implement a solution that would block the illegitimate requests with minimal impact on legitimate traffic.

Which of the following options fulfills this requirement?

- Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.
- Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer.
- Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests.
- Create a custom rule in the security group of the Application Load Balancer to block the offending requests.

Correct

AWS WAF is tightly integrated with Amazon CloudFront, the Application Load Balancer (ALB), Amazon API Gateway, and AWS AppSync – services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on regional services, such as Application Load Balancer, Amazon API Gateway, and AWS AppSync, your rules run in the region and can be used to protect Internet-facing resources as well as internal resources.

[WAF: Web Application Firewall](#)

Rule

Name
tutorialsdojo-rule
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Type
Rate-based rule

Request rate details

Rate limit
The rate limit is the maximum number of requests from a single IP address that are allowed in a five-minute period. This value is continually evaluated, and requests will be blocked once this limit is reached. The IP address is automatically unblocked after it falls below the limit.

100

Rate limit must be between 100 and 20,000,000.

IP address to use for rate limiting
When a request comes through a CDN or other proxy network, the source IP address identifies the proxy and the original IP address is sent in a header. Use caution with the option, IP address in header, because headers can be handled inconsistently by proxies and they can be modified to bypass inspection.

Source IP address
 IP address in header

Criteria to count request towards rate limit
Choose whether to count all requests for each IP address or to only count requests that match the criteria of a rule statement.

Consider all requests
 Only consider requests that match the criteria in a rule statement

A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests.

Based on the given scenario, the requirement is to limit the number of requests from the illegitimate requests without affecting the genuine requests. To accomplish this requirement, you can use AWS WAF web ACL. There are two types of rules in creating your own web ACL rule: regular and rate-based rules. You need to select the latter to add a rate limit to your web ACL. After creating the web ACL, you can associate it with ALB. When the rule action triggers, AWS WAF applies the action to additional requests from the IP address until the request rate falls below the limit.

Hence, the correct answer is: ***Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.***

The option that says: ***Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer*** is incorrect because a regular rule only matches the statement defined in the rule. If you need to add a rate limit to your rule, you should create a rate-based rule.

The option that says: ***Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests*** is incorrect. Although NACLs can help you block incoming traffic, this option wouldn't be able to limit the number of requests from a single IP address that is dynamically changing.

The option that says: ***Create a custom rule in the security group of the Application Load Balancer to block the offending requests*** is incorrect because the security group can only allow incoming traffic. Remember that you can't deny traffic using security groups. In addition, it is not capable of limiting the rate of traffic to your application unlike AWS WAF.

References:

<https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-rate-based.html>

<https://aws.amazon.com/waf/faqs/>

Check out this AWS WAF Cheat Sheet:

<https://tutorialsdojo.com/aws-waf/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

2. QUESTION

Category: CSAA – Design Resilient Architectures

A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture that is composed of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used by an online accounting application that requires path-based routing, host-based routing, and bi-directional communication channels using WebSockets.

Which is the most suitable type of Elastic Load Balancer that will satisfy the given requirement?

- Either a Classic Load Balancer or a Network Load Balancer
- Classic Load Balancer
- Network Load Balancer
- **Application Load Balancer**

Incorrect

Elastic Load Balancing supports three types of load balancers. You can select the appropriate load balancer based on your application needs.

If you need flexible application management and TLS termination then it is recommended to use Application Load Balancer. If extreme performance and static IP is needed for your application then it is recommend that you use Network Load Balancer. If your application is built within the EC2 Classic network then you should use Classic Load Balancer.

An **Application Load Balancer** functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.

THEN

1. Redirect to...

Original value: #{port}

Custom host, path, query

Host Original value: #{host}
#{host}

Path Original value: /#{path}
/new/#{path}

Query Original value: #{query}
#{query}

301 - Permanently moved

Application Load Balancers support path-based routing, host-based routing, and support for containerized applications hence, ***Application Load Balancer*** is the correct answer.

Network Load Balancer, ***Classic Load Balancer***, and ***either a Classic Load Balancer or a Network Load Balancer*** are all incorrect as none of these support path-based routing and host-based routing, unlike an Application Load Balancer.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits>

<https://aws.amazon.com/elasticloadbalancing/faqs/>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

<https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/>

3. QUESTION

Category: CSAA – Design Secure Applications and Architectures

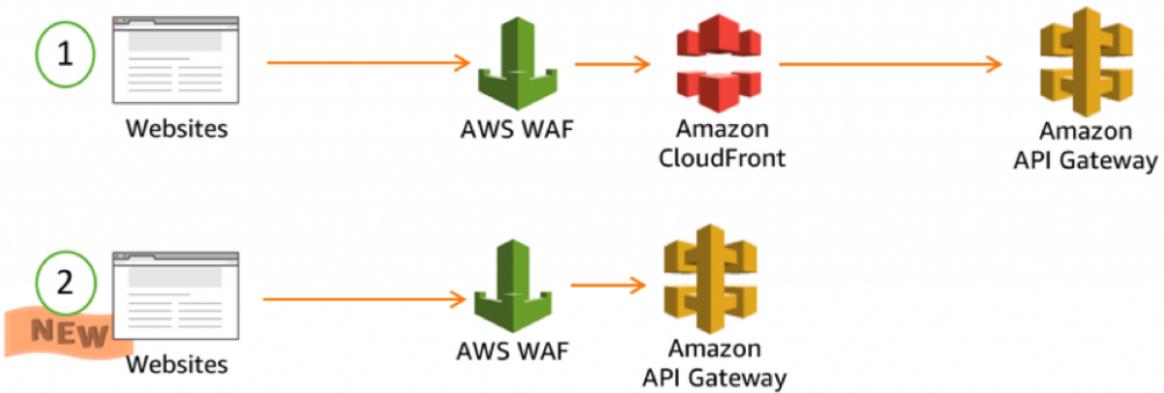
A company is hosting its web application in an Auto Scaling group of EC2 instances behind an Application Load Balancer. Recently, the Solutions Architect identified a series of SQL injection attempts and cross-site scripting attacks to the application, which had adversely affected their production data.

Which of the following should the Architect implement to mitigate this kind of attack?

- Block all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List.
- Using AWS Firewall Manager, set up security rules that block SQL injection and cross-site scripting attacks. Associate the rules to the Application Load Balancer.
- Use Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application.
- Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.

Correct

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.



At the simplest level, AWS WAF lets you choose one of the following behaviors:

Allow all requests except the ones that you specify – This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.

Block all requests except the ones that you specify – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.

Count the requests that match the properties that you specify – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

Hence, the correct answer in this scenario is: ***Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.***

Using Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application is incorrect because Amazon GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

Using AWS Firewall Manager to set up security rules that block SQL injection and cross-site scripting attacks, then associating the rules to the Application Load Balancer is incorrect because AWS Firewall Manager just simplifies your AWS WAF and AWS Shield Advanced administration and maintenance tasks across multiple accounts and resources.

Blocking all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List is incorrect because this is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACLs are not effective in blocking SQL injection and cross-site scripting attacks

References:

<https://aws.amazon.com/waf/>

<https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html>

Check out this AWS WAF Cheat Sheet:

<https://tutorialsdojo.com/aws-waf/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company has a web application hosted on a fleet of EC2 instances located in two Availability Zones that are all placed behind an Application Load Balancer. As a Solutions Architect, you have to add a health check configuration to ensure your application is highly-available.

Which health checks will you implement?

- **HTTP or HTTPS health check**
- ICMP health check
- TCP health check
- FTP health check

Correct

A load balancer takes requests from clients and distributes them across the EC2 instances that are registered with the load balancer. You can create a load balancer that listens to both the HTTP (80) and HTTPS (443) ports. If you specify that the HTTPS listener sends requests to the instances on port 80, the load balancer terminates the requests, and communication from the load balancer to the instances is not encrypted. If the HTTPS listener sends requests to the instances on port 443, communication from the load balancer to the instances is encrypted.

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port	
HTTP	80	HTTP	80	X
HTTPS (Secure HTTP)	443	HTTPS (Secure HTTP)	443	X
Add				

If your load balancer uses an encrypted connection to communicate with the instances, you can optionally enable authentication of the instances. This ensures that the load balancer communicates with an instance only if its public key matches the key that you specified to the load balancer for this purpose.

The type of ELB that is mentioned in this scenario is an Application Elastic Load Balancer. This is used if you want a flexible feature set for your web applications with HTTP and HTTPS traffic. Conversely, it only allows 2 types of health check: HTTP and HTTPS.

Hence, the correct answer is: ***HTTP or HTTPS health check.***

ICMP health check and ***FTP health check*** are incorrect as these are not supported.

TCP health check is incorrect. A TCP health check is only offered in Network Load Balancers and Classic Load Balancers.

References:

<http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html>

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

EC2 Instance Health Check vs ELB Health Check vs Auto Scaling and Custom Health Check:

<https://tutorialsdojo.com/ec2-instance-health-check-vs-elb-health-check-vs-auto-scaling-and-custom-health-check/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

5. QUESTION

Category: CSAA – Design High-Performing Architectures

A fast food company is using AWS to host their online ordering system which uses an Auto Scaling group of EC2 instances deployed across multiple Availability Zones with an Application Load Balancer in front. To better handle the incoming traffic from various digital devices, you are planning to implement a new routing system where requests which have a URL of /api/android are forwarded to one specific target group named “Android-Target-Group”. Conversely, requests which have a URL of /api/ios are forwarded to another separate target group named “iOS-Target-Group”.

How can you implement this change in AWS?

- Use path conditions to define rules that forward requests to different target groups based on the URL in the request.
- Replace your ALB with a Classic Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request.
- Use host conditions to define rules that forward requests to different target groups based on the host name in the host header. This enables you to support multiple domains using a single load balancer.
- Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request.

Correct

You can use path conditions to define rules that forward requests to different target groups based on the URL in the request (also known as *path-based routing*). This type of routing is the most appropriate solution for this scenario hence, ***using path conditions to define rules that forward requests to different target groups based on the URL in the request*** is the correct answer.

Each path condition has one path pattern. If the URL in a request matches the path pattern in a listener rule exactly, the request is routed using that rule.

A path pattern is case-sensitive, can be up to 128 characters in length, and can contain any of the following characters. You can include up to three wildcard characters.

- A-Z, a-z, 0-9
- _ - . \$ / ~ ' @ : +
- & (using &)
- * (matches 0 or more characters)
- ? (matches exactly 1 character)

Example path patterns

- /img/*
- /js/*

The option that says: ***Use host conditions to define rules that forward requests to different target groups based on the host name in the host header. This enables you to support multiple domains using a single load balancer*** is incorrect because host-based routing defines rules that forward requests to different target groups based on the host name in the host header instead of the URL, which is what is needed in this scenario.

The option that says: ***Replace your ALB with a Classic Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request*** is incorrect because a Classic Load Balancer does not support path-based routing. You must use an Application Load Balancer.

The option that says: ***Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request*** is incorrect because a Network Load Balancer is used for applications that need extreme network performance and static IP. It also does not support path-based routing which is what is needed in this

scenario. Furthermore, the statement mentions host-based routing yet, the description is about path-based routing.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits>

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html#path-conditions>

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

<https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/>

6. QUESTION

Category: CSAA – Design Resilient Architectures

A company hosted a movie streaming app in Amazon Web Services. The application is deployed to several EC2 instances on multiple availability zones.

Which of the following configurations allows the load balancer to distribute incoming requests evenly to all EC2 instances across multiple Availability Zones?

- Elastic Load Balancing request routing
- An Amazon Route 53 weighted routing policy
- An Amazon Route 53 latency routing policy
- **Cross-zone load balancing**

Correct

The right answer is to enable ***cross-zone load balancing.***

If the load balancer nodes for your **Classic Load Balancer** can distribute requests regardless of Availability Zone, this is known as **cross-zone load balancing**. With cross-zone load balancing enabled, your load balancer nodes distribute incoming requests evenly across the Availability Zones enabled for your load balancer. Otherwise, each load balancer node distributes requests only to instances in its Availability Zone.

For example, if you have 10 instances in Availability Zone us-west-2a and 2 instances in us-west-2b, the requests are distributed evenly across all 12 instances if cross-zone load balancing is enabled. Otherwise, the 2 instances in us-west-2b serve the same number of requests as the 10 instances in us-west-2a.

Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.

However, we still recommend that you maintain approximately equivalent numbers of instances in each enabled Availability Zone for higher fault tolerance.

Reference:

<http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-disable-crosszone-lb.html>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

7. QUESTION

Category: CSAA – Design High-Performing Architectures

A company plans to design a highly available architecture in AWS. They have two target groups with three EC2 instances each, which are added to an Application Load Balancer. In the security group of the EC2 instance, you have verified that port 80 for HTTP is allowed. However, the instances are still showing out of service from the load balancer.

What could be the root cause of this issue?

- The instances are using the wrong AMI.
- **The health check configuration is not properly defined.**
- The wrong instance type was used for the EC2 instance.
- The wrong subnet was used in your VPC

Correct

Since the security group is properly configured, the issue may be caused by a wrong health check configuration in the Target Group.

Edit health check

X

Protocol i

HTTP ▼

Path i

/healthcheck

Advanced health check settings

Port i

traffic port

override

Healthy threshold i

2

Unhealthy threshold i

2

Timeout i

6 seconds

Interval i

30 seconds

Success codes i

200-399

[Cancel](#)

[Save](#)

Your **Application Load Balancer** periodically sends requests to its registered targets to test their status. These tests are called *health checks*. Each load balancer node routes requests only to the healthy targets in the enabled Availability Zones for the load balancer. Each load balancer node checks the health of each target, using the health check settings for the target group with which the target is registered. After your target is registered, it must pass one health check to be considered healthy. After each health check is completed, the load balancer node closes the connection that was established for the health check.

Reference:

<http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

ELB Health Checks vs Route 53 Health Checks For Target Health Monitoring:

<https://tutorialsdojo.com/elb-health-checks-vs-route-53-health-checks-for-target-health-monitoring/>

8. QUESTION

Category: CSAA – Design Secure Applications and Architectures

The social media company that you are working for needs to capture the detailed information of all HTTP requests that went through their public-facing application load balancer every five minutes. They want to use this data for analyzing traffic patterns and for troubleshooting their web applications in AWS.

Which of the following options meet the customer requirements?

- Add an Amazon CloudWatch Logs agent on the application load balancer.
- **Enable access logs on the application load balancer.**
- Enable AWS CloudTrail for their application load balancer.
- Enable Amazon CloudWatch metrics on the application load balancer.

Incorrect

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues.

Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Load Balancing captures the logs and stores them in the Amazon S3 bucket that you specify as compressed files. You can disable access logging at any time.

Reference:

<http://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html>

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer vs Gateway Load Balancer:

<https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/>

1. QUESTION

Category: CSAA – Design Resilient Architectures

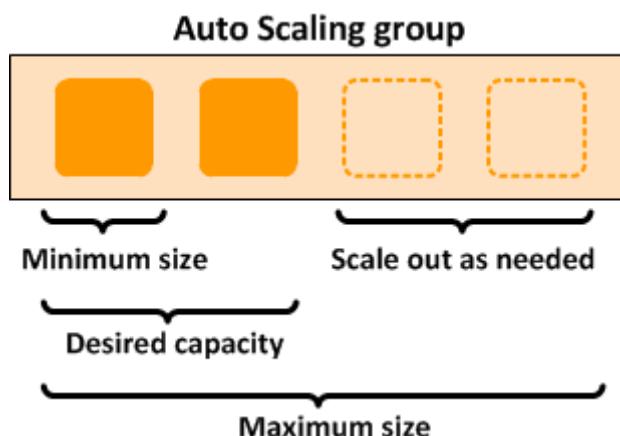
A company needs to deploy at least 2 EC2 instances to support the normal workloads of its application and automatically scale up to 6 EC2 instances to handle the peak load. The architecture must be highly available and fault-tolerant as it is processing mission-critical workloads.

As the Solutions Architect of the company, what should you do to meet the above requirement?

- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B.
- **Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.**

Incorrect

Amazon EC2 Auto Scaling helps ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can also specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.



To achieve highly available and fault-tolerant architecture for your applications, you must deploy all your instances in different Availability Zones. This will help you isolate your resources if an outage occurs. Take note that to achieve fault tolerance, you need to have redundant resources in place to avoid any system degradation in the event of a server fault or an Availability Zone outage. Having a fault-tolerant architecture entails an extra cost in running additional resources than what is usually needed. This is to ensure that the mission-critical workloads are processed.

Since the scenario requires at least 2 instances to handle regular traffic, you should have 2 instances running all the time even if an AZ outage occurred. You can use an Auto Scaling Group to automatically scale your compute resources across two or more Availability Zones. You have to specify the minimum capacity to 4 instances and the maximum capacity to 6 instances. If each AZ has 2 instances running, even if an AZ fails, your system will still run a minimum of 2 instances.

Hence, the correct answer in this scenario is: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.***

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A*** is incorrect because the instances are only deployed in a single Availability Zone. It cannot protect your applications and data from datacenter or AZ failures.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ*** is incorrect. It is required to have 2 instances running all the time. If an AZ outage happened, ASG will launch a new instance on the unaffected AZ. This provisioning does not happen instantly, which means that for a certain period of time, there will only be 1 running instance left.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B*** is incorrect. Although this fulfills the requirement of at least 2 EC2 instances and high availability, the maximum capacity setting is wrong. It should be set to 6 to properly handle the peak load. If an AZ outage occurs and the system is at its peak load, the number of running instances in this setup will only be 4 instead of 6 and this will affect the performance of your application.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

<https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

A company deployed a high-performance computing (HPC) cluster that spans multiple EC2 instances across multiple Availability Zones and processes various wind simulation models. Currently, the Solutions Architect is experiencing a slowdown in their applications and upon further investigation, it was discovered that it was due to latency issues.

Which is the MOST suitable solution that the Solutions Architect should implement to provide low-latency network performance necessary for tightly-coupled node-to-node communication of the HPC cluster?

- Use EC2 Dedicated Instances.
- Set up AWS Direct Connect connections across multiple Availability Zones for increased bandwidth throughput and more consistent network experience.
- Set up a spread placement group across multiple Availability Zones in multiple AWS Regions.
- **Set up a cluster placement group within a single Availability Zone in the same AWS Region.**

Incorrect

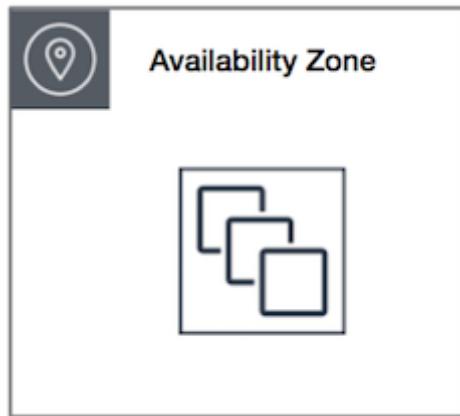
When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

Cluster – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.

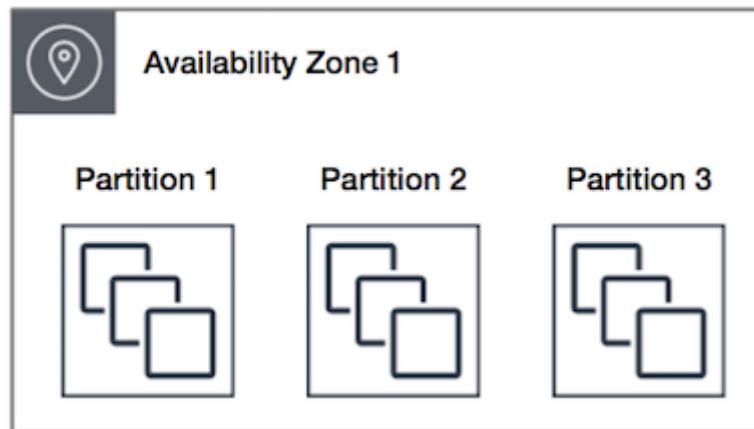
Partition – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.

Spread – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking.



Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.



Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time. A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.



Availability Zone 1



Hence, the correct answer is: ***Set up a cluster placement group within a single Availability Zone in the same AWS Region.***

The option that says: ***Set up a spread placement group across multiple Availability Zones in multiple AWS Regions*** is incorrect because although using a placement group is valid for this particular scenario, you can only set up a placement group in a **single** AWS Region only. A spread placement group can span multiple Availability Zones in the same Region.

The option that says: ***Set up AWS Direct Connect connections across multiple Availability Zones for increased bandwidth throughput and more consistent network experience*** is incorrect because this is primarily used for hybrid architectures. It bypasses the public Internet and establishes a secure, dedicated connection from your on-premises data center into AWS, and not used for having low latency within your AWS network.

The option that says: ***Use EC2 Dedicated Instances*** is incorrect because these are EC2 instances that run in a VPC on hardware that is dedicated to a single customer and are physically isolated at the host hardware level from instances that belong to other AWS accounts. It is not used for reducing latency.

References:

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

<https://aws.amazon.com/hpc/>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

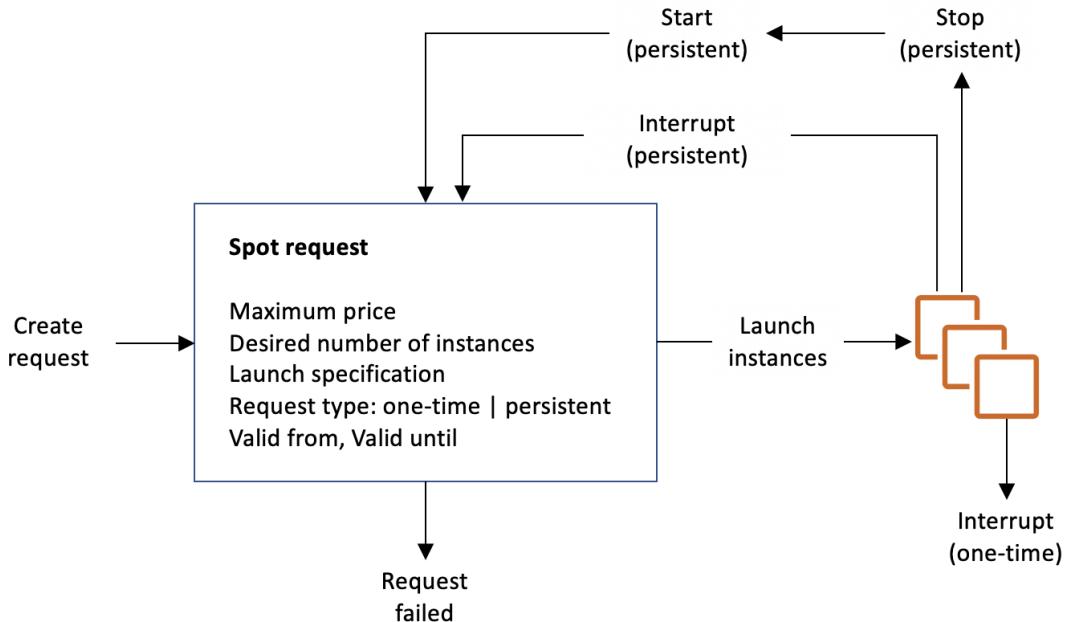
The media company that you are working for has a video transcoding application running on Amazon EC2. Each EC2 instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based on the queuing system. This application has a large backlog of videos which need to be transcoded. Your manager would like to reduce this backlog by adding more EC2 instances, however, these instances are only needed until the backlog is reduced.

In this scenario, which type of Amazon EC2 instance is the most cost-effective type to use?

- **Spot instances**
- Reserved instances
- Dedicated instances
- On-demand instances

Correct

You require an instance that will be used not as a primary server but as a spare compute resource to augment the transcoding process of your application. These instances should also be terminated once the backlog has been significantly reduced. In addition, the scenario mentions that if the current process is interrupted, the video can be transcoded by another instance based on the queuing system. This means that the application can gracefully handle an unexpected termination of an EC2 instance, like in the event of a Spot instance termination when the Spot price is greater than your set maximum price. Hence, an Amazon EC2 Spot instance is the best and cost-effective option for this scenario.



Amazon EC2 Spot instances are **spare** compute capacity in the AWS cloud available to you at **steep discounts compared to On-Demand prices**. EC2 Spot enables you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget. By simply selecting Spot when launching EC2 instances, you can save up-to 90% on On-Demand prices. **The only difference between On-Demand instances and Spot Instances is that Spot instances can be interrupted by EC2 with two minutes of notification when the EC2 needs the capacity back.**

You can specify whether Amazon EC2 should hibernate, stop, or terminate Spot Instances when they are interrupted. You can choose the interruption behavior that meets your needs.

Take note that there is no “*bid price*” anymore for Spot EC2 instances **since March 2018**. You simply have to set your **maximum price** instead.

Reserved instances and ***Dedicated instances*** are incorrect as both do not act as spare compute capacity.

On-demand instances is a valid option but a Spot instance is much cheaper than On-Demand.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html>

<https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

4. QUESTION

Category: CSAA – Design Resilient Architectures

A company has a cloud architecture that is composed of Linux and Windows EC2 instances that process high volumes of financial data 24 hours a day, 7 days a week. To ensure high availability of the systems, the Solutions Architect needs to create a solution that allows them to monitor the memory and disk utilization metrics of all the instances.

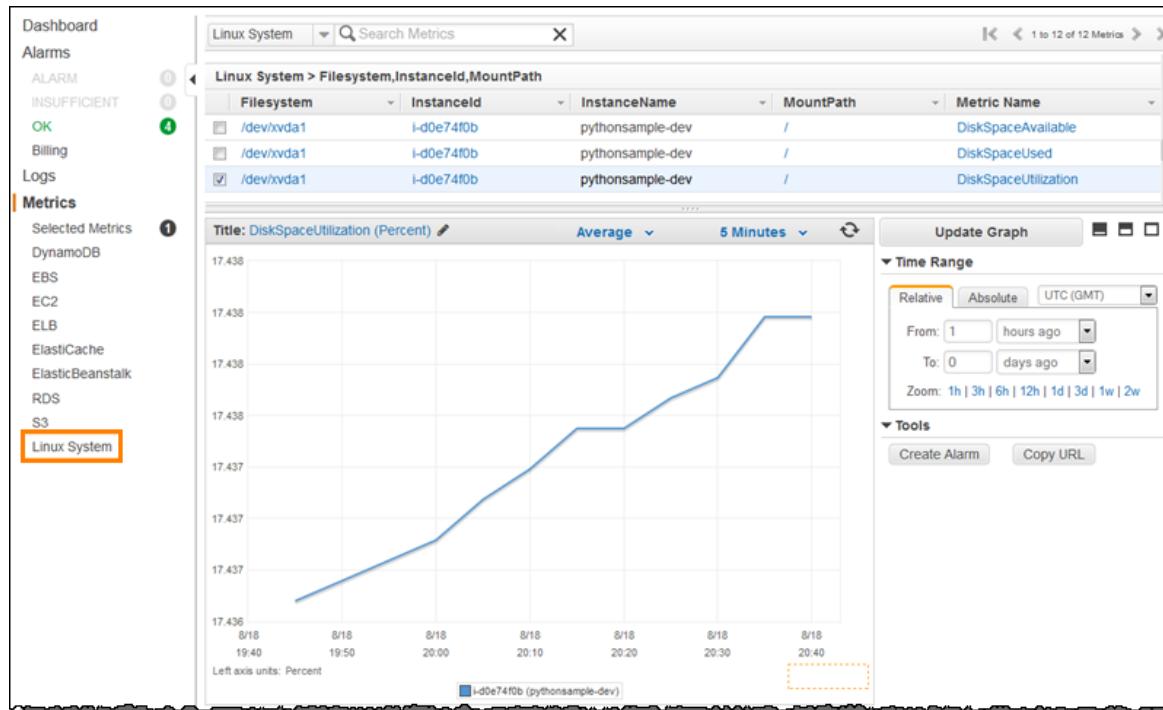
Which of the following is the most suitable monitoring solution to implement?

- Use Amazon Inspector and install the Inspector agent to all EC2 instances.
- **Install the CloudWatch agent to all the EC2 instances that gather the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.**
- Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard.
- Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances.

Correct

Amazon CloudWatch has available Amazon EC2 Metrics for you to use for monitoring CPU utilization, Network utilization, Disk performance, and Disk Reads/Writes. In case you need to monitor the below items, you need to prepare a custom metric using a Perl or other shell script, as there are no ready to use metrics for:

1. Memory utilization
2. Disk swap utilization
3. Disk space utilization
4. Page file utilization
5. Log collection



Take note that there is a multi-platform CloudWatch agent which can be installed on both Linux and Windows-based instances. You can use a single agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers. This agent supports both Windows Server and Linux and enables you to select the metrics to be collected, including sub-resource metrics such as per-CPU core. It is recommended that you use the new agent instead of the older monitoring scripts to collect metrics and logs.

Hence, the correct answer is: ***Install the CloudWatch agent to all the EC2 instances that gathers the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.***

The option that says: ***Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances*** is incorrect because, by default, CloudWatch does not automatically provide memory and disk utilization metrics of your instances. You have to set up custom CloudWatch metrics to monitor the memory, disk swap, disk space, and page file utilization of your instances.

The option that says: ***Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard*** is incorrect because Enhanced Monitoring is a feature of Amazon RDS. By default, Enhanced Monitoring metrics are stored for 30 days in the CloudWatch Logs.

The option that says: ***Use Amazon Inspector and install the Inspector agent to all EC2 instances*** is incorrect because Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. It does not provide a custom metric to track the memory and disk utilization of each and every EC2 instance in your VPC.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html#using_put_script

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

CloudWatch Agent vs SSM Agent vs Custom Daemon Scripts:

<https://tutorialsdojo.com/cloudwatch-agent-vs-ssm-agent-vs-custom-daemon-scripts/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

5. QUESTION

Category: CSAA – Design Resilient Architectures

You are automating the creation of EC2 instances in your VPC. Hence, you wrote a python script to trigger the Amazon EC2 API to request 50 EC2 instances in a single Availability Zone. However, you noticed that after 20 successful requests, subsequent requests failed.

What could be a reason for this issue and how would you resolve it?

- There was an issue with the Amazon EC2 API. Just resend the requests and these will be provisioned successfully.
- **There is a vCPU-based On-Demand Instance limit per region which is why subsequent requests failed. Just submit the limit increase form to AWS and retry the failed requests once approved.**
- By default, AWS allows you to provision a maximum of 20 instances per region. Select a different region and retry the failed request.
- By default, AWS allows you to provision a maximum of 20 instances per Availability Zone. Select a different Availability Zone and retry the failed request.

Incorrect

You are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region. New AWS accounts may start with limits that are lower than the limits described here.

aws Services Resource Groups 🔍

Tutorials Dojo Ohio Support

EC2 > Limits > Limits calculator

Calculate vCPU limit

Calculate number of vCPUs needed

Use this tool to calculate how many vCPUs you need to launch your On-Demand Instances

Select the instance type and the number of instances you require. The calculator will display the number of vCPUs assigned to the selected instances. Use the New Limit value as a guide for requesting a limit increase.

Instance type	Instance count	vCPU count	Current limit	New limit
<input type="text" value="t2.medium"/> X	<input type="text" value="12"/>	24 vCPUs	1,920 vCPUs	1,944 vCPUs X
Add instance type				

Limits calculation

Instance limit name	Current limit	vCPUs needed	New limit	Options
All Standard (A, C, D, H, I, M, R, T, Z) instances	1,920 vCPUs	24 vCPUs	1,944 vCPUs	Request limit increase

[Close](#)

Tutorials Dojo

If you need more instances, complete the Amazon EC2 limit increase request form with your use case, and your limit increase will be considered. Limit increases are tied to the region they were requested for.

Hence, the correct answer is: ***There is a vCPU-based On-Demand Instance limit per region which is why subsequent requests failed. Just submit the limit increase form to AWS and retry the failed requests once approved.***

The option that says: ***There was an issue with the Amazon EC2 API. Just resend the requests and these will be provisioned successfully*** is incorrect because you are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit. There is also a limit of purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region hence, there is no problem with the EC2 API.

The option that says: ***By default, AWS allows you to provision a maximum of 20 instances per region. Select a different region and retry the failed request*** is incorrect. There is no need to select a different region since this limit can be increased after submitting a request form to AWS.

The option that says: ***By default, AWS allows you to provision a maximum of 20 instances per Availability Zone. Select a different Availability Zone and retry the failed request*** is incorrect because the vCPU-based On-Demand Instance limit is set per region and not per Availability Zone. This can be increased after submitting a request form to AWS.

References:

https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html#limits_ec2

[https://aws.amazon.com/ec2/faqs/#How many instances can I run in Amazon EC2](https://aws.amazon.com/ec2/faqs/#How_many_instances_can_I_run_in_Amazon_EC2)

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

6. QUESTION

Category: CSAA – Design High-Performing Architectures

An organization needs to provision a new Amazon EC2 instance with a **persistent block storage volume** to migrate data from its on-premises network to AWS. The required maximum performance for the storage volume is 64,000 IOPS.

In this scenario, which of the following can be used to fulfill this requirement?

- Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance.
- Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.
- Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O.
- Launch any type of Amazon EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.

Incorrect

An **Amazon EBS volume** is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible.

The **AWS Nitro System** is the underlying platform for the latest generation of EC2 instances that enables AWS to innovate faster, further reduce the cost of the customers, and deliver added benefits like increased security and new instance types.

	Solid-state drives (SSD)		
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD	
		io2	io1
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	<ul style="list-style-type: none">Recommended for most workloadsSystem boot volumesVirtual desktopsLow-latency interactive appsDevelopment and test environments <ul style="list-style-type: none">Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volumeLarge database workloads, such as:<ul style="list-style-type: none">MongoDBCassandraMicrosoft SQL ServerMySQLPostgreSQLOracle		
Amazon EBS Multi-attach	Not supported	Not Supported	Supported
API name	gp2	io2	io1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	
Dominant performance attribute	IOPS	IOPS	
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	
Max throughput per volume	250 MiB/s *	1,000 MiB/s †	
Max IOPS per instance ‡‡	160,000		
Max throughput per instance ‡‡	4,750 MB/s		

Maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System provisioned with more than 32,000 IOPS.

Amazon EBS is a persistent block storage volume. It can persist independently from the life of an instance. Since the scenario requires you to have an EBS volume with up to 64,000 IOPS, you have to launch a Nitro-based EC2 instance.

Hence, the correct answer in this scenario is: ***Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.***

The option that says: ***Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance*** is incorrect. Although an Instance Store is a block storage volume, it is not persistent and the data will be gone if the instance is restarted from the stopped state (note that this is different from the OS-level reboot. In OS-level reboot, data still persists in the instance store). An instance store only provides temporary block-level storage for your instance. It means that the data in the instance store can be lost if the underlying disk drive fails, if the instance stops, and if the instance terminates.

The option that says: ***Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O*** is incorrect. Although Amazon EFS can provide over 64,000 IOPS, this solution uses a file system and not a block storage volume which is what is asked in the scenario.

The option that says: ***Launch an EC2 instance and attach an io1 EBS volume with 64,000 IOPS*** is incorrect. In order to achieve the 64,000 IOPS for a provisioned IOPS SSD, you must provision a Nitro-based EC2 instance. **The maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System provisioned with more than 32,000 IOPS. Other instances guarantee up to 32,000 IOPS only.**

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#EBSVolumeTypes_piops

<https://aws.amazon.com/s3/storage-classes/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

Amazon S3 vs EFS vs EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/>

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A multinational corporate and investment bank is regularly processing steady workloads of accruals, loan interests, and other critical financial calculations every night at 10 PM to 3 AM on their on-premises data center for their corporate clients. Once the process is done, the results are then uploaded to the Oracle General Ledger which means that the processing should not be delayed nor interrupted. The CTO has decided to move their IT infrastructure to AWS to save cost and to improve the scalability of their digital financial services.

As the Senior Solutions Architect, how can you implement a cost-effective architecture in AWS for their financial system?

- Use On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second.
- Use Spot EC2 Instances launched by a persistent Spot request, which can significantly lower your Amazon EC2 costs.
- **Use Scheduled Reserved Instances, which provide compute capacity that is always available on the specified recurring schedule.**
- Use Dedicated Hosts which provide a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.

Incorrect

Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time that the instances are scheduled, even if you do not use them.

The screenshot shows the AWS Scheduled Instances Reservation Wizard. In Step 1: Find a schedule, the user is creating a new schedule. They have set the starting date to Monday, January 4, 2016, at 16:00 UTC, for a duration of 8 hours. The schedule is set to recur weekly on Monday, Wednesday, and Friday. The instance details section specifies a Linux/UNIX (Amazon VPC) platform, a c3.4xlarge instance type, and any availability zone. A blue 'Find schedules' button is visible at the bottom.

Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.

Hence, the correct answer is to ***use Scheduled Reserved Instances, which provide compute capacity that is always available on the specified recurring schedule***.

Using On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second is incorrect because although an On-Demand instance is stable and suitable for processing critical data, **it costs more than any other option**. Moreover, the critical financial calculations are only done every night from 10 PM to 3 AM only and not 24/7. This means that **your compute capacity will not be utilized for a total of 19 hours every single day**.

Using Spot EC2 Instances launched by a persistent Spot request, which can significantly lower your Amazon EC2 costs is incorrect because although this is the most cost-effective solution, this type is not suitable for processing critical financial data since a Spot Instance has a risk of being interrupted.

Using Dedicated Hosts which provide a physical host that is fully dedicated to running your instances, and bringing your existing per-socket, per-core, or per-VM software licenses to reduce costs is incorrect because the use of a fully dedicated physical host is not warranted in this scenario. Moreover, this will be underutilized since you only run the process for 5 hours (from 10 PM to 3 AM only), wasting 19 hours of compute capacity every single day.

References:

<https://aws.amazon.com/blogs/aws/new-scheduled-reserved-instances/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-scheduled-instances.html>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

8. QUESTION

Category: CSAA – Design Resilient Architectures

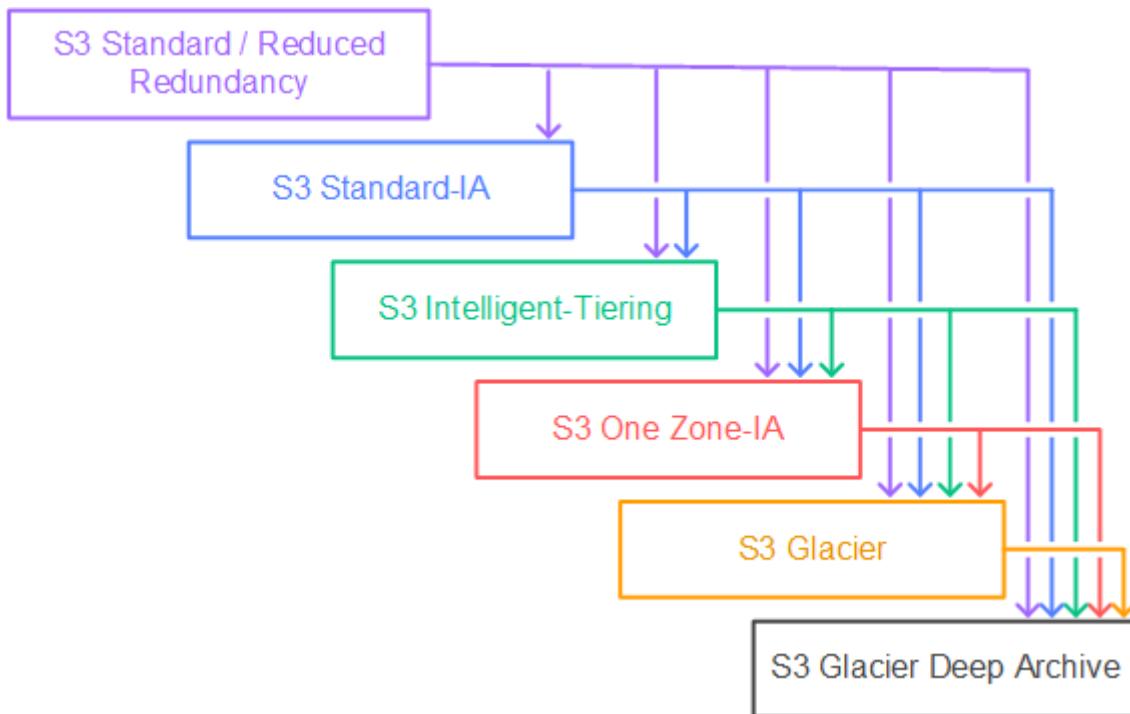
A company plans to host a web application in an Auto Scaling group of Amazon EC2 instances. The application will be used globally by users to upload and store several types of files. Based on user trends, files that are older than 2 years must be stored in a different storage class. The Solutions Architect of the company needs to create a cost-effective and scalable solution to store the old files yet still provide durability and high availability.

Which of the following approach can be used to fulfill this requirement? (Select TWO.)

- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.
- Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.
- Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years.
- Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.
- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.

Incorrect

Amazon S3 stores data as objects within buckets. An object is a file and any optional metadata that describes the file. To store a file in Amazon S3, you upload it to a bucket. When you upload a file as an object, you can set permissions on the object and any metadata. Buckets are containers for objects. You can have one or more buckets. You can control access for each bucket, deciding who can create, delete, and list objects in it. You can also choose the geographical region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.



To move a file to a different storage class, you can use **Amazon S3** or **Amazon EFS**. Both services have **lifecycle configurations**. Take note that **Amazon EFS** can only transition a file to the IA storage class after **90 days**. Since you need to move the files that are older than 2 years to a more cost-effective and scalable solution, you should use the Amazon S3 lifecycle configuration. With S3 lifecycle rules, you can transition files to S3 Standard IA or S3 Glacier. Using S3 Glacier expedited retrieval, you can quickly access your files within 1-5 minutes.

Hence, the correct answers are:

- *- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.*
- *- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.*

The option that says: ***Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years*** is incorrect because the maximum days for the EFS lifecycle policy is only 90 days. The requirement is to move the files that are older than 2 years or 730 days.

The option that says: ***Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because Amazon EBS costs more and is not as scalable as Amazon S3. It has some limitations when accessed by multiple EC2 instances. There are also huge costs involved in using the multi-attach feature on a Provisioned IOPS EBS volume to allow multiple EC2 instances to access the volume.

The option that says: ***Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because RAID (Redundant Array of Independent Disks) is just a data storage virtualization technology that combines multiple storage devices to achieve higher performance or data durability. RAID 0 can stripe multiple volumes together for greater I/O performance than you can achieve with a single volume. On the other hand, RAID 1 can mirror two volumes together to achieve on-instance redundancy.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

<https://docs.aws.amazon.com/efs/latest/ug/lifecycle-management-efs.html>

<https://aws.amazon.com/s3/faqs/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A company plans to migrate a MySQL database from an on-premises data center to the AWS Cloud. This database will be used by a legacy batch application that has steady-state workloads in the morning but has its peak load at night for the end-of-day processing. You need to choose an EBS volume that can handle a maximum of 450 GB of data and can also be used as the system boot volume for your EC2 instance.

Which of the following is the most cost-effective storage type to use in this scenario?

- Amazon EBS Throughput Optimized HDD (st1)
- Amazon EBS Provisioned IOPS SSD (io1)
- Amazon EBS Cold HDD (sc1)
- **Amazon EBS General Purpose SSD (gp2)**

Incorrect

In this scenario, a legacy batch application which has steady-state workloads requires a ***relational MySQL database***. The EBS volume that you should use has to handle a maximum of 450 GB of data and can also be used as the system ***boot volume*** for your EC2 instance. Since **HDD volumes cannot be used as a bootable volume**, we can narrow down our options by selecting SSD volumes. In addition, SSD volumes are more suitable for transactional database workloads, as shown in the table below:

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	<i>small, random</i> I/O operations	<i>large, sequential</i> I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none">- Best for transactional workloads- Critical business applications that require sustained IOPS performance- Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others...	<ul style="list-style-type: none">- Best for large streaming workloads requiring consistent, fast throughput at a low price- Big data, Data warehouses, Log processing- Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

General Purpose SSD (**gp2**) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. AWS designs **gp2** volumes to deliver the provisioned performance 99% of the time. A **gp2** volume can range in size from 1 GiB to 16 TiB.

Amazon EBS Provisioned IOPS SSD (io1) is incorrect because this is **not the most cost-effective EBS type** and is primarily used for critical business applications that require sustained IOPS performance.

Amazon EBS Throughput Optimized HDD (st1) is incorrect because this is primarily used for frequently accessed, throughput-intensive workloads. Although it is a low-cost HDD volume, it cannot be used as a system boot volume.

Amazon EBS Cold HDD (sc1) is incorrect. Although Amazon EBS Cold HDD provides lower cost HDD volume compared to General Purpose SSD, it cannot be used as a system boot volume.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#EBSVolumeTypes_gp2

Amazon EBS Overview – SSD vs HDD:

<https://youtu.be/LW7x8wyLFvw>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

2. QUESTION

Category: CSAA – Design Resilient Architectures

A company plans to migrate all of their applications to AWS. The Solutions Architect suggested to store all the data to EBS volumes. The Chief Technical Officer is worried that EBS volumes are not appropriate for the existing workloads due to compliance requirements, downtime scenarios, and IOPS performance.

Which of the following are valid points in proving that EBS is the best service to use for migration? (Select TWO.)

- When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component.
- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.
- EBS volumes can be attached to any EC2 Instance in any Availability Zone.
- Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones
- An EBS volume is off-instance storage that can persist independently from the life of an instance.

Incorrect

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Here is a list of important information about EBS Volumes:

- When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single hardware component.
- After you create a volume, you can attach it to any EC2 instance in the same Availability Zone
- Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1) volume to multiple Nitro-based instances that are in the same Availability Zone. However, other EBS types are not supported.
- An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.
- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.

- Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)
- EBS Volumes offer 99.999% SLA.

The option that says: ***When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component*** is incorrect because when you create an EBS volume in an Availability Zone, it is automatically replicated within that zone only, and not on a separate AWS region, to prevent data loss due to a failure of any single hardware component.

The option that says: ***EBS volumes can be attached to any EC2 Instance in any Availability Zone*** is incorrect as EBS volumes can only be attached to an EC2 instance in the same Availability Zone.

The option that says: ***Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones*** is almost correct. But instead of storing the volume to Amazon RDS, the EBS Volume snapshots are actually sent to Amazon S3.

References:

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

<https://aws.amazon.com/ebs/features/>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/>

Here is a short video tutorial on EBS:

<https://youtu.be/ljYH5HQdxo>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

A company plans to build a data analytics application in AWS which will be deployed in an Auto Scaling group of On-Demand EC2 instances and a MongoDB database. It is expected that the database will have high-throughput workloads performing small, random I/O operations. As the Solutions Architect, you are required to properly set up and launch the required resources in AWS.

Which of the following is the most suitable EBS type to use for your database?

- Cold HDD (sc1)
- General Purpose SSD (gp2)
- **Provisioned IOPS SSD (io1)**
- Throughput Optimized HDD (st1)

Correct

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes, such as General Purpose SSD (gp2) and Provisioned IOPS SSD (io1), deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes like Throughput Optimized HDD (st1) and Cold HDD (sc1) deliver optimal performance only when I/O operations are large and sequential.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has ***small, random*** I/O operations or ***large, sequential*** I/O operations.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

Provisioned IOPS SSD (`io1`) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike `gp2`, which uses a bucket and credit model to calculate performance, an `io1` volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

Volume Type	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
	General Purpose SSD (<code>gp2</code>)*	Provisioned IOPS SSD (<code>io1</code>)	Throughput Optimized HDD (<code>st1</code>)	Cold HDD (<code>sc1</code>)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none"> • Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: <ul style="list-style-type: none"> ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle 	<ul style="list-style-type: none"> • Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume 	<ul style="list-style-type: none"> • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume
API Name	<code>gp2</code>	<code>io1</code>	<code>st1</code>	<code>sc1</code>
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance††	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

General Purpose SSD (`gp2`) is incorrect because although General Purpose is a type of SSD that can handle small, random I/O operations, the Provisioned IOPS SSD volumes are much more suitable to meet the needs of I/O-intensive database workloads such as MongoDB, Oracle, MySQL, and many others.

Throughput Optimized HDD (`st1`) and ***Cold HDD (`sc1`)*** are incorrect because HDD volumes (such as Throughput Optimized HDD and Cold HDD volumes) are more suitable for workloads with large, sequential I/O operations instead of small, random I/O operations.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#EBSVolumeTypes_piops

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-io-characteristics.html>

Amazon EBS Overview - SSD vs HDD:

<https://youtu.be/LW7x8wyLFvw>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

4. QUESTION

Category: CSAA – Design Resilient Architectures

An application consists of multiple EC2 instances in private subnets in different availability zones. The application uses a single NAT Gateway for downloading software patches from the Internet to the instances. There is a requirement to protect the application from a single point of failure when the NAT Gateway encounters a failure or if its availability zone goes down.

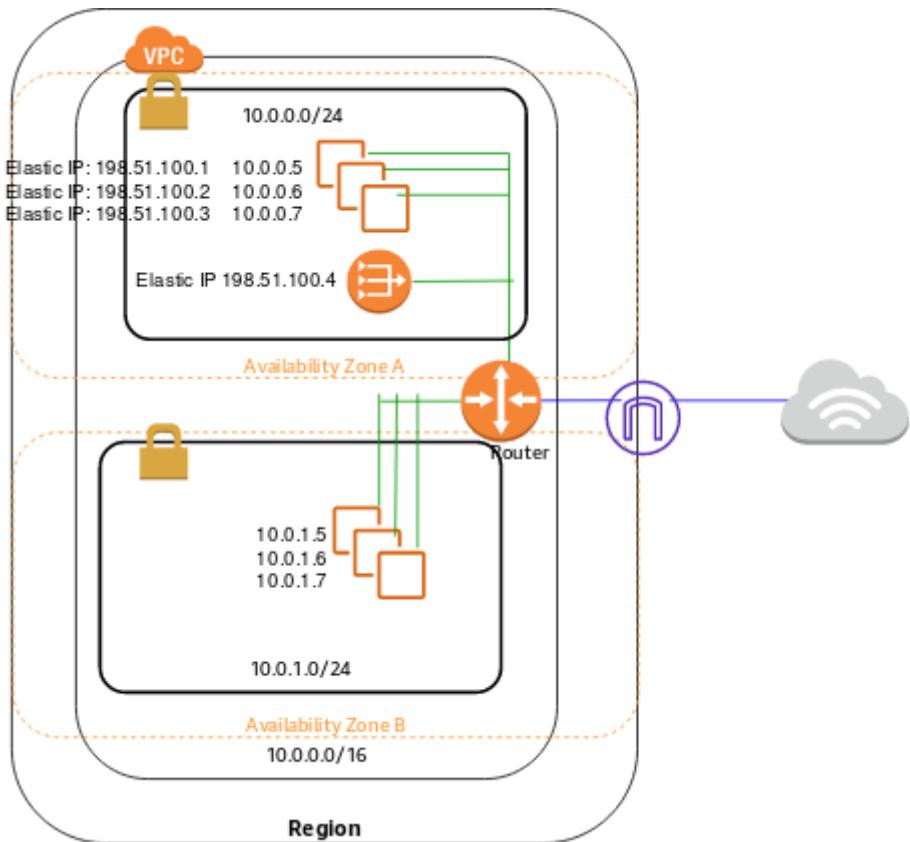
How should the Solutions Architect redesign the architecture to be more highly available and cost-effective?

- Create a NAT Gateway in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone
- Create a NAT Gateway in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone.
- Create two NAT Gateways in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone.
- Create three NAT Gateways in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone.

Correct

A **NAT Gateway** is a highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet. NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.

You must create a NAT gateway on a public subnet to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances.



If you have resources in multiple Availability Zones and they share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose Internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

Hence, the correct answer is: ***Create a NAT Gateway in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone***.

The option that says: ***Create a NAT Gateway in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone*** is incorrect because you should configure the route table in the private subnet and not the public subnet to associate the right instances in the private subnet.

The options that say: ***Create two NAT Gateways in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone*** and ***Create three NAT Gateways in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone*** are both incorrect because a single NAT Gateway in each availability zone is enough. NAT Gateway is already redundant in nature, meaning, AWS already handles any failures that occur in your NAT Gateway in an availability zone.

References:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

5. QUESTION

Category: CSAA – Design Resilient Architectures

A company is planning to launch an application which requires a data warehouse that will be used for their infrequently accessed data. You need to use an EBS Volume that can handle large, sequential I/O operations.

Which of the following is the most cost-effective storage type that you should use to meet the requirement?

- EBS General Purpose SSD (gp2)
- **Cold HDD (sc1)**
- Provisioned IOPS SSD (io1)
- Throughput Optimized HDD (st1)

Incorrect

Cold HDD volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than Throughput Optimized HDD, this is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, Cold HDD provides inexpensive block storage. Take note that bootable Cold HDD volumes are not supported.

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none"> • Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: <ul style="list-style-type: none"> ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle 	<ul style="list-style-type: none"> • Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume 	<ul style="list-style-type: none"> • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

Cold HDD provides the lowest cost HDD volume and is designed for less frequently accessed workloads. Hence, ***Cold HDD (sc1)*** is the correct answer.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has ***small, random*** I/O operations or ***large, sequential*** I/O operations.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

EBS General Purpose SSD (gp2) is incorrect because a General purpose SSD volume costs more and it is mainly used for a wide variety of workloads. It is recommended to be used as system boot volumes, virtual desktops, low-latency interactive apps, and many more.

Provisioned IOPS SSD (io1) is incorrect because this costs more than Cold HDD and thus, not cost-effective for this scenario. It provides the highest performance SSD volume for mission-critical low-latency or high-throughput workloads, which is not needed in the scenario.

Throughput Optimized HDD (st1) is incorrect because this is primarily used for **frequently** accessed, throughput-intensive workloads. In this scenario, Cold HDD perfectly fits the requirement as it is used for their infrequently accessed data and provides the lowest cost, unlike Throughput Optimized HDD.

References:

<https://aws.amazon.com/ebs/details/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

6. QUESTION

Category: CSAA – Design Resilient Architectures

As part of the Business Continuity Plan of your company, your IT Director instructed you to set up an automated backup of all of the EBS Volumes for your EC2 instances as soon as possible.

What is the fastest and most cost-effective solution to automatically back up all of your EBS Volumes?

- Use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots.
- Use an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes.
- Set your Amazon Storage Gateway with EBS volumes as the data source and store the backups in your on-premises servers through the storage gateway.

- For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically.

Correct

You can use **Amazon Data Lifecycle Manager (Amazon DLM)** to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes. Automating snapshot management helps you to:

- Protect valuable data by enforcing a regular backup schedule.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.

Combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon DLM provides a complete backup solution for EBS volumes at no additional cost.

Hence, ***using Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots*** is the correct answer as it is the fastest and most cost-effective solution that provides an automated way of backing up your EBS volumes.

The option that says: ***For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically*** is incorrect because even though this is a valid solution, you would still need additional time to create a scheduled job that calls the "create-snapshot" command. It would be better to use Amazon Data Lifecycle Manager (Amazon DLM) instead as this provides you the fastest solution which enables you to automate the creation, retention, and deletion of the EBS snapshots without having to write custom shell scripts or creating scheduled jobs.

Setting your Amazon Storage Gateway with EBS volumes as the data source and storing the backups in your on-premises servers through the storage gateway is incorrect as the **Amazon Storage Gateway is used only for creating a backup of data from your on-premises server** and not from the Amazon Virtual Private Cloud.

Using an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes is incorrect as there is no such thing as EBS-cycle policy in Amazon S3.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

Amazon EBS Overview – SSD vs HDD:

<https://youtu.be/LW7x8wyLFvw>

7. QUESTION

Category: CSAA – Design High-Performing Architectures

A global online sports betting company has its popular web application hosted in AWS. They are planning to develop a new online portal for their new business venture and they hired you to implement the cloud architecture for a new online portal that will accept bets globally for world sports. You started to design the system with a relational database that runs on a single EC2 instance, which requires a single EBS volume that can support up to 30,000 IOPS.

In this scenario, which Amazon EBS volume type can you use that will meet the performance requirements of this new online portal?

- EBS Throughput Optimized HDD (st1)
- **EBS Provisioned IOPS SSD (io1)**
- EBS Cold HDD (sc1)
- EBS General Purpose SSD (gp2)

Correct

The scenario requires a storage type for a relational database with a high IOPS performance. For these scenarios, SSD volumes are more suitable to use instead of HDD volumes. Remember that the dominant performance attribute of SSD is **IOPS** while HDD is **Throughput**.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has ***small, random*** I/O operations or ***large, sequential*** I/O operations.

Since the requirement is 30,000 IOPS, you have to use an EBS type of Provisioned IOPS SSD. This provides sustained performance for mission-critical low-latency workloads. Hence, ***EBS Provisioned IOPS SSD (io1)*** is the correct answer.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

EBS Throughput Optimized HDD (st1) and ***EBS Cold HDD (sc1)*** are incorrect because these are HDD volumes which are more suitable for large streaming workloads rather than transactional database workloads.

EBS General Purpose SSD (gp2) is incorrect because although a General Purpose SSD volume can be used for this scenario, it does not provide the high IOPS required by the application, unlike the Provisioned IOPS SSD volume.

Reference:

<https://aws.amazon.com/ebs/details/>

Amazon EBS Overview – SSD vs HDD:

<https://youtu.be/LW7x8wyLFvw>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

8. QUESTION

Category: CSAA – Design High-Performing Architectures

A company is building an internal application that processes loans, accruals, and interest rates for their clients. They require a storage service that is able to handle future increases in storage capacity of up to 16 TB and can provide the lowest-latency access to their data. The web application will be hosted in a single m5ad.24xlarge Reserved EC2 instance that will process and store data to the storage service.

Which of the following storage services would you recommend?

- **EBS**
- Storage Gateway
- S3
- EFS

Correct

Amazon Web Services (AWS) offers cloud storage services to support a wide range of storage workloads such as Amazon S3, EFS and EBS. **Amazon EFS is a file storage service for use with Amazon EC2.** Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for up to thousands of Amazon EC2 instances. **Amazon S3 is an object storage service.** Amazon S3 makes data available through an Internet API that can be accessed anywhere. **Amazon EBS is a block-level storage service for use with Amazon EC2.** Amazon EBS can deliver performance for workloads that require the **lowest-latency access to data** from a single EC2 instance. You can also increase EBS storage for up to 16TB or add new volumes for additional storage.

In this scenario, the company is looking for a storage service which can provide the lowest-latency access to their data which will be fetched by a single m5ad.24xlarge Reserved EC2 instance. This type of workloads can be supported better by using either EFS or EBS but in this case, the latter is the most suitable storage service. As mentioned above, EBS provides the *lowest-latency* access to the data for your EC2 instance since the volume is directly attached to the instance. In addition, the scenario does not require concurrently-accessible storage since they only have one instance.

Hence, the correct answer is ***EBS***.

Storage Need	Solution	AWS Services
Temporary storage	Consider using local instance store volumes for needs such as scratch disks, buffers, queues, and caches.	Amazon Local Instance Store
Multi-instance storage	Amazon EBS volumes can only be attached to one EC2 instance at a time. If you need multiple EC2 instances accessing volume data at the same time, consider using Amazon EFS as a file system.	Amazon EFS
Highly durable storage	If you need very highly durable storage, use S3 or Amazon EFS. Amazon S3 Standard storage is designed for 99.999999999 percent (11 nines) annual durability per object. You can even decide to take a snapshot of the EBS volumes. Such a snapshot then gets saved in Amazon S3, thus providing you the durability of Amazon S3. For more information on EBS durability, see the Durability and Availability section. EFS is designed for high durability and high availability, with data stored in multiple Availability Zones within an AWS Region.	Amazon S3 Amazon EFS
Static data or web content	If your data doesn't change that often, Amazon S3 might represent a more cost-effective and scalable solution for storing this fixed information. Also, web content served out of Amazon EBS requires a web server running on Amazon EC2; in contrast, you can deliver web content directly out of Amazon S3 or from multiple EC2 instances using Amazon EFS.	Amazon S3 Amazon EFS

Storage Gateway is incorrect since this is primarily used to extend your on-premises storage to your AWS Cloud.

S3 is incorrect because although this is also highly available and highly scalable, it still does not provide the lowest-latency access to the data, unlike EBS. Remember that S3 does not reside within your VPC by default, which means the data will traverse the public Internet that may result to higher latency. You can set up a VPC Endpoint for S3 yet still, its latency is greater than that of EBS.

EFS is incorrect because the scenario does not require concurrently-accessible storage since the internal application is only hosted in one instance. Although EFS can provide low latency data access to the EC2 instance as compared with S3, the storage service that can provide the lowest latency access is still EBS.

References:

<https://aws.amazon.com/ebs/>

<https://aws.amazon.com/efs/faq/>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A content management system (CMS) is hosted on a fleet of auto-scaled, On-Demand EC2 instances that use Amazon Aurora as its database. Currently, the system stores the file documents that the users upload in one of the attached EBS Volumes. Your manager noticed that the system performance is quite slow and he has instructed you to improve the architecture of the system.

In this scenario, what will you do to implement a scalable, high-available POSIX-compliant shared file system?

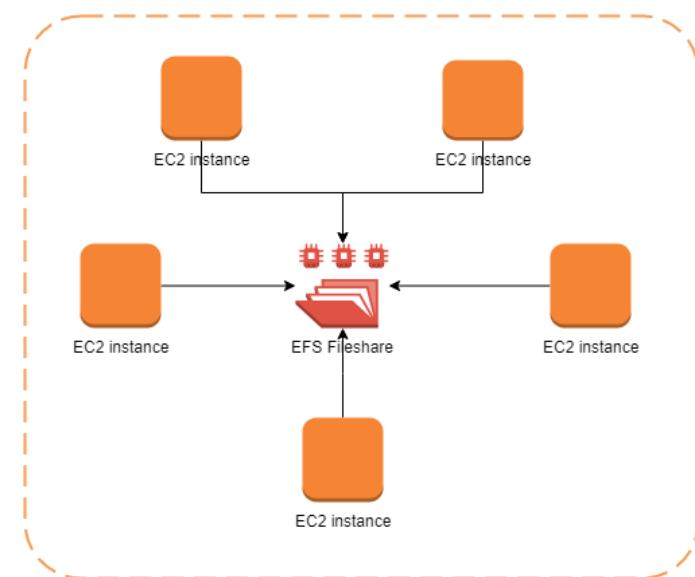
- Create an S3 bucket and use this as the storage for the CMS
- **Use EFS**
- Use ElastiCache
- Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes

Correct

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. When mounted on Amazon EC2 instances, an Amazon EFS file system provides a standard file system interface and file system access semantics, allowing you to seamlessly integrate Amazon EFS with your existing applications and tools. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, allowing Amazon EFS to provide a common data source for workloads and applications running on more than one Amazon EC2 instance.

This particular scenario tests your understanding of EBS, EFS, and S3. In this scenario, there is a fleet of On-Demand EC2 instances that store file documents from the users to one of the attached EBS Volumes. The system performance is quite slow because the architecture doesn't provide the EC2 instances parallel shared access to the file documents.

Although an EBS Volume can be attached to multiple EC2 instances, you can only do so on instances within an availability zone. What we need is high-available storage that can span multiple availability zones. Take note as well that the type of storage needed here is "file storage" which means that ***S3*** is not the best service to use because it is mainly used for "object storage", and S3 does not provide the notion of "folders" too. This is why ***using EFS*** is the correct answer.



Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes is incorrect because an EBS volume is a storage area network (SAN) storage and not a POSIX-compliant shared file system. You have to use EFS instead.

Using ElastiCache is incorrect because this is an in-memory data store that improves the performance of your applications, which is not what you need since it is not a file storage.

Reference:

<https://aws.amazon.com/efs/>

Check out this Amazon EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-efs/>

Check out this Amazon S3 vs EBS vs EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

A data analytics company has been building its new generation big data and analytics platform on their AWS cloud infrastructure. They need a storage service that provides the scale and performance that their big data applications require such as high throughput to compute nodes coupled with read-after-write consistency and low-latency file operations. In addition, their data needs to be stored redundantly across multiple AZs and allows concurrent connections from multiple EC2 instances hosted on multiple AZs.

Which of the following AWS storage services will you use to meet this requirement?

- Glacier
- S3
- EBS
- **EFS**

Correct

In this question, you should take note of the two keywords/phrases: “file operation” and “allows concurrent connections from multiple EC2 instances”. There are various AWS storage options that you can choose but whenever these criteria show up, always consider using EFS instead of using EBS Volumes which is mainly used as a “block” storage and can only have one connection to one EC2 instance at a time. Amazon EFS provides the scale and performance required for big data applications that require high throughput to compute nodes coupled with read-after-write consistency and low-latency file operations.

Amazon EFS is a fully-managed service that makes it easy to set up and scale file storage in the Amazon Cloud. With a few clicks in the AWS Management Console, you can create file systems that are accessible to Amazon EC2 instances via a file system interface (using standard operating system file I/O APIs) and supports full file system access semantics (such as strong consistency and file locking).

Amazon EFS file systems can automatically scale from gigabytes to petabytes of data without needing to provision storage. Tens, hundreds, or even thousands of Amazon EC2 instances can access an Amazon EFS file system at the same time, and Amazon EFS provides consistent performance to each Amazon EC2 instance. Amazon EFS is designed to be highly durable and highly available.

EBS is incorrect because it does not allow concurrent connections from multiple EC2 instances hosted on multiple AZs and it does not store data redundantly across multiple AZs by default, unlike EFS.

S3 is incorrect because although it can handle concurrent connections from multiple EC2 instances, it does not have the ability to provide low-latency file operations, which is required in this scenario.

Glacier is incorrect because this is an archiving storage solution and is not applicable in this scenario.

References:

<https://docs.aws.amazon.com/efs/latest/ug/performance.html>

<https://aws.amazon.com/efs/faq/>

Check out this Amazon EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-efs/>

Check out this Amazon S3 vs EBS vs EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/>

Here's a short video tutorial on Amazon EFS:

<https://youtu.be/AvgAozsfCrY>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

A leading e-commerce company is in need of a storage solution that can be simultaneously accessed by 1000 Linux servers in multiple availability zones. The servers are hosted in EC2 instances that use a hierarchical directory structure via the NFSv4 protocol. The service should be able to handle the rapidly changing data at scale while still maintaining high performance. It should also be highly durable and highly available whenever the servers will pull data from it, with little need for management.

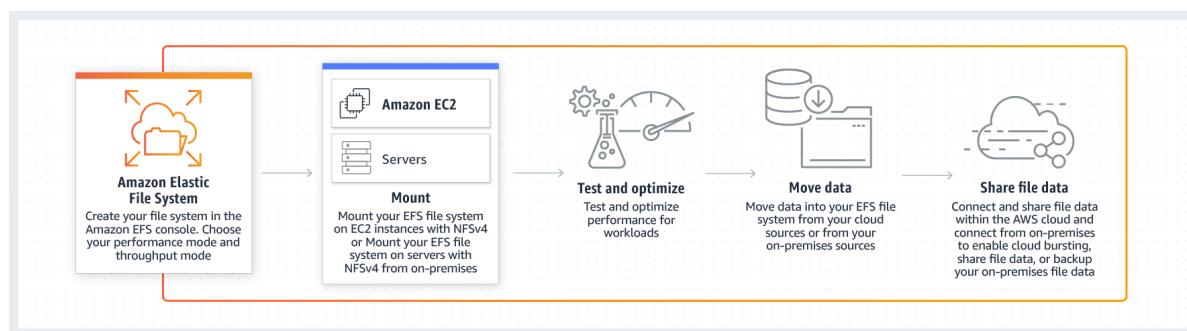
As the Solutions Architect, which of the following services is the most cost-effective choice that you should use to meet the above requirement?

- S3
- **EFS**
- Storage Gateway
- EBS

Correct

Amazon Web Services (AWS) offers cloud storage services to support a wide range of storage workloads such as EFS, S3 and EBS. You have to understand when you should use Amazon EFS, Amazon S3 and Amazon Elastic Block Store (EBS) based on the specific workloads. In this scenario, the keywords are **rapidly changing data** and 1000 Linux servers.

Amazon EFS is a file storage service for use with Amazon EC2. Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for **up to thousands of Amazon EC2 instances**. EFS provides the same level of high availability and high scalability like S3 however, this service is more suitable for scenarios where it is required to have a POSIX-compatible file system or if you are storing rapidly changing data.



Data that must be updated very frequently might be better served by storage solutions that take into account read and write latencies, such as Amazon EBS volumes, Amazon RDS, Amazon DynamoDB, Amazon EFS, or relational databases running on Amazon EC2.

Amazon EBS is a block-level storage service for use with Amazon EC2. Amazon EBS can deliver performance for workloads that require the lowest-latency access to data from a single EC2 instance.

Amazon S3 is an object storage service. Amazon S3 makes data available through an Internet API that can be accessed anywhere.

In this scenario, ***EFS*** is the best answer. As stated above, Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for **up to thousands of Amazon EC2 instances**. EFS provides the performance, durability, high availability, and storage capacity needed by the 1000 Linux servers in the scenario.

S3 is incorrect because although this provides the same level of high availability and high scalability like EFS, this service is not suitable for storing data which are rapidly changing, just as mentioned in the above explanation. It is still more effective to use EFS as it offers strong consistency and file locking which the S3 service lacks.

EBS is incorrect because an EBS Volume cannot be shared by multiple instances.

Storage Gateway is incorrect because this is primarily used to extend the storage of your on-premises data center to your AWS Cloud.

References:

<https://docs.aws.amazon.com/efs/latest/ug/how-it-works.html>

<https://aws.amazon.com/efs/features/>

<https://d1.awsstatic.com/whitepapers/AWS%20Storage%20Services%20Whitepaper-v9.pdf#page=9>

Check out this Amazon EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-efs/>

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A multinational company has been building its new data analytics platform with high-performance computing workloads (HPC) which requires a scalable, POSIX-compliant storage service. The data need to be stored redundantly across multiple AZs and allows concurrent connections from thousands of EC2 instances hosted on multiple Availability Zones.

Which of the following AWS storage service is the most suitable one to use in this scenario?

- **Amazon Elastic File System**
- Amazon ElastiCache
- Amazon EBS Volumes
- Amazon S3

Correct

In this question, you should take note of this phrase: “allows concurrent connections from multiple EC2 instances”. There are various AWS storage options that you can choose but whenever these criteria show up, always consider using EFS instead of using **EBS Volumes which is mainly used as a “block” storage and can only have one connection to one EC2 instance at a time.**

Amazon EFS is a fully-managed service that makes it easy to set up and scale file storage in the Amazon Cloud. With a few clicks in the AWS Management Console, you can create file systems that are accessible to Amazon EC2 instances via a file system interface (using standard operating system file I/O APIs) and supports full file system access semantics (such as strong consistency and file locking).

Amazon EFS file systems can automatically scale from gigabytes to petabytes of data without needing to provision storage. Tens, hundreds, or even thousands of Amazon EC2 instances can access an Amazon EFS file system at the same time, and Amazon EFS provides consistent performance to each Amazon EC2 instance. Amazon EFS is designed to be highly durable and highly available.

References:

<https://docs.aws.amazon.com/efs/latest/ug/performance.html>

<https://aws.amazon.com/efs/faq/>

Check out this Amazon EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-efs/>

Here's a short video tutorial on Amazon EFS:

<https://youtu.be/AvgAozsfCrY>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company has an On-Demand EC2 instance located in a subnet in AWS that hosts a web application. The security group attached to this EC2 instance has the following Inbound Rules:

The screenshot shows the AWS Security Groups console. At the top, it displays the security group name "sg-a282cf6", its owner "TutorialsDojo", and its ID "vpc-f2bf5897". Below this, there are tabs for "Description", "Inbound" (which is selected), "Outbound", and "Tags". An "Edit" button is visible. The main table lists one inbound rule: Type SSH, Protocol TCP, Port Range 22, and Source 0.0.0.0/0.

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	0.0.0.0/0	

The Route table attached to the VPC is shown below. You can establish an SSH connection into the EC2 instance from the Internet. However, you are not able to connect to the web server using your Chrome browser.

The screenshot shows the AWS Route Tables console. It displays a route table named "rtb-46b1813b" associated with "TutorialsDojo". The table has 0 subnets and is set to Yes for propagation. Below the table, there are tabs for "Summary", "Routes" (which is selected), "Subnet Associations", "Route Propagation", and "Tags". An "Edit" button is visible. The "Routes" table shows two entries: one for destination 10.0.0.0/27 targetting "local" with status "Active" and propagation "No"; and another for destination 0.0.0.0/0 targetting "igw-b51618cc" with status "Active" and propagation "No".

Destination	Target	Status	Propagated
10.0.0.0/27	local	Active	No
0.0.0.0/0	igw-b51618cc	Active	No

Which of the below steps would resolve the issue?

- In the Route table, add this new route entry: 0.0.0.0 -> igw-b51618cc
- In the Security Group, remove the SSH rule.
- In the Security Group, add an Inbound HTTP rule.**
- In the Route table, add this new route entry: 10.0.0.0/27 -> local

Correct

In this particular scenario, you can already connect to the EC2 instance via SSH. This means that there is no problem in the Route Table of your VPC. To fix this issue, you simply need to update your Security Group and add an Inbound rule to allow HTTP traffic.

Create Security Group

Security group name	Web Server Security Group
Description	Security for production web server.
VPC	vpc-e68d9c81 DefaultVPC (default)

Security group rules:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Anywhere	Admin access.
HTTP	TCP	80	Anywhere	Web traffic.
HTTPS	TCP	443	Custom	Secure web traffic.

Add Rule

Cancel **Create**

The option that says: ***In the Security Group, remove the SSH rule*** is incorrect as doing so will not solve the issue. It will just disable SSH traffic that is already available.

The options that say: ***In the Route table, add this new route entry: 0.0.0.0 -> igw-b51618cc*** and ***In the Route table, add this new route entry: 10.0.0.0/27 -> local*** are incorrect as there is no need to change the Route Tables.

Reference:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A local bank has an in-house application that handles sensitive financial data in a private subnet. After the data is processed by the EC2 worker instances, they will be delivered to S3 for ingestion by other services.

How should you design this solution so that the data does not pass through the public Internet?

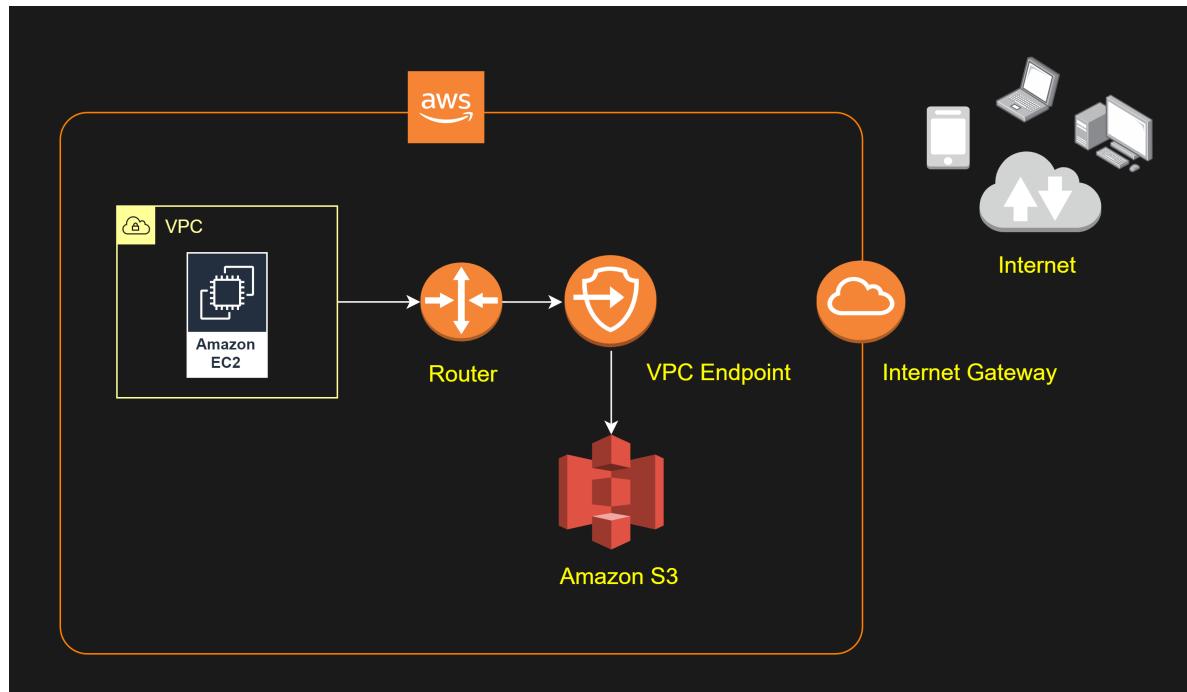
- Provision a NAT gateway in the private subnet with a corresponding route entry that directs the data to S3.
- Configure a VPC Endpoint along with a corresponding route entry that directs the data to S3.**
- Configure a Transit gateway along with a corresponding route entry that directs the data to S3.
- Create an Internet gateway in the public subnet with a corresponding route entry that directs the data to S3.

Incorrect

The important concept that you have to understand in this scenario is that your VPC and your S3 bucket are located within the larger AWS network. However, the traffic coming from your VPC to your S3 bucket is traversing the public Internet by default. **To better protect your data in transit, you can set up a VPC endpoint so the incoming traffic from your VPC will not pass through the public Internet, but instead through the private AWS network.**

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other services does not leave the Amazon network.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.



Hence, the correct answer is: ***Configure a VPC Endpoint along with a corresponding route entry that directs the data to S3.***

The option that says: ***Create an Internet gateway in the public subnet with a corresponding route entry that directs the data to S3*** is incorrect because the Internet gateway is used for instances in the public subnet to have accessibility to the Internet.

The option that says: ***Configure a Transit gateway along with a corresponding route entry that directs the data to S3*** is incorrect because the **Transit Gateway is used for interconnecting VPCs and on-premises networks through a central hub.** Since Amazon S3 is outside of VPC, you still won't be able to connect to it privately.

The option that says: ***Provision a NAT gateway in the private subnet with a corresponding route entry that directs the data to S3*** is incorrect because **NAT Gateway allows instances in the private subnet to gain access to the Internet, but not vice versa.**

References:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

3. QUESTION

Category: CSAA – Design Resilient Architectures

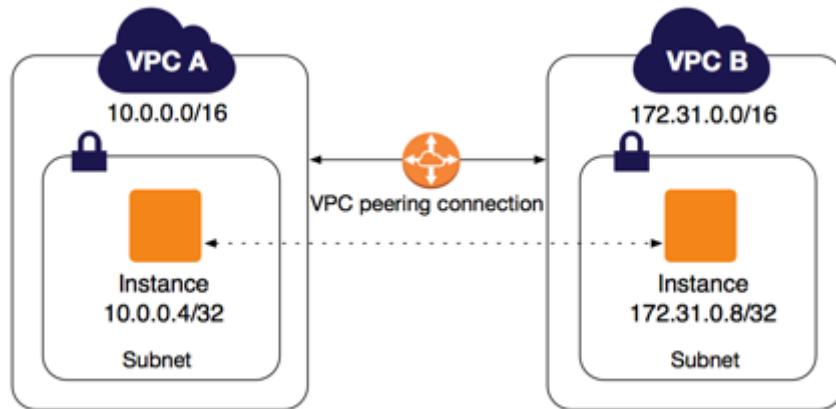
A large insurance company has an AWS account that contains three VPCs (DEV, UAT and PROD) in the same region. UAT is peered to both PROD and DEV using a VPC peering connection. All VPCs have non-overlapping CIDR blocks. The company wants to push minor code releases from Dev to Prod to speed up time to market.

Which of the following options helps the company accomplish this?

- Create a new entry to PROD in the DEV route table using the VPC peering connection as the target.
- **Create a new VPC peering connection between PROD and DEV with the appropriate routes.**
- Do nothing. Since these two VPCs are already connected via UAT, they already have a connection to each other.
- Change the DEV and PROD VPCs to have overlapping CIDR blocks to be able to connect them.

Correct

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.



AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

Creating a new entry to PROD in the DEV route table using the VPC peering connection as the target is incorrect because even if you configure the route tables, the two VPCs will still be disconnected until you set up a VPC peering connection between them.

Changing the DEV and PROD VPCs to have overlapping CIDR blocks to be able to connect them is incorrect because you cannot peer two VPCs with overlapping CIDR blocks.

The option that says: ***Do nothing. Since these two VPCs are already connected via UAT, they already have a connection to each other*** is incorrect as transitive VPC peering is not allowed hence, even though DEV and PROD are both connected in UAT, these two VPCs do not have a direct connection to each other.

Reference:

<https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-peering.html>

Check out these Amazon VPC and VPC Peering Cheat Sheets:

<https://tutorialsdojo.com/amazon-vpc/>

<https://tutorialsdojo.com/vpc-peering/>

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company hosted a web application on a Linux Amazon EC2 instance in the public subnet that uses a default network ACL. The instance uses a default security group and has an attached Elastic IP address. The network ACL has been configured to block all traffic to the instance. The Solutions Architect must allow incoming traffic on port 443 to access the application from any source.

Which combination of steps will accomplish this requirement? (Select TWO.)

- In the Security Group, add a new rule to allow TCP connection on port 443 from source 0.0.0.0/0
- In the Network ACL, update the rule to allow both inbound and outbound TCP connection on port 443 from source 0.0.0.0/0 and to destination 0.0.0.0/0
- In the Security Group, create a new rule to allow TCP connection on port 443 to destination 0.0.0.0/0
- In the Network ACL, update the rule to allow outbound TCP connection on port 32768 - 65535 to destination 0.0.0.0/0
- In the Network ACL, update the rule to allow inbound TCP connection on port 443 from source 0.0.0.0/0 and outbound TCP connection on port 32768 - 65535 to destination 0.0.0.0/0

Incorrect

To enable the connection to a service running on an instance, the associated network ACL must allow both inbound traffic on the port that the service is listening on as well as allow outbound traffic from ephemeral ports. When a client connects to a server, a random port from the ephemeral port range (1024-65535) becomes the client's source port.

The designated ephemeral port then becomes the destination port for return traffic from the service, so outbound traffic from the ephemeral port must be allowed in the network ACL. By default, network ACLs allow all inbound and outbound traffic. If your network ACL is more restrictive, then you need to explicitly allow traffic from the ephemeral port range.

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
1	HTTPS (443)	TCP (6)	443	0.0.0.0/0	Allow
2	Custom TCP	TCP (6)	32768 - 65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

The client that initiates the request chooses the ephemeral port range. The range varies depending on the client's operating system.

- Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000.
- Requests originating from Elastic Load Balancing use ports 1024-65535.
- Windows operating systems through Windows Server 2003 use ports 1025-5000.

- Windows Server 2008 and later versions use ports 49152-65535.
- A NAT gateway uses ports 1024-65535.
- AWS Lambda functions use ports 1024-65535.

For example, if a request comes into a web server in your VPC from a Windows 10 client on the Internet, your network ACL must have an outbound rule to enable traffic destined for ports 49152 – 65535. If an instance in your VPC is the client initiating a request, your network ACL must have an inbound rule to enable traffic destined for the ephemeral ports specific to the type of instance (Amazon Linux, Windows Server 2008, and so on).

In this scenario, you only need to allow the incoming traffic on port 443. Since security groups are stateful, you can apply any changes to an incoming rule and it will be automatically applied to the outgoing rule.

To enable the connection to a service running on an instance, the associated network ACL must allow both inbound traffic on the port that the service is listening on as well as allow outbound traffic from ephemeral ports. When a client connects to a server, a random port from the ephemeral port range (32768 – 65535) becomes the client's source port.

Hence, the correct answers are:

- *- In the Security Group, add a new rule to allow TCP connection on port 443 from source 0.0.0.0/0.***
- *- In the Network ACL, update the rule to allow inbound TCP connection on port 443 from source 0.0.0.0/0 and outbound TCP connection on port 32768 - 65535 to destination 0.0.0.0/0.***

The option that says: ***In the Security Group, create a new rule to allow TCP connection on port 443 to destination 0.0.0.0/0*** is incorrect because this step just allows outbound connections from the EC2 instance out to the public Internet which is unnecessary. Remember that **a default security group already includes an outbound rule that allows all outbound traffic.**

The option that says: ***In the Network ACL, update the rule to allow both inbound and outbound TCP connection on port 443 from source 0.0.0.0/0 and to destination 0.0.0.0/0*** is incorrect because your network ACL must have an outbound rule to allow ephemeral ports (32768 - 65535). These are the specific ports that will be used as the client's source port for the traffic response.

The option that says: ***In the Network ACL, update the rule to allow outbound TCP connection on port 32768 - 65535 to destination 0.0.0.0/0*** is incorrect because this step is just partially right. You still need to add an inbound rule from port 443 and not just the outbound rule for the ephemeral ports (32768 - 65535).

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/connect-http-https-ec2/>

https://docs.amazonaws.cn/en_us/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

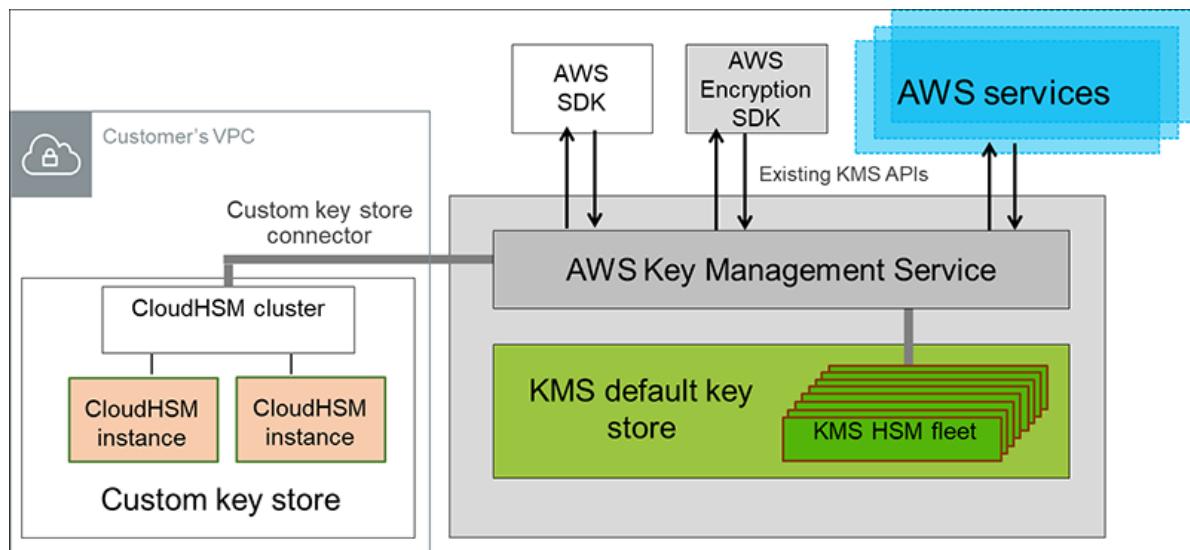
A company requires all the data stored in the cloud to be encrypted at rest. To easily integrate this with other AWS services, they must have full control over the encryption of the created keys and also the ability to immediately remove the key material from AWS KMS. The solution should also be able to audit the key usage independently of AWS CloudTrail.

Which of the following options will meet this requirement?

- Use AWS Key Management Service to create a CMK in a custom key store and store the non-extractable key material in AWS CloudHSM.
- Use AWS Key Management Service to create AWS-managed CMKs and store the non-extractable key material in AWS CloudHSM.
- Use AWS Key Management Service to create a CMK in a custom key store and store the non-extractable key material in Amazon S3.
- Use AWS Key Management Service to create AWS-owned CMKs and store the non-extractable key material in AWS CloudHSM.

Incorrect

The **AWS Key Management Service (KMS)** custom key store feature combines the controls provided by **AWS CloudHSM** with the integration and ease of use of AWS KMS. You can configure your own CloudHSM cluster and authorize AWS KMS to use it as a dedicated key store for your keys rather than the default AWS KMS key store. When you create keys in AWS KMS you can choose to generate the key material in your CloudHSM cluster. CMKs that are generated in your custom key store never leave the HSMs in the CloudHSM cluster in plaintext and all AWS KMS operations that use those keys are only performed in your HSMs.



AWS KMS can help you integrate with other AWS services to encrypt the data that you store in these services and control access to the keys that decrypt it. To immediately remove the key material from AWS KMS, you can use a custom key store. Take note that each custom key store is associated with an AWS CloudHSM cluster in your AWS account. Therefore, when you create an AWS KMS CMK in a custom key store, AWS KMS generates and stores the non-extractable key material for the CMK in an AWS CloudHSM cluster that you own and manage. This is also suitable if you want to be able to audit the usage of all your keys independently of AWS KMS or AWS CloudTrail.

Since you control your AWS CloudHSM cluster, you have the option to manage the lifecycle of your CMKs independently of AWS KMS. There are four reasons why you might find a custom key store useful:

1. You might have keys that are explicitly required to be protected in a single-tenant HSM or in an HSM over which you have direct control.
2. You might have keys that are required to be stored in an HSM that has been validated to FIPS 140-2 level 3 overall (the HSMs used in the standard AWS KMS key store are either validated or in the process of being validated to level 2 with level 3 in multiple categories).
3. You might need the ability to immediately remove key material from AWS KMS and to prove you have done so by independent means.
4. You might have a requirement to be able to audit all use of your keys independently of AWS KMS or AWS CloudTrail.

Hence, the correct answer in this scenario is: ***Use AWS Key Management Service to create a CMK in a custom key store and store the non-extractable key material in AWS CloudHSM***.

The option that says: ***Use AWS Key Management Service to create a CMK in a custom key store and store the non-extractable key material in Amazon S3*** is incorrect because Amazon S3 is not a suitable storage service to use in storing encryption keys. You have to use AWS CloudHSM instead.

The options that say: ***Use AWS Key Management Service to create AWS-owned CMKs and store the non-extractable key material in AWS CloudHSM*** and ***Use AWS Key Management Service to create AWS-managed CMKs and store the non-extractable key material in AWS CloudHSM*** are both incorrect because the scenario requires you to have **full control over the encryption of the created key. AWS-owned CMKs and AWS-managed CMKs are managed by AWS. Moreover, these options do not allow you to audit the key usage independently of AWS CloudTrail.**

References:

<https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html>

<https://aws.amazon.com/kms/faqs/>

<https://aws.amazon.com/blogs/security/are-kms-custom-key-stores-right-for-you/>

Check out this AWS KMS Cheat Sheet:

<https://tutorialsdojo.com/aws-key-management-service-aws-kms/>

2. QUESTION

Category: CSAA – Design High-Performing Architectures

A startup is using Amazon RDS to store data from a web application. Most of the time, the application has low user activity but it receives bursts of traffic within seconds whenever there is a new product announcement. The Solutions Architect needs to create a solution that will allow users around the globe to access the data using an API.

What should the Solutions Architect do meet the above requirement?

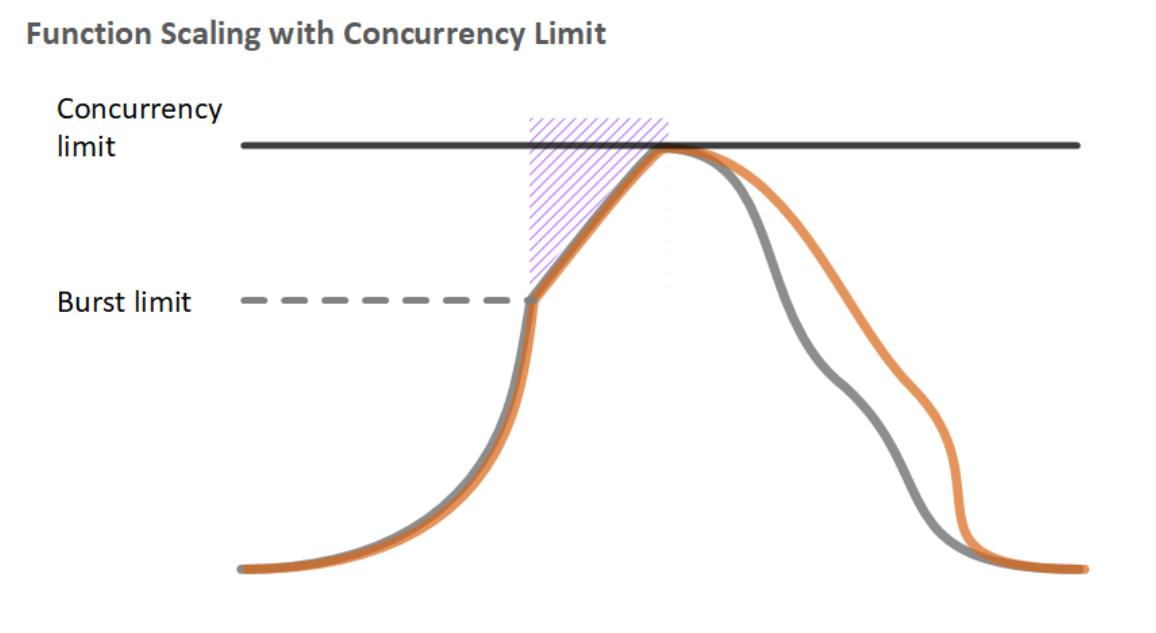
- Create an API using Amazon API Gateway and use the Amazon ECS cluster with Service Auto Scaling to handle the bursts of traffic in seconds.
- Create an API using Amazon API Gateway and use Amazon Elastic Beanstalk with Auto Scaling to handle the bursts of traffic in seconds.
- Create an API using Amazon API Gateway and use an Auto Scaling group of Amazon EC2 instances to handle the bursts of traffic in seconds.
- **Create an API using Amazon API Gateway and use AWS Lambda to handle the bursts of traffic in seconds.**

Correct

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. With Lambda, you can run code for virtually any type of application or backend service – all with zero administration. Just upload your code, and Lambda takes care of everything required to run and scale your code with high availability. You can set up your code to automatically trigger from other AWS services or call it directly from any web or mobile app.

The first time you invoke your function, AWS Lambda creates an instance of the function and runs its handler method to process the event. When the function returns a response, it stays active and waits to process additional events. If you invoke the function again while the first event is being processed, Lambda initializes another instance, and the function processes the two events concurrently. As more events come in, Lambda routes them to available instances and creates new instances as needed. When the number of requests decreases, Lambda stops unused instances to free up the scaling capacity for other functions.

Function Scaling with Concurrency Limit



Your functions' *concurrency* is the number of instances that serve requests at a given time. For an initial burst of traffic, your functions' cumulative concurrency in a Region can reach an initial level of between 500 and 3000, which varies per Region.

Based on the given scenario, you need to create a solution that will satisfy the two requirements. The first requirement is to create a solution that will allow the users to access the data using an API. To implement this solution, you can use Amazon API Gateway. The second requirement is to handle the burst of traffic within seconds. You should use AWS Lambda in this scenario because Lambda functions can absorb reasonable bursts of traffic for approximately 15-30 minutes.

Lambda can scale faster than the regular Auto Scaling feature of Amazon EC2, Amazon Elastic Beanstalk, or Amazon ECS. This is because AWS Lambda is more lightweight than other computing services. Under the hood, Lambda can run your code to thousands of available AWS-managed EC2 instances (that could already be running) within seconds to accommodate traffic. This is faster than the Auto Scaling process of launching new EC2 instances that could take a few minutes or so. An alternative is to overprovision your compute capacity but that will incur significant costs. The best option to implement given the requirements is a combination of AWS Lambda and Amazon API Gateway.

Hence, the correct answer is: ***Create an API using Amazon API Gateway and use AWS Lambda to handle the bursts of traffic.***

The option that says: ***Create an API using Amazon API Gateway and use the Amazon ECS cluster with Service Auto Scaling to handle the bursts of traffic in seconds*** is incorrect. AWS Lambda is a better option than Amazon ECS since it can handle a sudden burst of traffic within seconds and not minutes.

The option that says: ***Create an API using Amazon API Gateway and use Amazon Elastic Beanstalk with Auto Scaling to handle the bursts of traffic in seconds*** is incorrect because just like the previous option, the use of Auto Scaling has a delay of a few minutes as it launches new EC2 instances that will be used by Amazon Elastic Beanstalk.

The option that says: ***Create an API using Amazon API Gateway and use an Auto Scaling group of Amazon EC2 instances to handle the bursts of traffic in seconds*** is incorrect because the processing time of Amazon EC2 Auto Scaling to provision new resources takes minutes. Take note that in the scenario, a burst of traffic within seconds is expected to happen.

References:

<https://aws.amazon.com/blogs/startups/from-0-to-100-k-in-seconds-instant-scale-with-aws-lambda/>

<https://docs.aws.amazon.com/lambda/latest/dg/invocation-scaling.html>

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

3. QUESTION

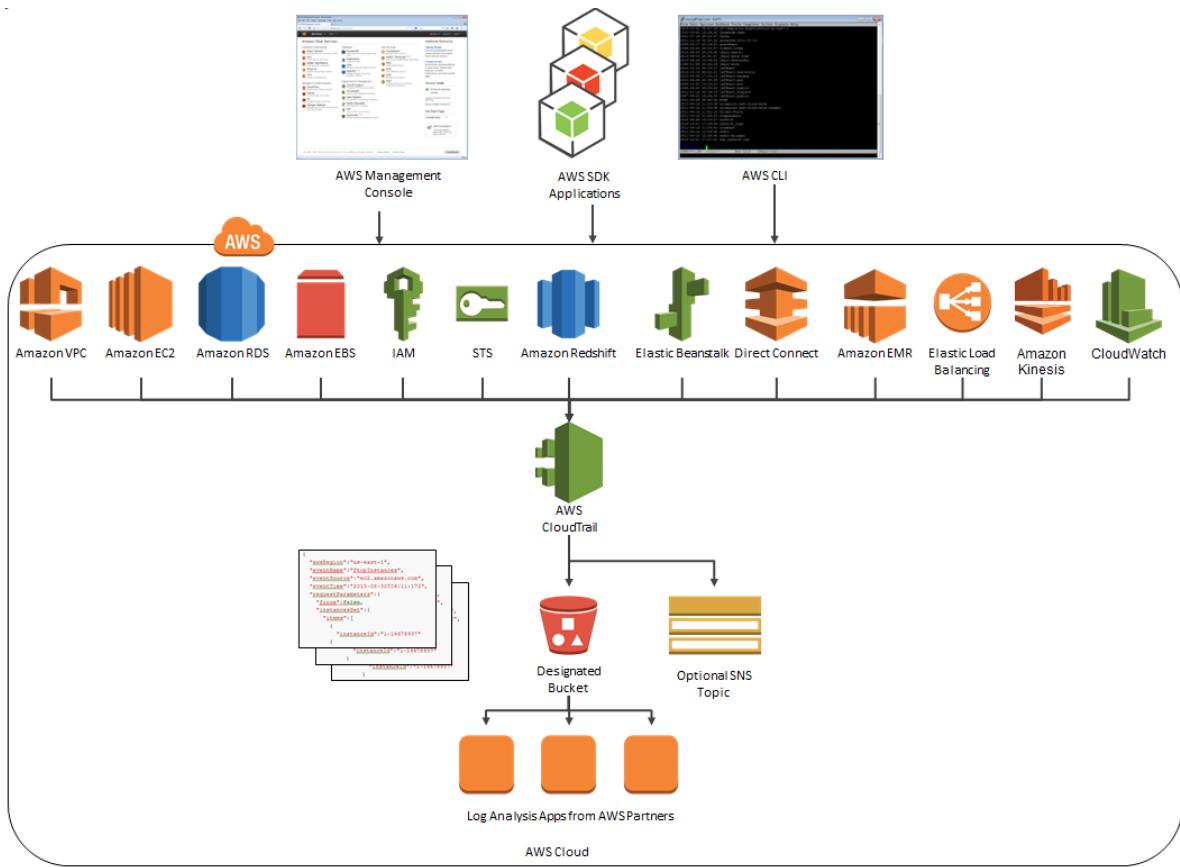
Category: CSAA – Design Secure Applications and Architectures

A company needs to design an online analytics application that uses Redshift Cluster for its data warehouse. Which of the following services allows them to monitor all API calls in Redshift instance and can also provide secured data for auditing and compliance purposes?

- Amazon Redshift Spectrum
- AWS X-Ray
- Amazon CloudWatch
- **AWS CloudTrail**

Correct

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. By default, CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to Event history.



CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, API calls, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting.

Hence, the correct answer is: ***AWS CloudTrail.***

Amazon CloudWatch is incorrect. Although this is also a monitoring service, it cannot track the API calls to your AWS resources.

AWS X-Ray is incorrect because this is not a suitable service to use to track each API call to your AWS resources. It just helps you debug and analyze your microservices applications with request tracing so you can find the root cause of issues and performance.

Amazon Redshift Spectrum is incorrect because this is not a monitoring service but rather a feature of Amazon Redshift that enables you to query and analyze all of your data in Amazon S3 using the open data formats you already use, with no data loading or transformations needed.

References:

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

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-user-guide.html>

Check out this AWS CloudTrail Cheat Sheet:

<https://tutorialsdojo.com/aws-cloudtrail/>

4. QUESTION

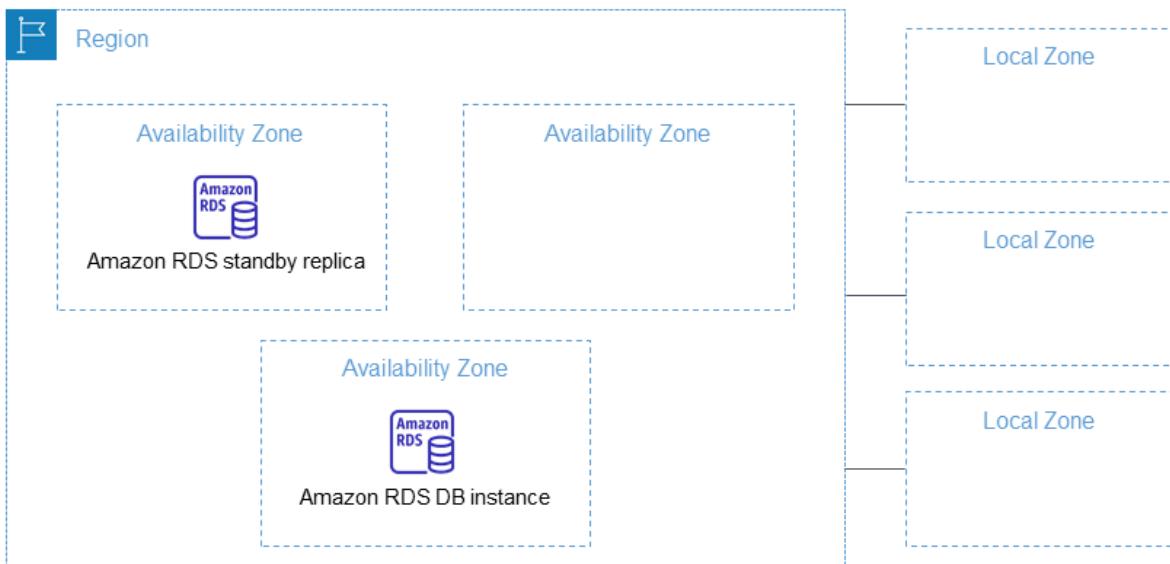
Category: CSAA – Design Resilient Architectures

There are a lot of outages in the Availability Zone of your RDS database instance to the point that you have lost access to the database. What could you do to prevent losing access to your database in case that this event happens again?

- **Enable Multi-AZ failover**
- Make a snapshot of the database
- Increase the database instance size
- Create a read replica

Correct

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances, making them a natural fit for production database workloads. For this scenario, ***enabling Multi-AZ failover*** is the correct answer. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.



In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora), so that you can resume database operations as soon as the failover is complete.

Making a snapshot of the database allows you to have a backup of your database, but it does not provide immediate availability in case of AZ failure. So this is incorrect.

Increasing the database instance size is not a solution for this problem. Doing this action addresses the need to upgrade your compute capacity but does not solve the requirement of providing access to your database even in the event of a loss of one of the Availability Zones.

Creating a read replica is incorrect because this simply provides enhanced performance for read-heavy database workloads. Although you can promote a read replica, its asynchronous replication might not provide you the latest version of your database.

Reference:

<https://aws.amazon.com/rds/details/multi-az/>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company hosts multiple applications in their VPC. While monitoring the system, they noticed that multiple port scans are coming in from a specific IP address block that is trying to connect to several AWS resources inside their VPC. The internal security team has requested that all offending IP addresses be denied for the next 24 hours for security purposes.

Which of the following is the best method to quickly and temporarily deny access from the specified IP addresses?

- Configure the firewall in the operating system of the EC2 instances to deny access from the IP address block.
- Create a policy in IAM to deny access from the IP Address block.
- **Modify the Network Access Control List associated with all public subnets in the VPC to deny access from the IP Address block.**
- Add a rule in the Security Group of the EC2 instances to deny access from the IP Address block.

Correct

To control the traffic coming in and out of your VPC network, you can use the **network access control list (ACL)***. It is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. This is the best solution among other options as you can easily add and remove the restriction in a matter of minutes.

Creating a policy in IAM to deny access from the IP Address block is incorrect as an IAM policy does not control the inbound and outbound traffic of your VPC.

Adding a rule in the Security Group of the EC2 instances to deny access from the IP Address block is incorrect as **although a Security Group acts as a firewall, it will only control both inbound and outbound traffic at the instance level and not on the whole VPC.**

Configuring the firewall in the operating system of the EC2 instances to deny access from the IP address block is incorrect because adding a firewall in the underlying operating system of the EC2 instance is not enough; the attacker can just connect to other AWS resources since the network access control list still allows them to do so.

Reference:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html

Check out this Amazon VPC Cheat Sheet:

6. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company has 3 DevOps engineers that are handling its software development and infrastructure management processes. One of the engineers accidentally deleted a file hosted in Amazon S3 which has caused disruption of service.

What can the DevOps engineers do to prevent this from happening again?

- Use S3 Infrequently Accessed storage to store the data.
- **Enable S3 Versioning and Multi-Factor Authentication Delete on the bucket.**
- Create an IAM bucket policy that disables delete operation.
- Set up a signed URL for all users.

Correct

To avoid accidental deletion in Amazon S3 bucket, you can:

- Enable Versioning
- Enable MFA (Multi-Factor Authentication) Delete

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

If the MFA (Multi-Factor Authentication) Delete is enabled, it requires additional authentication for either of the following operations:

- Change the versioning state of your bucket
- Permanently delete an object version

Using S3 Infrequently Accessed storage to store the data is incorrect. Switching your storage class to S3 Infrequent Access won't help mitigate accidental deletions.

Setting up a signed URL for all users is incorrect. Signed URLs give you more control over access to your content, so this feature deals more on accessing rather than deletion.

Creating an IAM bucket policy that disables delete operation is incorrect. If you create a bucket policy preventing deletion, other users won't be able to delete objects that should be deleted. You only want to prevent accidental deletion, not disable the action itself.

Reference:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

7. QUESTION

Category: CSAA – Design Resilient Architectures

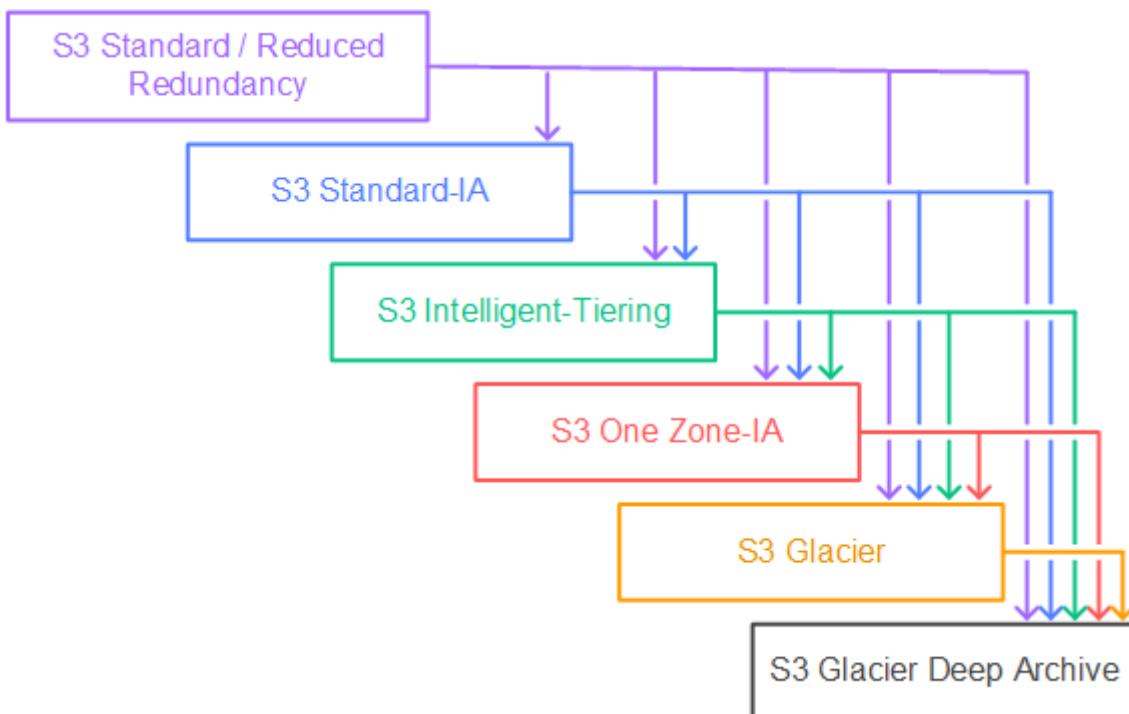
A company plans to host a web application in an Auto Scaling group of Amazon EC2 instances. The application will be used globally by users to upload and store several types of files. Based on user trends, files that are older than 2 years must be stored in a different storage class. The Solutions Architect of the company needs to create a cost-effective and scalable solution to store the old files yet still provide durability and high availability.

Which of the following approach can be used to fulfill this requirement? (Select TWO.)

- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.
- Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.
- Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years.
- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.
- Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.

Correct

Amazon S3 stores data as objects within buckets. An object is a file and any optional metadata that describes the file. To store a file in Amazon S3, you upload it to a bucket. When you upload a file as an object, you can set permissions on the object and any metadata. Buckets are containers for objects. You can have one or more buckets. You can control access for each bucket, deciding who can create, delete, and list objects in it. You can also choose the geographical region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.



To move a file to a different storage class, you can use Amazon S3 or Amazon EFS. Both services have lifecycle configurations. Take note that Amazon EFS can only transition a file to the IA storage class after 90 days. Since you need to move the files that are older than 2 years to a more cost-effective and scalable solution, you should use the Amazon S3 lifecycle configuration. With S3 lifecycle rules, you can transition files to S3 Standard IA or S3 Glacier. Using S3 Glacier expedited retrieval, you can quickly access your files within 1-5 minutes.

Hence, the correct answers are:

- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.

- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.

The option that says: ***Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years*** is incorrect because the **maximum days for the EFS lifecycle policy is only 90 days**. The requirement is to move the files that are older than 2 years or 730 days.

The option that says: ***Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because **Amazon EBS costs more and is not as scalable as Amazon S3. It has some limitations when accessed by multiple EC2 instances. There are also huge costs involved in using the multi-attach feature on a Provisioned IOPS EBS volume to allow multiple EC2 instances to access the volume.**

The option that says: ***Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because RAID (Redundant Array of Independent Disks) is just a data storage virtualization technology that combines multiple storage devices to achieve higher performance or data durability. RAID 0 can stripe multiple volumes together for greater I/O performance than you can achieve with a single volume. On the other hand, RAID 1 can mirror two volumes together to achieve on-instance redundancy.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

<https://docs.aws.amazon.com/efs/latest/ug/lifecycle-management-efs.html>

<https://aws.amazon.com/s3/faqs/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

8. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A government entity is conducting a population and housing census in the city. Each household information uploaded on their online portal is stored in encrypted files in Amazon S3. The government assigned its Solutions Architect to set compliance policies that verify sensitive data in a manner that meets their compliance standards. They should also be alerted if there are compromised files detected containing personally identifiable information (PII), protected health information (PHI) or intellectual properties (IP).

Which of the following should the Architect implement to satisfy this requirement?

- Set up and configure Amazon Inspector to send out alert notifications whenever a security violation is detected on their Amazon S3 data.
- **Set up and configure Amazon Macie to monitor and detect usage patterns on their Amazon S3 data.**
- Set up and configure Amazon Rekognition to monitor and recognize patterns on their Amazon S3 data.

- Set up and configure Amazon GuardDuty to monitor malicious activity on their Amazon S3 data.

Incorrect

Amazon Macie is an ML-powered security service that helps you prevent data loss by automatically discovering, classifying, and protecting sensitive data stored in Amazon S3. Amazon Macie uses machine learning to recognize sensitive data such as personally identifiable information (PII) or intellectual property, assigns a business value, and provides visibility into where this data is stored and how it is being used in your organization.

Amazon Macie continuously monitors data access activity for anomalies, and delivers alerts when it detects risk of unauthorized access or inadvertent data leaks. Amazon Macie has ability to detect global access permissions inadvertently being set on sensitive data, detect uploading of API keys inside source code, and verify sensitive customer data is being stored and accessed in a manner that meets their compliance standards.

The screenshot shows the Amazon Macie console interface. On the left, there's a navigation sidebar with options like ALERTS, DASHBOARD, REPORTS, RESEARCH, SETTINGS, and INTEGRATIONS. The main area displays a list of alerts. The first two alerts are for "S3 Bucket uses IAM policy to grant read rights to Everyone" and have a severity of 100. The third alert is for "Access Denied In Secure Account" and has a severity of 50. Each alert card includes a timestamp (e.g., 11 minutes ago, 21 minutes ago, 30 minutes ago), the number of comments (0), and the number of views (0).

Hence, the correct answer is: ***Set up and configure Amazon Macie to monitor and detect usage patterns on their Amazon S3 data.***

The option that says: ***Set up and configure Amazon Rekognition to monitor and recognize patterns on their Amazon S3 data*** is incorrect because Rekognition is simply a service that can identify the objects, people, text, scenes, and activities, as well as detect any inappropriate content on your images or videos.

The option that says: ***Set up and configure Amazon GuardDuty to monitor malicious activity on their Amazon S3 data*** is incorrect because GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

The option that says: ***Set up and configure Amazon Inspector to send out alert notifications whenever a security violation is detected on their Amazon S3 data*** is incorrect because Inspector is basically an automated security assessment service that helps improve the security and compliance of applications deployed on AWS.

References:

<https://docs.aws.amazon.com/macie/latest/userguide/what-is-macie.html>

<https://aws.amazon.com/macie/faq/>

<https://docs.aws.amazon.com/macie/index.html>

Check out this Amazon Macie Cheat Sheet:

<https://tutorialsdojo.com/amazon-macie/>

AWS Security Services Overview – Secrets Manager, ACM, Macie:

9. QUESTION

Category: CSAA – Design Resilient Architectures

A company is in the process of migrating their applications to AWS. One of their systems requires a database that can scale globally and handle frequent schema changes. The application should not have any downtime or performance issues whenever there is a schema change in the database. It should also provide a low latency response to high-traffic queries.

Which is the most suitable database solution to use to achieve this requirement?

- An Amazon RDS instance in Multi-AZ Deployments configuration
- **Amazon DynamoDB**
- An Amazon Aurora database with Read Replicas
- Redshift

Correct

Before we proceed in answering this question, we must first be clear with the actual definition of a “**schema**”. Basically, the english definition of a schema is: *a representation of a plan or theory in the form of an outline or model*.

Just think of a schema as the “structure” or a “model” of your data in your database. Since the scenario requires that the schema, or the structure of your data, changes frequently, then you have to pick a database which provides a non-rigid and flexible way of adding or removing new types of data. This is a classic example of choosing between a relational database and non-relational (NoSQL) database.

Characteristic	Relational Database Management System (RDBMS)	Amazon DynamoDB
Optimal Workloads	Ad hoc queries; data warehousing; OLAP (online analytical processing).	Web-scale applications, including social networks, gaming, media sharing, and IoT (Internet of Things).
Data Model	The relational model requires a well-defined schema, where data is normalized into tables, rows and columns. In addition, all of the relationships are defined among tables, columns, indexes, and other database elements.	DynamoDB is schemaless. Every table must have a primary key to uniquely identify each data item, but there are no similar constraints on other non-key attributes. DynamoDB can manage structured or semi-structured data, including JSON documents.
Data Access	SQL (Structured Query Language) is the standard for storing and retrieving data. Relational databases offer a rich set of tools for simplifying the development of database-driven applications, but all of these tools use SQL.	You can use the AWS Management Console or the AWS CLI to work with DynamoDB and perform ad hoc tasks. Applications can leverage the AWS software development kits (SDKs) to work with DynamoDB using object-based, document-centric, or low-level interfaces.
Performance	Relational databases are optimized for storage, so performance generally depends on the disk subsystem. Developers and database administrators must optimize queries, indexes, and table structures in order to achieve peak performance.	DynamoDB is optimized for compute, so performance is mainly a function of the underlying hardware and network latency. As a managed service, DynamoDB insulates you and your applications from these implementation details, so that you can focus on designing and building robust, high-performance applications.
Scaling	It is easiest to scale up with faster hardware. It is also possible for database tables to span across multiple hosts in a distributed system, but this requires additional investment. Relational databases have maximum sizes for the number and size of files, which imposes upper limits on scalability.	DynamoDB is designed to scale out using distributed clusters of hardware. This design allows increased throughput without increased latency. Customers specify their throughput requirements, and DynamoDB allocates sufficient resources to meet those requirements. There are no upper limits on the number of items per table, nor the total size of that table.

A relational database is known for having a rigid schema, with a lot of constraints and limits as to which (and what type of) data can be inserted or not. It is primarily used for scenarios where you have to support complex queries which fetch data across a number of tables. It is best for scenarios where you have complex table relationships but for use cases where you need to have a flexible schema, this is not a suitable database to use.

For NoSQL, it is not as rigid as a relational database because you can easily add or remove rows or elements in your table/collection entry. It also has a more flexible schema because it can store complex hierarchical data within a single item which, unlike a relational database, does not entail changing multiple related tables. Hence, the best answer to be used here is a NoSQL database, like DynamoDB. When your business requires a low-latency response to high-traffic queries, taking advantage of a NoSQL system generally makes technical and economic sense.

Amazon DynamoDB helps solve the problems that limit the relational system scalability by avoiding them. In DynamoDB, you design your schema specifically to make the most common and important queries as fast and as inexpensive as possible. Your data structures are tailored to the specific requirements of your business use cases.

Remember that a relational database system **does not scale** well for the following reasons:

- It normalizes data and stores it on multiple tables that require multiple queries to write to disk.
- It generally incurs the performance costs of an ACID-compliant transaction system.
- It uses expensive joins to reassemble required views of query results.

For DynamoDB, it scales well due to these reasons:

- Its **schema flexibility** lets DynamoDB store complex hierarchical data within a single item.
- DynamoDB is not a totally *schemaless* database since the very definition of a schema is just the model or structure of your data.
- Composite key design lets it store related items close together on the same table.

An Amazon RDS instance in Multi-AZ Deployments configuration and ***an Amazon Aurora database with Read Replicas*** are incorrect because both of them are a type of relational database.

Redshift is incorrect because it is primarily used for OLAP systems.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-general-nosql-design.html>
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-relational-modeling.html>
<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/SQLtoNoSQL.html>

Also check the **AWS Certified Solutions Architect Official Study Guide: Associate Exam** 1st Edition and turn to page 161 which talks about NoSQL Databases.

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

10. QUESTION

Category: CSAA – Design High-Performing Architectures

A popular social media website uses a CloudFront web distribution to serve their static contents to their millions of users around the globe. They are receiving a number of complaints recently that their users take a lot of time to log into their website. There are also occasions when their users are getting HTTP 504 errors. You are instructed by your manager to significantly reduce the user's login time to further optimize the system.

Which of the following options should you use together to set up a cost-effective solution that can improve your application's performance? (Select TWO.)

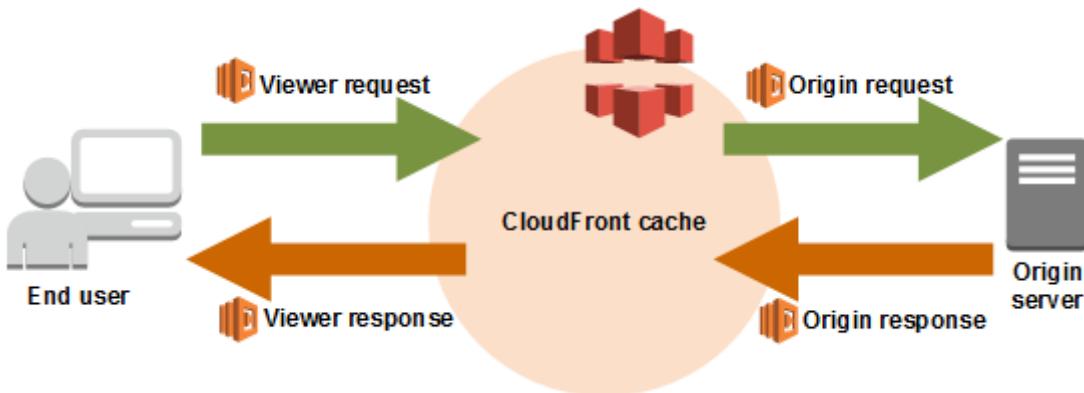
- Configure your origin to add a `Cache-Control max-age` directive to your objects, and specify the longest practical value for `max-age` to increase the cache hit ratio of your CloudFront distribution.
- Deploy your application to multiple AWS regions to accommodate your users around the world. Set up a Route 53 record with latency routing policy to route incoming traffic to the region that provides the best latency to the user.
- Use multiple and geographically disperse VPCs to various AWS regions then create a transit VPC to connect all of your resources. In order to handle the requests faster, set up Lambda functions in each region using the AWS Serverless Application Model (SAM) service.

- Set up an origin failover by creating an origin group with two origins. Specify one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.
- Customize the content that the CloudFront web distribution delivers to your users using Lambda@Edge, which allows your Lambda functions to execute the authentication process in AWS locations closer to the users.

Correct

Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests and responses at the following points:

- After CloudFront receives a request from a viewer (viewer request)
- Before CloudFront forwards the request to the origin (origin request)
- After CloudFront receives the response from the origin (origin response)
- Before CloudFront forwards the response to the viewer (viewer response)



In the given scenario, you can use Lambda@Edge to allow your Lambda functions to customize the content that CloudFront delivers and to execute the authentication process in AWS locations closer to the users. In addition, you can set up an origin failover by creating an origin group with two origins with one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin fails. This will alleviate the occasional HTTP 504 errors that users are experiencing. Therefore, the correct answers are:

- *- Customize the content that the CloudFront web distribution delivers to your users using Lambda@Edge, which allows your Lambda functions to execute the authentication process in AWS locations closer to the users.***
- *- Set up an origin failover by creating an origin group with two origins. Specify one as the primary origin and the other as the second origin which CloudFront automatically switches to when the primary origin returns specific HTTP status code failure responses.***

The option that says: ***Use multiple and geographically disperse VPCs to various AWS regions then create a transit VPC to connect all of your resources. In order to handle the requests faster, set up Lambda functions in each region using the AWS Serverless Application Model (SAM) service*** is incorrect because of the same reason provided above. Although setting up multiple VPCs across various regions which are connected with a transit VPC is valid, this solution still entails higher setup and maintenance costs. A more cost-effective option would be to use Lambda@Edge instead.

The option that says: ***Configure your origin to add a Cache-Control max-age directive to your objects, and specify the longest practical value for max-age to increase the cache hit ratio of your CloudFront distribution*** is incorrect because improving the cache hit ratio for the CloudFront distribution is irrelevant in this scenario. You can improve your cache performance by increasing the proportion of your viewer requests that are served from CloudFront edge caches instead of going to your origin servers for

content. However, take note that the problem in the scenario is the sluggish authentication process of your global users and not just the caching of the static objects.

The option that says: ***Deploy your application to multiple AWS regions to accommodate your users around the world. Set up a Route 53 record with latency routing policy to route incoming traffic to the region that provides the best latency to the user*** is incorrect because although this may resolve the performance issue, this solution entails a significant implementation cost since you have to deploy your application to multiple AWS regions. Remember that the scenario asks for a solution that will improve the performance of the application with **minimal cost**.

References:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/high_availability_origin_failover.html

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-edge.html>

Check out these Amazon CloudFront and AWS Lambda Cheat Sheets:

<https://tutorialsdojo.com/amazon-cloudfront/>

<https://tutorialsdojo.com/aws-lambda/>

11. QUESTION

Category: CSAA – Design Resilient Architectures

A company plans to migrate its on-premises workload to AWS. The current architecture is composed of a Microsoft SharePoint server that uses a Windows shared file storage. The Solutions Architect needs to use a cloud storage solution that is highly available and can be integrated with Active Directory for access control and authentication.

Which of the following options can satisfy the given requirement?

- Create a file system using Amazon FSx for Windows File Server and join it to an Active Directory domain in AWS.
- Create a Network File System (NFS) file share using AWS Storage Gateway.
- Create a file system using Amazon EFS and join it to an Active Directory domain.
- Launch an Amazon EC2 Windows Server to mount a new S3 bucket as a file volume.

Correct

Amazon FSx for Windows File Server provides fully managed, highly reliable, and scalable file storage that is accessible over the industry-standard Service Message Block (SMB) protocol. It is built on Windows Server, delivering a wide range of administrative features such as user quotas, end-user file restore, and Microsoft Active Directory (AD) integration. Amazon FSx is accessible from Windows, Linux, and MacOS compute instances and devices. Thousands of compute instances and devices can access a file system concurrently.

The screenshot shows the 'Windows authentication' configuration page. It includes a note about choosing an Active Directory for user authentication and access control. Two options are listed: 'AWS Managed Microsoft Active Directory' (selected with a blue circle) and 'Self-managed Microsoft Active Directory'. Below this, there's a dropdown menu labeled 'Choose a directory' with a downward arrow, and a button labeled 'Create new directory' with a plus sign icon. The entire dropdown menu is highlighted with an orange border.

Amazon FSx works with Microsoft Active Directory to integrate with your existing Microsoft Windows environments. You have two options to provide user authentication and access control for your file system: AWS Managed Microsoft Active Directory and Self-managed Microsoft Active Directory.

Take note that after you create an Active Directory configuration for a file system, you can't change that configuration. However, you can create a new file system from a backup and change the Active Directory integration configuration for that file system. These configurations allow the users in your domain to use their existing identity to access the Amazon FSx file system and to control access to individual files and folders.

Hence, the correct answer is: ***Create a file system using Amazon FSx for Windows File Server and join it to an Active Directory domain in AWS.***

The option that says: ***Create a file system using Amazon EFS and join it to an Active Directory domain*** is incorrect because Amazon EFS does not support Windows systems, only Linux OS. You should use Amazon FSx for Windows File Server instead to satisfy the requirement in the scenario.

The option that says: ***Launch an Amazon EC2 Windows Server to mount a new S3 bucket as a file volume*** is incorrect because you can't integrate Amazon S3 with your existing Active Directory to provide authentication and access control.

The option that says: ***Create a Network File System (NFS) file share using AWS Storage Gateway*** is incorrect because NFS file share is mainly used for Linux systems. Remember that the requirement in the scenario is to use a Windows shared file storage. Therefore, you must use an SMB file share instead, which supports Windows OS and Active Directory configuration. Alternatively, you can also use the Amazon FSx for Windows File Server file system.

References:

<https://docs.aws.amazon.com/fsx/latest/WindowsGuide/aws-ad-integration-fsxW.html>

<https://aws.amazon.com/fsx/windows/faqs/>

<https://docs.aws.amazon.com/storagegateway/latest/userguide/CreatingAnSMBFileShare.html>

Check out this Amazon FSx Cheat Sheet:

<https://tutorialsdojo.com/amazon-fsx/>

12. QUESTION

Category: CSAA – Design High-Performing Architectures

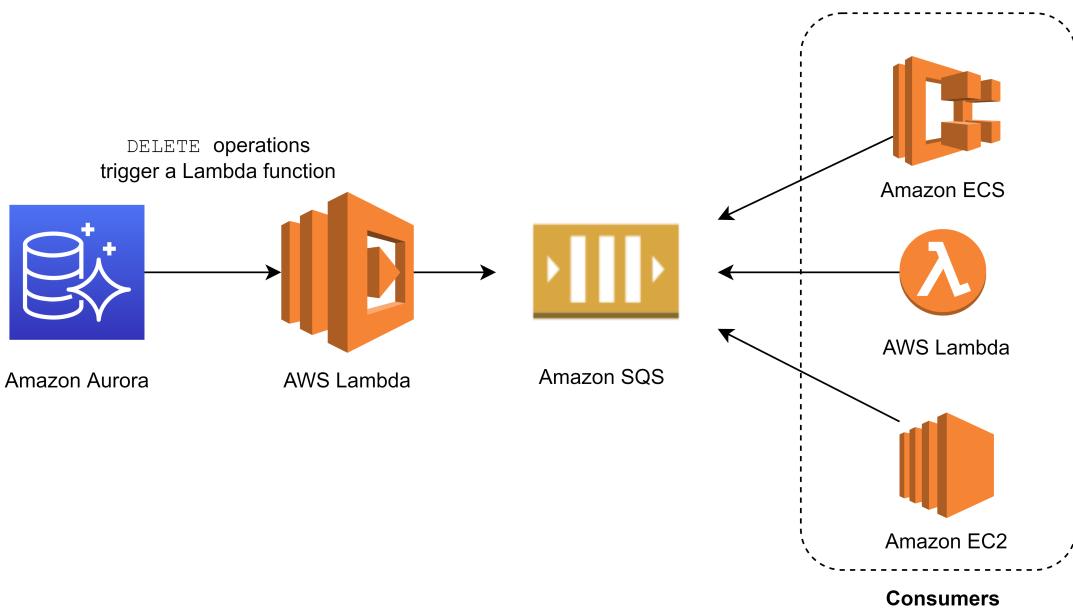
A car dealership website hosted in Amazon EC2 stores car listings in an Amazon Aurora database managed by Amazon RDS. Once a vehicle has been sold, its data must be removed from the current listings and forwarded to a distributed processing system.

Which of the following options can satisfy the given requirement?

- Create an RDS event subscription and send the notifications to Amazon SQS. Configure the SQS queues to fan out the event notifications to multiple Amazon SNS topics. Process the data using Lambda functions.
- **Create a native function or a stored procedure that invokes a Lambda function. Configure the Lambda function to send event notifications to an Amazon SQS queue for the processing system to consume.**
- Create an RDS event subscription and send the notifications to AWS Lambda. Configure the Lambda function to fanout the event notifications to multiple Amazon SQS queues to update the target groups.
- Create an RDS event subscription and send the notifications to AWS Lambda. Configure the Lambda function to fan out the event notifications to multiple Amazon SQS queues to update the processing system.

Incorrect

You can invoke an AWS Lambda function from an Amazon Aurora MySQL-Compatible Edition DB cluster with a native function or a stored procedure. This approach can be useful when you want to integrate your database running on Aurora MySQL with other AWS services. For example, you might want to capture data changes whenever a row in a table is modified in your database.



In the scenario, you can trigger a Lambda function whenever a listing is deleted from the database. You can then write the logic of the function to send the listing data to an SQS queue and have different processes consume it.

Hence, the correct answer is: ***Create a native function or a stored procedure that invokes a Lambda function. Configure the Lambda function to send event notifications to an Amazon SQS queue for the processing system to consume.***

RDS events only provide operational events such as DB instance events, DB parameter group events, DB security group events, and DB snapshot events. What we need in the scenario is to capture data-modifying events (`INSERT`, `DELETE`, `UPDATE`) which can be achieved thru native functions or stored procedures. Hence, the following options are incorrect:

- Create an RDS event subscription and send the notifications to Amazon SQS. Configure the SQS queues to fan out the event notifications to multiple Amazon SNS topics. Process the data using Lambda functions.

- Create an RDS event subscription and send the notifications to AWS Lambda. Configure the Lambda function to fan out the event notifications to multiple Amazon SQS queues to update the processing system.

- Create an RDS event subscription and send the notifications to Amazon SNS. Configure the SNS topic to fan out the event notifications to multiple Amazon SQS queues. Process the data using Lambda functions.

References:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Integrating.Lambda.html>

<https://aws.amazon.com/blogs/database/capturing-data-changes-in-amazon-aurora-using-aws-lambda/>

Check out this Amazon Aurora Cheat Sheet:

<https://tutorialsdojo.com/amazon-aurora/>

13. QUESTION

Category: CSAA – Design Resilient Architectures

A retail website has intermittent, sporadic, and unpredictable transactional workloads throughout the day that are hard to predict. The website is currently hosted on-premises and is slated to be migrated to AWS. A new relational database is needed that autoscales capacity to meet the needs of the application's peak load and scales back down when the surge of activity is over.

Which of the following option is the MOST cost-effective and suitable database setup in this scenario?

- Launch an Amazon Aurora Serverless DB cluster then set the minimum and maximum capacity for the cluster.
- Launch a DynamoDB Global table with Auto Scaling enabled.
- Launch an Amazon Aurora Provisioned DB cluster with burstable performance DB instance class types.
- Launch an Amazon Redshift data warehouse cluster with Concurrency Scaling.

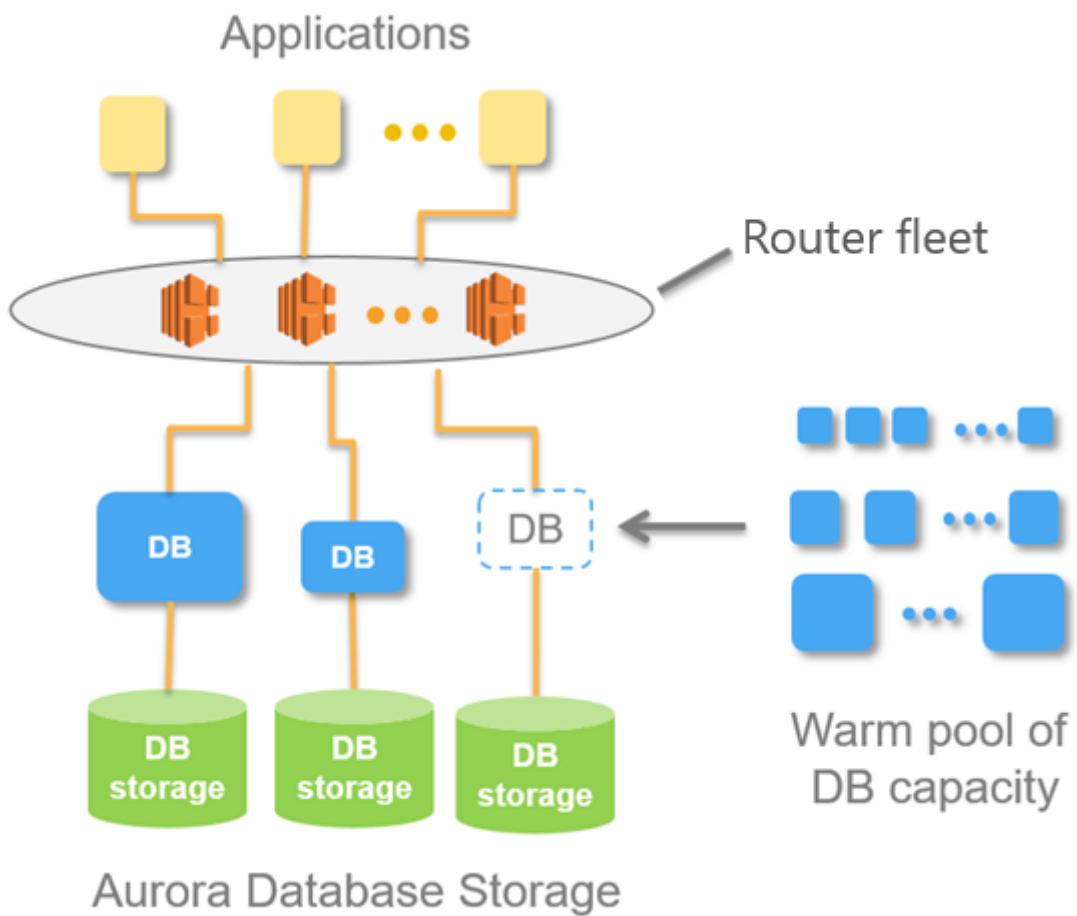
Correct

Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora. An Aurora Serverless DB cluster is a DB cluster that automatically starts up, shuts down, and scales up or down its compute capacity based on your application's needs. Aurora Serverless provides a relatively simple, cost-effective option for infrequent, intermittent, sporadic or unpredictable workloads. It can provide this because it automatically starts up, scales compute capacity to match your application's usage and shuts down when it's not in use.

Take note that a non-Serverless DB cluster for Aurora is called a provisioned DB cluster. Aurora Serverless clusters and provisioned clusters both have the same kind of high-capacity, distributed, and highly available storage volume.

When you work with Amazon Aurora without Aurora Serverless (provisioned DB clusters), you can choose your DB instance class size and create Aurora Replicas to increase read throughput. If your workload changes, you can modify the DB instance class size and change the number of Aurora Replicas. This model works well when the database workload is predictable, because you can adjust capacity manually based on the expected workload.

However, in some environments, workloads can be intermittent and unpredictable. There can be periods of heavy workloads that might last only a few minutes or hours, and also long periods of light activity, or even no activity. Some examples are retail websites with intermittent sales events, reporting databases that produce reports when needed, development and testing environments, and new applications with uncertain requirements. In these cases and many others, it can be difficult to configure the correct capacity at the right times. It can also result in higher costs when you pay for capacity that isn't used.



With Aurora Serverless , you can create a database endpoint without specifying the DB instance class size. You set the minimum and maximum capacity. With Aurora Serverless, the database endpoint connects to a *proxy fleet* that routes the workload to a fleet of resources that are automatically scaled. Because of the proxy fleet, connections are continuous as Aurora Serverless scales the resources automatically based on the minimum and maximum capacity specifications. Database client applications don't need to change to use the proxy fleet. Aurora Serverless manages the connections automatically. Scaling is rapid because it uses a pool of "warm" resources that are always ready to service requests. Storage and processing are separate, so you can scale down to zero processing and pay only for storage.

Aurora Serverless introduces a new `serverless` DB engine mode for Aurora DB clusters. Non-Serverless DB clusters use the `provisioned` DB engine mode.

Hence, the correct answer is: ***Launch an Amazon Aurora Serverless DB cluster then set the minimum and maximum capacity for the cluster.***

The option that says: ***Launch an Amazon Aurora Provisioned DB cluster with burstable performance DB instance class types*** is incorrect because an Aurora Provisioned DB cluster is not suitable for intermittent, sporadic, and unpredictable transactional workloads. This model works well when the database workload is predictable because you can adjust capacity manually based on the expected workload. A better database setup here is to use an Amazon Aurora Serverless cluster.

The option that says: ***Launch a DynamoDB Global table with Auto Scaling enabled*** is incorrect because although it is using Auto Scaling, the scenario explicitly indicated that you need a relational database to handle your transactional workloads. DynamoDB is a NoSQL database and is not suitable for this use case. Moreover, the use of a DynamoDB Global table is not warranted since this is primarily used if you need a fully managed, multi-region, and multi-master database that provides fast, local, read and write performance for massively scaled, global applications.

The option that says: ***Launch an Amazon Redshift data warehouse cluster with Concurrency Scaling*** is incorrect because this type of database is primarily used for online analytical processing (OLAP) and not for online transactional processing (OLTP). Concurrency Scaling is simply an Amazon Redshift feature that automatically and elastically scales query processing power of your Redshift cluster to provide

consistently fast performance for hundreds of concurrent queries.

References:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.how-it-works.html>

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless.html>

14. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company hosted an e-commerce website on an Auto Scaling group of EC2 instances behind an Application Load Balancer. The Solutions Architect noticed that the website is receiving a large number of illegitimate external requests from multiple systems with IP addresses that constantly change. To resolve the performance issues, the Solutions Architect must implement a solution that would block the illegitimate requests with minimal impact on legitimate traffic.

Which of the following options fulfills this requirement?

- Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer.
- Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests.
- **Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.**
- Create a custom rule in the security group of the Application Load Balancer to block the offending requests.

Incorrect

AWS WAF is tightly integrated with Amazon CloudFront, the Application Load Balancer (ALB), Amazon API Gateway, and AWS AppSync – services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on regional services, such as Application Load Balancer, Amazon API Gateway, and AWS AppSync, your rules run in the region and can be used to protect Internet-facing resources as well as internal resources.

Rule

Name

[Validate](#)

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Type

Select Rate-based rule

Request rate details

Rate limit
 The rate limit is the maximum number of requests from a single IP address that are allowed in a five-minute period. This value is continually evaluated, and requests will be blocked once this limit is reached. The IP address is automatically unblocked after it falls below the limit.

Rate limit must be between 100 and 20,000,000.

IP address to use for rate limiting
 When a request comes through a CDN or other proxy network, the source IP address identifies the proxy and the original IP address is sent in a header. Use caution with the option, IP address in header, because headers can be handled inconsistently by proxies and they can be modified to bypass inspection.

 Source IP address

 IP address in header

Criteria to count request towards rate limit
 Choose whether to count all requests for each IP address or to only count requests that match the criteria of a rule statement.

 Consider all requests

 Only consider requests that match the criteria in a rule statement

A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests.

Based on the given scenario, the requirement is to limit the number of requests from the illegitimate requests without affecting the genuine requests. To accomplish this requirement, you can use AWS WAF web ACL. There are two types of rules in creating your own web ACL rule: regular and rate-based rules. You need to select the latter to add a rate limit to your web ACL. After creating the web ACL, you can associate it with ALB. When the rule action triggers, AWS WAF applies the action to additional requests from the IP address until the request rate falls below the limit.

Hence, the correct answer is: ***Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.***

The option that says: ***Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer*** is incorrect because a regular rule only matches the statement defined in the rule. If you need to add a rate limit to your rule, you should create a rate-based rule.

The option that says: ***Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests*** is incorrect. Although NACLs can help you block incoming traffic, this option wouldn't be able to limit the number of requests from a single IP address that is dynamically changing.

The option that says: ***Create a custom rule in the security group of the Application Load Balancer to block the offending requests*** is incorrect because the security group can only allow incoming traffic. Remember that you can't deny traffic using security groups. In addition, it is not capable of limiting the rate of traffic to your application unlike AWS WAF.

References:

<https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-rate-based.html>

<https://aws.amazon.com/waf/faqs/>

Check out this AWS WAF Cheat Sheet:

<https://tutorialsdojo.com/aws-waf/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

15. QUESTION

Category: CSAA – Design Resilient Architectures

A suite of web applications is hosted in an Auto Scaling group of EC2 instances across three Availability Zones and is configured with default settings. There is an Application Load Balancer that forwards the request to the respective target group on the URL path. The scale-in policy has been triggered due to the low number of incoming traffic to the application.

Which EC2 instance will be the first one to be terminated by your Auto Scaling group?

- The EC2 instance launched from the oldest launch configuration
- The EC2 instance which has been running for the longest time
- The EC2 instance which has the least number of user sessions
- The instance will be randomly selected by the Auto Scaling group

Correct

The default termination policy is designed to help ensure that your network architecture spans Availability Zones evenly. With the default termination policy, the behavior of the Auto Scaling group is as follows:

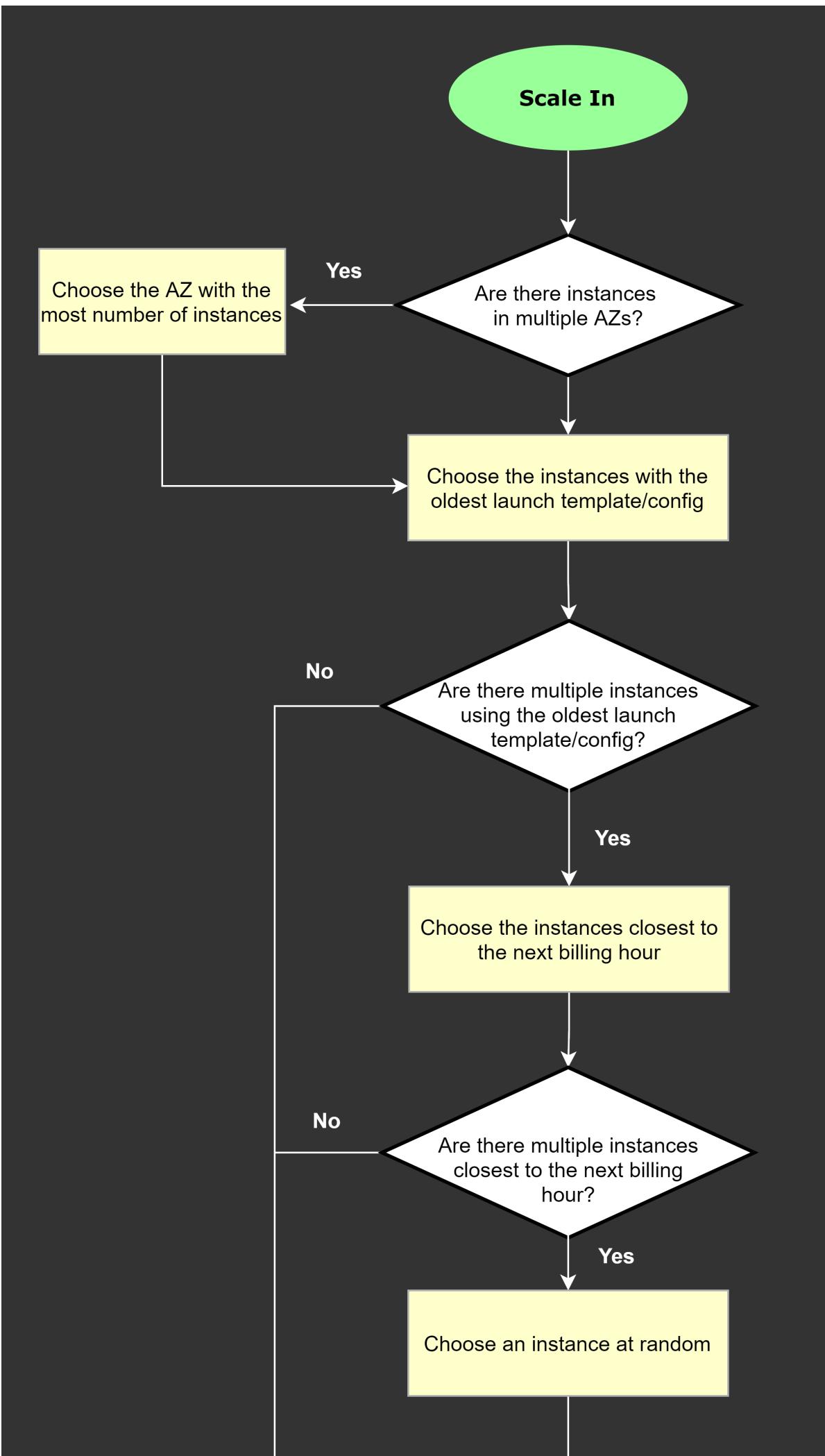
\1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that use the oldest launch configuration.

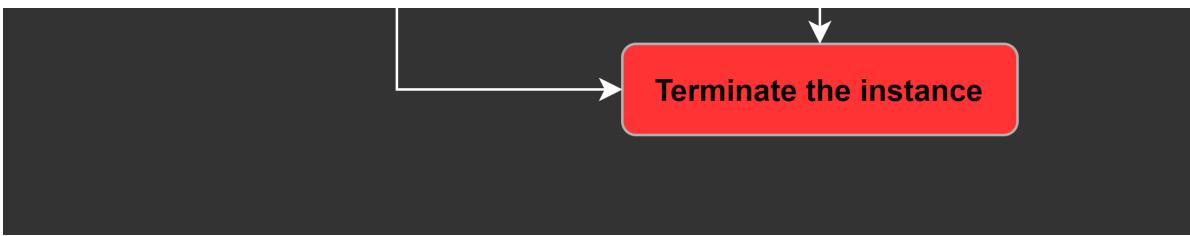
\2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.

\3. If there are multiple instances to terminate based on the above criteria, determine which unprotected instances are closest to the next billing hour. (This helps you maximize the use of your EC2 instances and manage your Amazon EC2 usage costs.) If there is one such instance, terminate it.

\4. If there is more than one unprotected instance closest to the next billing hour, choose one of these instances at random.

The following flow diagram illustrates how the default termination policy works:





References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html#default-termination-policy>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-instance-termination.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

16. QUESTION

Category: CSAA – Design High-Performing Architectures

A company is using a combination of API Gateway and Lambda for the web services of the online web portal that is being accessed by hundreds of thousands of clients each day. They will be announcing a new revolutionary product and it is expected that the web portal will receive a massive number of visitors all around the globe.

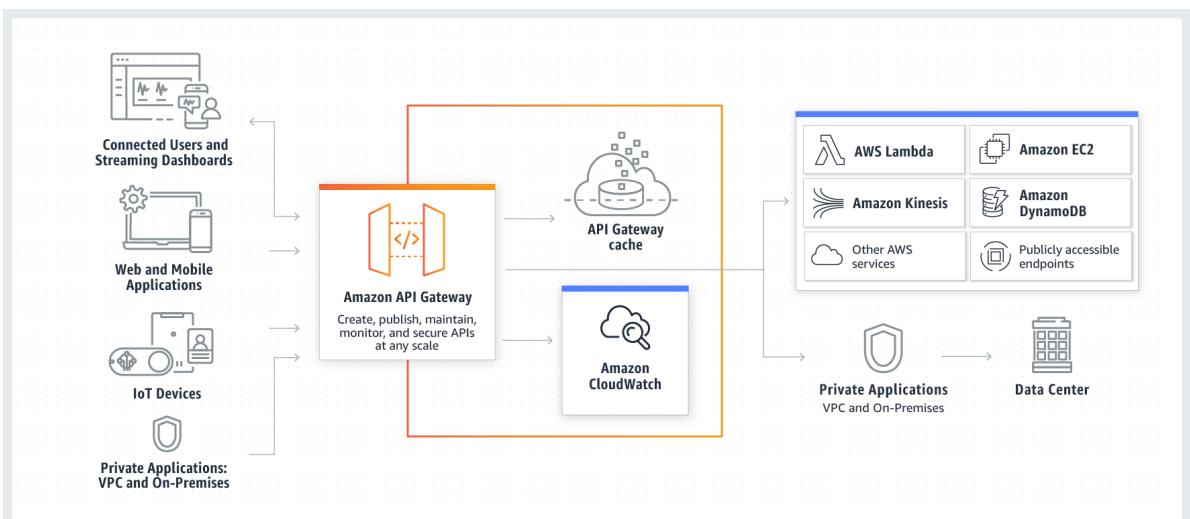
How can you protect the backend systems and applications from traffic spikes?

- Deploy Multi-AZ in API Gateway with Read Replica
- Manually upgrade the EC2 instances being used by API Gateway
- API Gateway will automatically scale and handle massive traffic spikes so you do not have to do anything.
- **Use throttling limits in API Gateway**

Incorrect

Amazon API Gateway provides throttling at multiple levels including global and by a service call.

Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds.



Amazon API Gateway tracks the number of requests per second. Any requests over the limit will receive a 429 HTTP response. The client SDKs generated by Amazon API Gateway retry calls automatically when met with this response.

Hence, the correct answer is: ***Use throttling limits in API Gateway.***

The option that says: ***API Gateway will automatically scale and handle massive traffic spikes so you do not have to do anything*** is incorrect. Although it can scale using AWS Edge locations, you still need to configure the throttling to further manage the bursts of your APIs.

Manually upgrading the EC2 instances being used by API Gateway is incorrect because API Gateway is a fully managed service and hence, you do not have access to its underlying resources.

Deploying Multi-AZ in API Gateway with Read Replica is incorrect because RDS has Multi-AZ and Read Replica capabilities, and not API Gateway.

Reference:

https://aws.amazon.com/api-gateway/faqs/#Throttling_and_Caching

Check out this Amazon API Gateway Cheat Sheet:

<https://tutorialsdojo.com/amazon-api-gateway/>

17. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect designed a serverless architecture that allows AWS Lambda to access an Amazon DynamoDB table named tutorialsdojo in the US East (N. Virginia) region. The IAM policy attached to a Lambda function allows it to put and delete items in the table. The policy must be updated to only allow two operations in the tutorialsdojo table and prevent other DynamoDB tables from being modified.

Which of the following IAM policies fulfill this requirement and follows the principle of granting the least privilege?

- {
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "TutorialsojoTablePolicy",
 "Effect": "Allow",
 "Action": [
 "dynamodb:PutItem",
 "dynamodb>DeleteItem"
],
 "Resource": "arn:aws:dynamodb:us-east-1:120618981206:table/*"
 }
]
}

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "TutorialsojoTablePolicy1",  
            "Effect": "Allow",  
            "Action": [  
                "dynamodb:PutItem",  
                "dynamodb>DeleteItem"  
            ],  
            "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"  
        },  
  
        {  
            "Sid": "TutorialsojoTablePolicy2",  
            "Effect": "Allow",  
            "Action": "dynamodb:*" ,  
            "Resource": "*"  
        }  
    ]  
}
```

```
        "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"
    }
]
}
```

- {
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "TutorialsdojoTablePolicy",
 "Effect": "Allow",
 "Action": [
 "dynamodb:PutItem",
 "dynamodb>DeleteItem"
],
 "Resource": "arn:aws:dynamodb:us-east-1:120618981206:table/tutorialsdojo"
 }
]
}

- {
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "TutorialsdojoTablePolicy1",
 "Effect": "Allow",
 "Action": [
 "dynamodb:PutItem",
 "dynamodb>DeleteItem"
],
 "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"
 },

 {
 "Sid": "TutorialsdojoTablePolicy2",
 "Effect": "Deny",
 "Action": "dynamodb:*",
 "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/*"
 }
]
}

Incorrect

Every AWS resource is owned by an AWS account, and permissions to create or access a resource are governed by permissions policies. An account administrator can attach permissions policies to IAM identities (that is, users, groups, and roles), and some services (such as AWS Lambda) also support attaching permissions policies to resources.

In DynamoDB, the primary resources are tables. DynamoDB also supports additional resource types, indexes, and streams. However, you can create indexes and streams only in the context of an existing DynamoDB table. These are referred to as subresources. These resources and subresources have unique Amazon Resource Names (ARNs) associated with them.

For example, an AWS Account (123456789012) has a DynamoDB table named *Books* in the US East (N. Virginia) (us-east-1) region. The ARN of the *Books* table would be:

```
arn:aws:dynamodb:us-east-1:123456789012:table/Books
```

A policy is an entity that, when attached to an identity or resource, defines their permissions. By using an IAM policy and role to control access, it will grant a Lambda function access to a DynamoDB table.

Create role

1 Trust 2 Permissions 3 Review

Review

Provide the required information below and review this role before you create it.

Role name*	MyLambdaRole
Maximum 64 characters. Use alphanumeric and '+-=,@-_' characters.	
Role description	
Maximum 1000 characters. Use alphanumeric and '+-=,@-_' characters.	
Trusted entities	AWS service: lambda.amazonaws.com
Policies	MyLambdaPolicy 

* Required

[Cancel](#) [Previous](#) [Create role](#)

It is stated in the scenario that a Lambda function will be used to modify the DynamoDB table named **tutorialsdojo**. Since you only need to access one table, you will need to indicate that table in the resource element of the IAM policy. Also, you must specify the effect and action elements that will be generated in the policy.

Hence, the correct answer in this scenario is:

```
{  
  "version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "TutorialsdojoTablePolicy",  
      "Effect": "Allow",  
      "Action": [  
        "dynamodb:PutItem",  
        "dynamodb>DeleteItem"  
      ],  
      "Resource": "arn:aws:dynamodb:us-east-1:120618981206:table/tutorialsdojo"  
    }  
  ]  
}
```

The IAM policy below is incorrect because the scenario only requires you to allow the permissions in the `tutorialsdojo` table. Having a wildcard: `table/*` in this policy would allow the Lambda function to modify all the DynamoDB tables in your account.

```
{  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "TutorialsdojoTablePolicy",  
      "Effect": "Allow",  
      "Action": [  
        "dynamodb:PutItem"  
      ]  
    }  
  ]  
}
```

```

        "dynamodb>DeleteItem"
    ],
    "Resource": "arn:aws:dynamodb:us-east-1:120618981206:table/*"
}
]
}

```

The IAM policy below is incorrect. The first statement is correctly allowing PUT and DELETE actions to the *tutorialsdojo* DynamoDB table. However, the second statement counteracts the first one as it allows all DynamoDB actions in the *tutorialsdojo* table.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialsdojoTablePolicy1",
            "Effect": "Allow",
            "Action": [ "dynamodb:PutItem", "dynamodb>DeleteItem" ],
            "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"
        },
        {
            "Sid": "TutorialsdojoTablePolicy2",
            "Effect": "Allow",
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"
        }
    ]
}

```

The IAM policy below is incorrect. Just like the previous option, the first statement of this policy is correctly allowing PUT and DELETE actions to the *tutorialsdojo* DynamoDB table. However, the second statement counteracts the first one as it denies all DynamoDB actions. Therefore, this policy will not allow any actions on all DynamoDB tables of the AWS account.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TutorialsdojoTablePolicy1",
            "Effect": "Allow",
            "Action": [
                "dynamodb:PutItem",
                "dynamodb>DeleteItem"
            ],
            "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/tutorialsdojo"
        },
        {
            "Sid": "TutorialsdojoTablePolicy2",
            "Effect": "Deny",
            "Action": "dynamodb:*",
            "Resource": "arn:aws:dynamodb:us-east-1:1206189812061898:table/*"
        }
    ]
}

```

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/using-identity-based-policies.html>

https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_examples_lambda-access-dynamodb.html

<https://aws.amazon.com/blogs/security/how-to-create-an-aws-iam-policy-to-grant-aws-lambda-access-to-an-amazon-dynamodb-table/>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

18. QUESTION

Category: CSAA – Design High-Performing Architectures

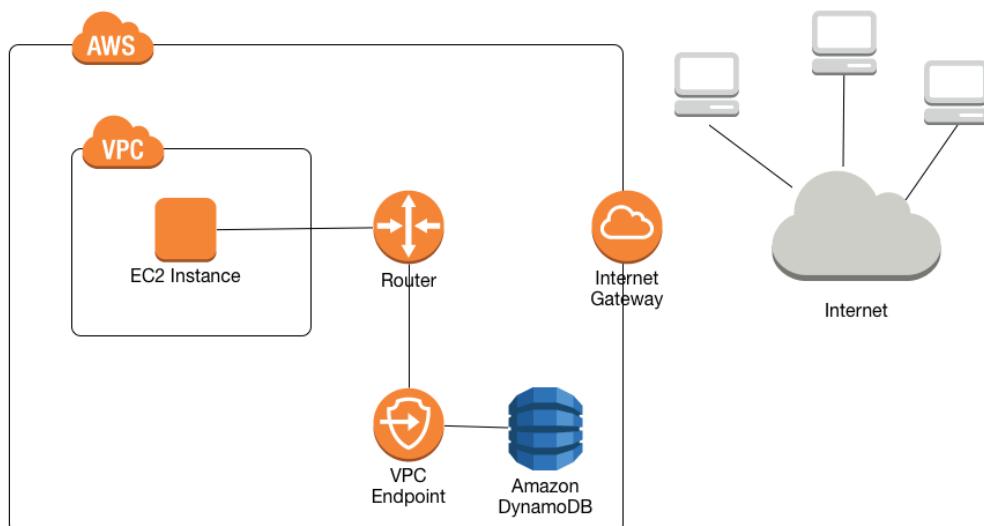
A company plans to launch an Amazon EC2 instance in a private subnet for its internal corporate web portal. For security purposes, the EC2 instance must send data to Amazon DynamoDB and Amazon S3 via private endpoints that don't pass through the public Internet.

Which of the following can meet the above requirements?

- Use AWS VPN CloudHub to route all access to S3 and DynamoDB via private endpoints.
- Use AWS Transit Gateway to route all access to S3 and DynamoDB via private endpoints.
- Use VPC endpoints to route all access to S3 and DynamoDB via private endpoints.
- Use AWS Direct Connect to route all access to S3 and DynamoDB via private endpoints.

Correct

A **VPC endpoint** allows you to privately connect your VPC to supported AWS and VPC endpoint services powered by AWS PrivateLink without needing an Internet gateway, NAT computer, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.



In the scenario, you are asked to configure private endpoints to send data to Amazon DynamoDB and Amazon S3 without accessing the public Internet. Among the options given, VPC endpoint is the most suitable service that will allow you to use private IP addresses to access both DynamoDB and S3 without any exposure to the public internet.

Hence, the correct answer is the option that says: ***Use VPC endpoints to route all access to S3 and DynamoDB via private endpoints.***

The option that says: ***Use AWS Transit Gateway to route all access in S3 and DynamoDB to a public endpoint*** is incorrect because a Transit Gateway simply connects your VPC and on-premises networks through a central hub. It acts as a cloud router that allows you to integrate multiple networks.

The option that says: ***Use AWS Direct Connect to route all access to S3 and DynamoDB via private endpoints*** is incorrect because AWS Direct Connect is primarily used to establish a dedicated network connection from your premises to AWS. The scenario didn't say that the company is using its on-premises server or has a hybrid cloud architecture.

The option that says: ***Use AWS VPN CloudHub to route all access in S3 and DynamoDB to a private endpoint*** is incorrect because AWS VPN CloudHub is mainly used to provide secure communication between remote sites and not for creating a private endpoint to access Amazon S3 and DynamoDB within the Amazon network.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/vpc-endpoints-dynamodb.html>

<https://docs.aws.amazon.com/glue/latest/dg/vpc-endpoints-s3.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

19. QUESTION

Category: CSAA – Design Resilient Architectures

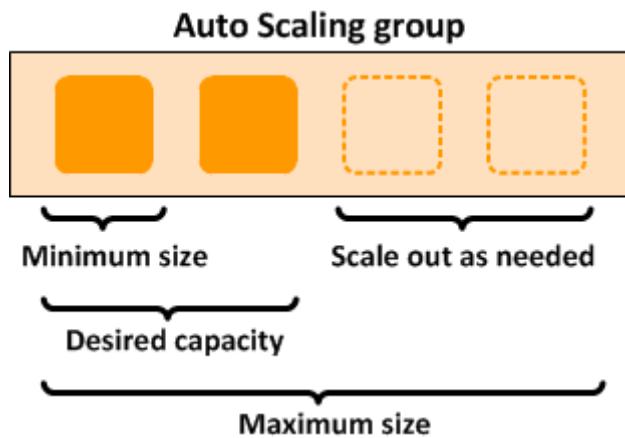
A company needs to deploy at least 2 EC2 instances to support the normal workloads of its application and automatically scale up to 6 EC2 instances to handle the peak load. The architecture must be highly available and fault-tolerant as it is processing mission-critical workloads.

As the Solutions Architect of the company, what should you do to meet the above requirement?

- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A.
- **Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.**

Correct

Amazon EC2 Auto Scaling helps ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can also specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.



To achieve highly available and fault-tolerant architecture for your applications, you must deploy all your instances in different Availability Zones. This will help you isolate your resources if an outage occurs. Take note that to achieve fault tolerance, you need to have redundant resources in place to avoid any system degradation in the event of a server fault or an Availability Zone outage. Having a fault-tolerant architecture entails an extra cost in running additional resources than what is usually needed. This is to ensure that the mission-critical workloads are processed.

Since the scenario requires at least 2 instances to handle regular traffic, you should have 2 instances running all the time even if an AZ outage occurred. You can use an Auto Scaling Group to automatically scale your compute resources across two or more Availability Zones. You have to specify the minimum capacity to 4 instances and the maximum capacity to 6 instances. If each AZ has 2 instances running, even if an AZ fails, your system will still run a minimum of 2 instances.

Hence, the correct answer in this scenario is: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.***

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A*** is incorrect because the instances are only deployed in a single Availability Zone. It cannot protect your applications and data from datacenter or AZ failures.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ*** is incorrect. It is required to have 2 instances running all the time. If an AZ outage happened, ASG will launch a new instance on the unaffected AZ. This provisioning does not happen instantly, which means that for a certain period of time, there will only be 1 running instance left.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B*** is incorrect. Although this fulfills the requirement of at least 2 EC2 instances and high availability, the maximum capacity setting is wrong. It should be set to 6 to properly handle the peak load. If an AZ outage occurs and the system is at its peak load, the number of running instances in this setup will only be 4 instead of 6 and this will affect the performance of your application.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html>

<https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

20. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A newly hired Solutions Architect is assigned to manage a set of CloudFormation templates that are used in the company's cloud architecture in AWS. The Architect accessed the templates and tried to analyze the configured IAM policy for an S3 bucket.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:Get*",  
                "s3>List*"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::boracay/*"  
        }  
    ]  
}
```

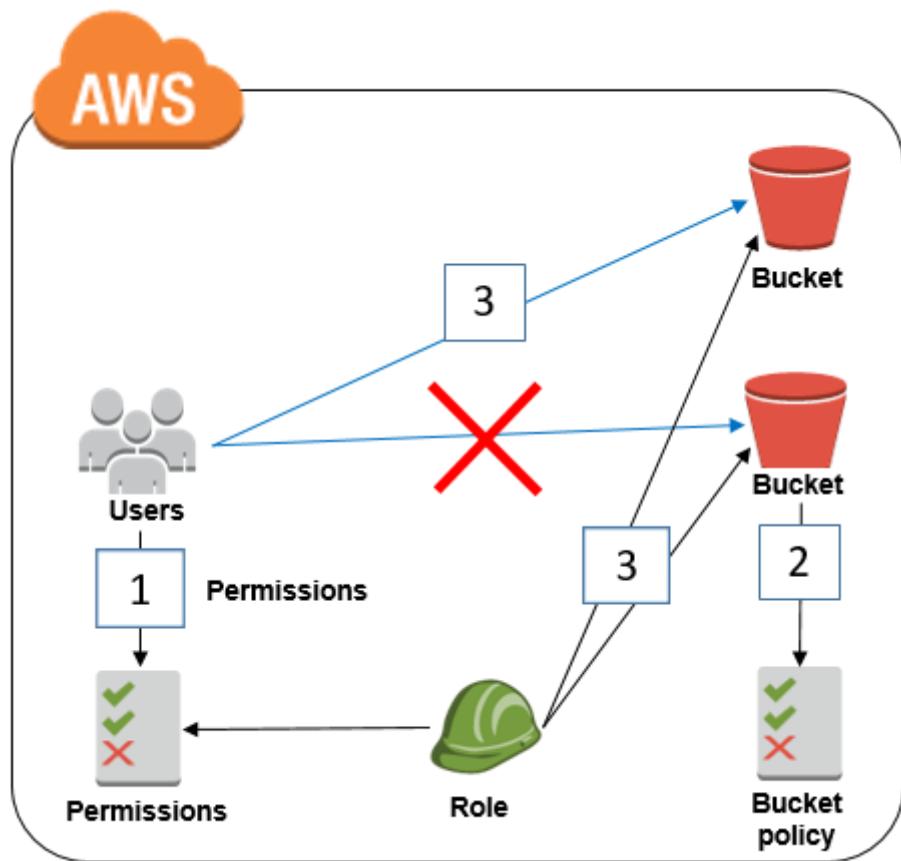
What does the above IAM policy allow? (Select THREE.)

- An IAM user with this IAM policy is allowed to read objects in the `boracay` S3 bucket but not allowed to list the objects in the bucket.
- An IAM user with this IAM policy is allowed to change access rights for the `boracay` S3 bucket.
- An IAM user with this IAM policy is allowed to write objects into the `boracay` S3 bucket.
- An IAM user with this IAM policy is allowed to read objects from all S3 buckets owned by the account.
- An IAM user with this IAM policy is allowed to read and delete objects from the `boracay` S3 bucket.
- An IAM user with this IAM policy is allowed to read objects from the `boracay` S3 bucket.

Correct

You manage access in AWS by creating policies and attaching them to IAM identities (users, groups of users, or roles) or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when an IAM principal (user or role) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. AWS supports six types of policies: identity-based policies, resource-based policies, permissions boundaries, AWS Organizations SCPs, ACLs, and session policies.

IAM policies define permissions for action regardless of the method that you use to perform the operation. For example, if a policy allows the [GetUser](#) action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API. When you create an IAM user, you can choose to allow console or programmatic access. If console access is allowed, the IAM user can sign in to the console using a user name and password. Or if programmatic access is allowed, the user can use access keys to work with the CLI or API.



Based on the provided IAM policy, the user is only allowed to get, write, and list all of the objects for the `boracay` S3 bucket. The `s3:Putobject` basically means that you can submit a PUT object request to the S3 bucket to store data.

Hence, the correct answers are:

- *- An IAM user with this IAM policy is allowed to read objects from all S3 buckets owned by the account.*
- *- An IAM user with this IAM policy is allowed to write objects into the `boracay` S3 bucket.*
- *- An IAM user with this IAM policy is allowed to read objects from the `boracay` S3 bucket.*

The option that says: *An IAM user with this IAM policy is allowed to change access rights for the `boracay` S3 bucket* is incorrect because the template does not have any statements which allow the user to change access rights in the bucket.

The option that says: *An IAM user with this IAM policy is allowed to read objects in the `boracay` S3 bucket but not allowed to list the objects in the bucket* is incorrect because it can clearly be seen in the template that there is a `s3>List*` which permits the user to list objects.

The option that says: *An IAM user with this IAM policy is allowed to read and delete objects from the `boracay` S3 bucket* is incorrect. Although you can read objects from the bucket, you cannot delete any objects.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectOps.html>

https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies.html

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

21. QUESTION

Category: CSAA – Design High-Performing Architectures

A popular social network is hosted in AWS and is using a DynamoDB table as its database. There is a requirement to implement a ‘follow’ feature where users can subscribe to certain updates made by a particular user and be notified via email.

Which of the following is the most suitable solution that you should implement to meet the requirement?

- Enable DynamoDB Stream and create an AWS Lambda trigger, as well as the IAM role which contains all of the permissions that the Lambda function will need at runtime. The data from the stream record will be processed by the Lambda function which will then publish a message to SNS Topic that will notify the subscribers via email.
- Set up a DAX cluster to access the source DynamoDB table. Create a new DynamoDB trigger and a Lambda function. For every update made in the user data, the trigger will send data to the Lambda function which will then notify the subscribers via email using SNS.
- Create a Lambda function that uses DynamoDB Streams Kinesis Adapter which will fetch data from the DynamoDB Streams endpoint. Set up an SNS Topic that will notify the subscribers via email when there is an update made by a particular user.
- Using the Kinesis Client Library (KCL), write an application that leverages on DynamoDB Streams Kinesis Adapter that will fetch data from the DynamoDB Streams endpoint. When there are updates made by a particular user, notify the subscribers via email using SNS.

Correct

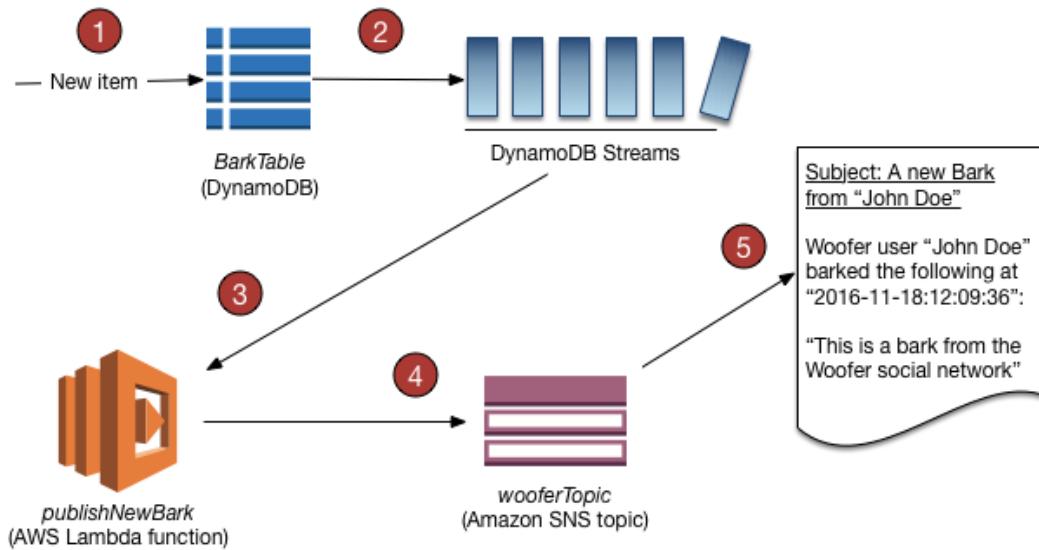
A **DynamoDB stream** is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A *stream record* contains information about a data modification to a single item in a DynamoDB table. You can configure the stream so that the stream records capture additional information, such as the “before” and “after” images of modified items.

Amazon DynamoDB is integrated with AWS Lambda so that you can create *triggers*—pieces of code that automatically respond to events in DynamoDB Streams. With triggers, you can build applications that react to data modifications in DynamoDB tables.

If you enable DynamoDB Streams on a table, you can associate the stream ARN with a Lambda function that you write. Immediately after an item in the table is modified, a new record appears in the table’s stream. AWS Lambda polls the stream and invokes your Lambda function synchronously when it detects new stream records. The Lambda function can perform any actions you specify, such as sending a notification or initiating a workflow.

Hence, the correct answer in this scenario is the option that says: ***Enable DynamoDB Stream and create an AWS Lambda trigger, as well as the IAM role which contains all of the permissions that the Lambda function will need at runtime. The data from the stream record will be processed by the Lambda function which will then publish a message to SNS Topic that will notify the subscribers via email*.**



The option that says: ***Using the Kinesis Client Library (KCL), write an application that leverages on DynamoDB Streams Kinesis Adapter that will fetch data from the DynamoDB Streams endpoint. When there are updates made by a particular user, notify the subscribers via email using SNS*** is incorrect because although this is a valid solution, it is missing a vital step which is to enable DynamoDB Streams. With the DynamoDB Streams Kinesis Adapter in place, you can begin developing applications via the KCL interface, with the API calls seamlessly directed at the DynamoDB Streams endpoint. Remember that the DynamoDB Stream feature is not enabled by default.

The option that says: ***Create a Lambda function that uses DynamoDB Streams Kinesis Adapter which will fetch data from the DynamoDB Streams endpoint. Set up an SNS Topic that will notify the subscribers via email when there is an update made by a particular user*** is incorrect because just like in the above, you have to manually enable DynamoDB Streams first before you can use its endpoint.

The option that says: ***Set up a DAX cluster to access the source DynamoDB table. Create a new DynamoDB trigger and a Lambda function. For every update made in the user data, the trigger will send data to the Lambda function which will then notify the subscribers via email using SNS*** is incorrect because the DynamoDB Accelerator (DAX) feature is primarily used to significantly improve the in-memory read performance of your database, and not to capture the time-ordered sequence of item-level modifications. You should use DynamoDB Streams in this scenario instead.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.html>

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Streams.Lambda.Tutorial.html>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

22. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A financial application is composed of an Auto Scaling group of EC2 instances, an Application Load Balancer, and a MySQL RDS instance in a Multi-AZ Deployments configuration. To protect the confidential data of your customers, you have to ensure that your RDS database can only be accessed using the profile credentials specific to your EC2 instances via an authentication token.

As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- Use a combination of IAM and STS to restrict access to your RDS instance via a temporary token.

- Create an IAM Role and assign it to your EC2 instances which will grant exclusive access to your RDS instance.
- Configure SSL in your application to encrypt the database connection to RDS.
- **Enable the IAM DB Authentication.**

Incorrect

You can authenticate to your DB instance using AWS Identity and Access Management (IAM) database authentication. IAM database authentication works with MySQL and PostgreSQL. With this authentication method, you don't need to use a password when you connect to a DB instance. Instead, you use an authentication token.

An **authentication token** is a unique string of characters that Amazon RDS generates on request. Authentication tokens are generated using AWS Signature Version 4. Each token has a lifetime of 15 minutes. You don't need to store user credentials in the database, because authentication is managed externally using IAM. You can also still use standard database authentication.

Database options

DB cluster identifier [Info](#)
tutorialsdojo

If you do not provide one, a default identifier based on the instance identifier will be used.

Database name [Info](#)
tutorialsdojo

If you do not specify a database name, Amazon RDS does not create a database.

Port [Info](#)
TCP/IP port the DB instance will use for application connections.
3306

DB parameter group [Info](#)
default.aurora5.6

DB cluster parameter group [Info](#)
default.aurora5.6

Option group [Info](#)
default:aurora-5-6

IAM DB authentication [Info](#)

Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.

Disable

IAM database authentication provides the following benefits:

1. Network traffic to and from the database is encrypted using Secure Sockets Layer (SSL).
2. You can use IAM to centrally manage access to your database resources, instead of managing access individually on each DB instance.

3. For applications running on Amazon EC2, you can use profile credentials specific to your EC2 instance to access your database instead of a password, for greater security

Hence, ***enabling IAM DB Authentication*** is the correct answer based on the above reference.

Configuring SSL in your application to encrypt the database connection to RDS is incorrect because an SSL connection is not using an authentication token from IAM. Although configuring SSL to your application can improve the security of your data in flight, it is still not a suitable option to use in this scenario.

Creating an IAM Role and assigning it to your EC2 instances which will grant exclusive access to your RDS instance is incorrect because although you can create and assign an IAM Role to your EC2 instances, you still need to configure your RDS to use IAM DB Authentication.

Using a combination of IAM and STS to restrict access to your RDS instance via a temporary token is incorrect because you have to use IAM DB Authentication for this scenario, and not a combination of an IAM and STS. Although STS is used to send temporary tokens for authentication, this is not a compatible use case for RDS.

Reference:

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.IAMDBAuth.html>

Check out this Amazon RDS cheat sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

23. QUESTION

Category: CSAA – Design Resilient Architectures

A company conducted a surprise IT audit on all of the AWS resources being used in the production environment. During the audit activities, it was noted that you are using a combination of Standard and Convertible Reserved EC2 instances in your applications. They argued that you should have used Spot EC2 instances instead as it is cheaper than the Reserved Instance.

Which of the following are the characteristics and benefits of using these two types of Reserved EC2 instances, which you can use as justification? (Select TWO.)

- It runs in a VPC on hardware that's dedicated to a single customer.
- It can enable you to reserve capacity for your Amazon EC2 instances in multiple Availability Zones and multiple AWS Regions for any duration.
- Standard Reserved Instances can be later exchanged for other Convertible Reserved Instances
- Reserved Instances doesn't get interrupted unlike Spot instances in the event that there are not enough unused EC2 instances to meet the demand.
- Convertible Reserved Instances allow you to exchange for another Convertible Reserved instance with a different instance type and tenancy.

Incorrect

Reserved Instances (RIs) provide you with a significant discount (up to 75%) compared to On-Demand instance pricing. You have the flexibility to change families, OS types, and tenancies while benefiting from RI pricing when you use Convertible RIs. One important thing to remember here is that Reserved Instances are not physical instances, but rather a billing discount applied to the use of On-Demand Instances in your account.

The offering class of a Reserved Instance is either Standard or Convertible. A **Standard Reserved Instance** provides a more significant discount than a **Convertible Reserved Instance**, but you can't exchange a Standard Reserved Instance unlike Convertible Reserved Instances. You can modify Standard and Convertible Reserved Instances. Take note that in Convertible Reserved Instances, you are allowed to exchange another Convertible Reserved instance with a different instance type and tenancy.

The configuration of a Reserved Instance comprises a single instance type, platform, scope, and tenancy over a term. If your computing needs change, you might be able to modify or exchange your Reserved Instance.

When your computing needs change, you can modify your Standard or Convertible Reserved Instances and continue to take advantage of the billing benefit. You can modify the Availability Zone, scope, network platform, or instance size (within the same instance type) of your Reserved Instance. You can also sell your unused instance on the Reserved Instance Marketplace.

Hence, the correct options are:

- Reserved Instances don't get interrupted unlike Spot instances in the event that there are not enough unused EC2 instances to meet the demand

- Convertible Reserved Instances allows you to exchange for another Convertible Reserved instance with a different instance type and tenancy.

The option that says: ***Standard Reserved Instances can be later exchanged for other Convertible Reserved Instances*** is incorrect because only Convertible Reserved Instances can be exchanged for other Convertible Reserved Instances.

The option that says: ***It can enable you to reserve capacity for your Amazon EC2 instances in multiple Availability Zones and multiple AWS Regions for any duration*** is incorrect because **you can reserve capacity to a specific AWS Region (regional Reserved Instance) or specific Availability Zone (zonal Reserved Instance)** only. You cannot reserve capacity to **multiple AWS Regions** in a single RI purchase.

The option that says: ***It runs in a VPC on hardware that's dedicated to a single customer*** is incorrect because that is the description of a Dedicated instance and not a Reserved Instance. A Dedicated instance runs in a VPC on hardware that's dedicated to a single customer.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ri-modifying.html>

<https://aws.amazon.com/ec2/pricing/reserved-instances/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-reserved-instances.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/reserved-instances-types.html>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

24. QUESTION

Category: CSAA – Design High-Performing Architectures

A global IT company with offices around the world has multiple AWS accounts. To improve efficiency and drive costs down, the Chief Information Officer (CIO) wants to set up a solution that centrally manages their AWS resources. This will allow them to procure AWS resources centrally and share resources such as AWS Transit Gateways, AWS License Manager configurations, or Amazon Route 53 Resolver rules across their various accounts.

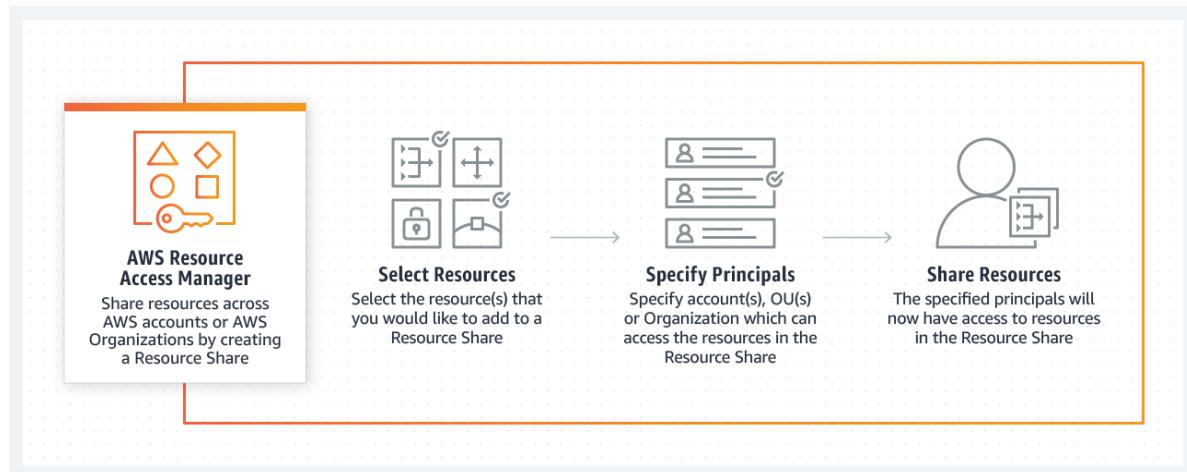
As the Solutions Architect, which combination of options should you implement in this scenario? (Select TWO.)

- Use the AWS Identity and Access Management service to set up cross-account access that will easily and securely share your resources with your AWS accounts.
- Use AWS Control Tower to easily and securely share your resources with your AWS accounts.
- **Consolidate all of the company's accounts using AWS Organizations.**
- **Use the AWS Resource Access Manager (RAM) service to easily and securely share your resources with your AWS accounts.**
- Consolidate all of the company's accounts using AWS ParallelCluster.

Incorrect

AWS Resource Access Manager (RAM) is a service that enables you to easily and securely share AWS resources with any AWS account or within your AWS Organization. You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM.

Many organizations use multiple accounts to create administrative or billing isolation, and limit the impact of errors. RAM eliminates the need to create duplicate resources in multiple accounts, reducing the operational overhead of managing those resources in every single account you own. You can create resources centrally in a multi-account environment, and use RAM to share those resources across accounts in three simple steps: create a Resource Share, specify resources, and specify accounts. RAM is available to you at no additional charge.



You can procure AWS resources centrally, and use RAM to share resources such as subnets or License Manager configurations with other accounts. This eliminates the need to provision duplicate resources in every account in a multi-account environment, reducing the operational overhead of managing those resources in every account.

AWS Organizations is an account management service that lets you consolidate multiple AWS accounts into an organization that you create and centrally manage. With Organizations, you can create member accounts and invite existing accounts to join your organization. You can organize those accounts into groups and attach policy-based controls.

Hence, the correct combination of options in this scenario is:

– Consolidate all of the company's accounts using AWS Organizations.

– Use the AWS Resource Access Manager (RAM) service to easily and securely share your resources with your AWS accounts.

The option that says: ***Use the AWS Identity and Access Management service to set up cross-account access that will easily and securely share your resources with your AWS accounts*** is incorrect because although you can delegate access to resources that are in different AWS accounts using IAM, this process is extremely tedious and entails a lot of operational overhead since you have to manually set up cross-account access to each and every AWS account of the company. A better solution is to use AWS Resources Access Manager instead.

The option that says: ***Use AWS Control Tower to easily and securely share your resources with your AWS accounts*** is incorrect because AWS Control Tower simply offers the easiest way to set up and govern a new, secure, multi-account AWS environment. This is not the most suitable service to use to securely share your resources across AWS accounts or within your Organization. You have to use AWS Resources Access Manager (RAM) instead.

The option that says: ***Consolidate all of the company's accounts using AWS ParallelCluster*** is incorrect because AWS ParallelCluster is simply an AWS-supported open-source cluster management tool that makes it easy for you to deploy and manage High-Performance Computing (HPC) clusters on AWS. In this particular scenario, it is more appropriate to use AWS Organizations to consolidate all of your AWS

accounts.

References:

<https://aws.amazon.com/ram/>

<https://docs.aws.amazon.com/ram/latest/userguide/shareable.html>

25. QUESTION

Category: CSAA – Design Secure Applications and Architectures

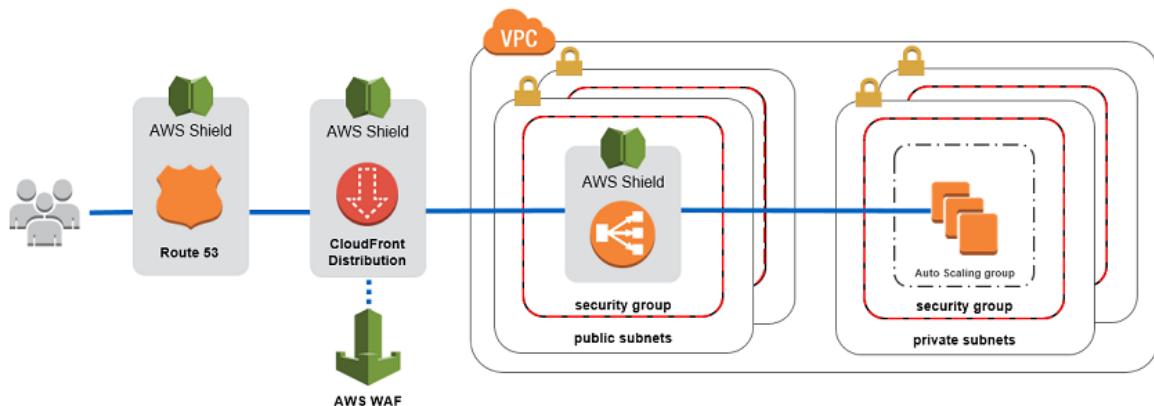
A Solutions Architect identified a series of DDoS attacks while monitoring the VPC. The Architect needs to fortify the current cloud infrastructure to protect the data of the clients.

Which of the following is the most suitable solution to mitigate these kinds of attacks?

- A combination of Security Groups and Network Access Control Lists to only allow authorized traffic to access your VPC.
- Set up a web application firewall using AWS WAF to filter, monitor, and block HTTP traffic.
- Using the AWS Firewall Manager, set up a security layer that will prevent SYN floods, UDP reflection attacks, and other DDoS attacks.
- Use AWS Shield Advanced to detect and mitigate DDoS attacks.

Correct

For higher levels of protection against attacks targeting your applications running on Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing(ELB), Amazon CloudFront, and Amazon Route 53 resources, you can subscribe to AWS Shield Advanced. In addition to the network and transport layer protections that come with Standard, AWS Shield Advanced provides additional detection and mitigation against large and sophisticated DDoS attacks, near real-time visibility into attacks, and integration with AWS WAF, a web application firewall.



AWS Shield Advanced also gives you 24x7 access to the AWS DDoS Response Team (DRT) and protection against DDoS related spikes in your Amazon Elastic Compute Cloud (EC2), Elastic Load Balancing(ELB), Amazon CloudFront, and Amazon Route 53 charges.

Hence, the correct answer is: ***Use AWS Shield Advanced to detect and mitigate DDoS attacks.***

The option that says: ***Using the AWS Firewall Manager, set up a security layer that will prevent SYN floods, UDP reflection attacks and other DDoS attacks*** is incorrect because AWS Firewall Manager is mainly used to simplify your AWS WAF administration and maintenance tasks across multiple accounts and resources. It does not protect your VPC against DDoS attacks.

The option that says: ***Set up a web application firewall using AWS WAF to filter, monitor, and block HTTP traffic*** is incorrect. Even though AWS WAF can help you block common attack patterns to your VPC such as SQL injection or cross-site scripting, this is still not enough to withstand DDoS attacks. It is better to use AWS Shield in this scenario.

The option that says: ***A combination of Security Groups and Network Access Control Lists to only allow authorized traffic to access your VPC*** is incorrect. Although using a combination of Security Groups and NACLs are valid to provide security to your VPC, this is not enough to mitigate a DDoS attack. You should use AWS Shield for better security protection.

References:

https://d1.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

<https://aws.amazon.com/shield/>

Check out this AWS Shield Cheat Sheet:

<https://tutorialsdojo.com/aws-shield/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

26. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A media company has an Amazon ECS Cluster, which uses the Fargate launch type, to host its news website. The database credentials should be supplied using environment variables, to comply with strict security compliance. As the Solutions Architect, you have to ensure that the credentials are secure and that they cannot be viewed in plaintext on the cluster itself.

Which of the following is the most suitable solution in this scenario that you can implement with minimal effort?

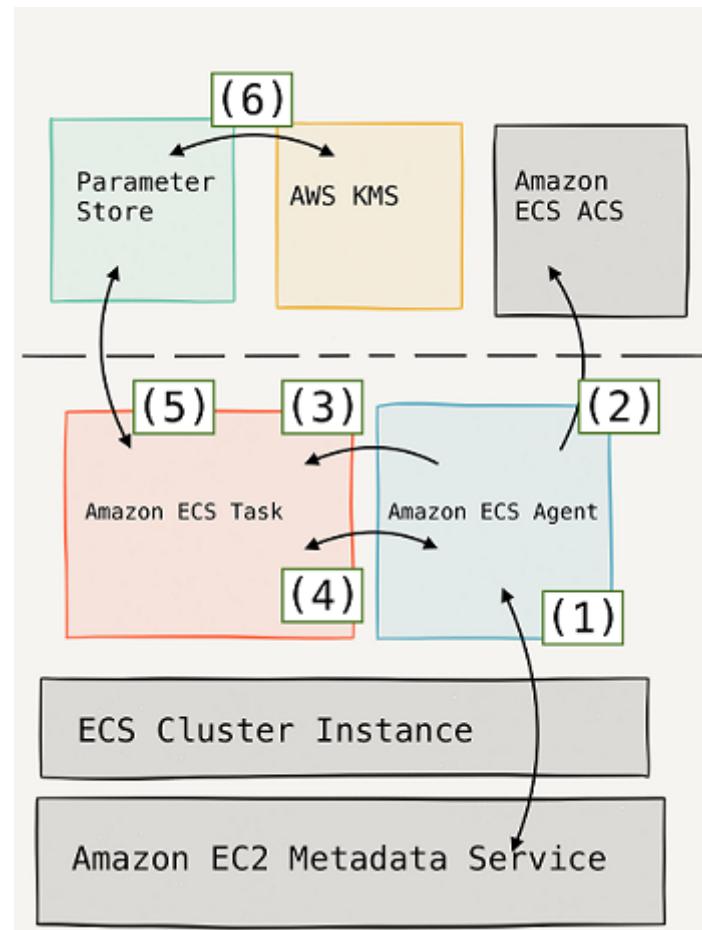
- In the ECS task definition file of the ECS Cluster, store the database credentials using Docker Secrets to centrally manage these sensitive data and securely transmit it to only those containers that need access to it. Secrets are encrypted during transit and at rest. A given secret is only accessible to those services which have been granted explicit access to it via IAM Role, and only while those service tasks are running.
- Store the database credentials in the ECS task definition file of the ECS Cluster and encrypt it with KMS. Store the task definition JSON file in a private S3 bucket and ensure that HTTPS is enabled on the bucket to encrypt the data in-flight. Create an IAM role to the ECS task definition script that allows access to the specific S3 bucket and then pass the `--cli-input-json` parameter when calling the ECS register-task-definition. Reference the task definition JSON file in the S3 bucket which contains the database credentials.
- Use the AWS Secrets Manager to store the database credentials and then encrypt them using AWS KMS. Create a resource-based policy for your Amazon ECS task execution role (`taskRoleArn`) and reference it with your task definition which allows access to both KMS and AWS Secrets Manager. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Secrets Manager secret which contains the sensitive data, to present to the container.
- Use the AWS Systems Manager Parameter Store to keep the database credentials and then encrypt them using AWS KMS. Create an IAM Role for your Amazon ECS task execution role (`taskRoleArn`) and reference it with your task definition, which allows access to both KMS and the Parameter Store. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Systems Manager Parameter Store parameter containing the sensitive data to present to the container.

Incorrect

Amazon ECS enables you to inject sensitive data into your containers by storing your sensitive data in either AWS Secrets Manager secrets or AWS Systems Manager Parameter Store parameters and then referencing them in your container definition. This feature is supported by tasks using both the EC2 and Fargate launch types.

Secrets can be exposed to a container in the following ways:

- To inject sensitive data into your containers as environment variables, use the `secrets` container definition parameter.
- To reference sensitive information in the log configuration of a container, use the `secretoptions` container definition parameter



Within your container definition, specify `secrets` with the name of the environment variable to set in the container and the full ARN of either the Secrets Manager secret or Systems Manager Parameter Store parameter containing the sensitive data to present to the container. The parameter that you reference can be from a different Region than the container using it, but must be from within the same account.

Hence, the correct answer is the option that says: ***Use the AWS Systems Manager Parameter Store to keep the database credentials and then encrypt them using AWS KMS. Create an IAM Role for your Amazon ECS task execution role (`taskRoleArn`) and reference it with your task definition, which allows access to both KMS and the Parameter Store. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Systems Manager Parameter Store parameter containing the sensitive data to present to the container***.

The option that says: ***In the ECS task definition file of the ECS Cluster, store the database credentials using Docker Secrets to centrally manage these sensitive data and securely transmit it to only those containers that need access to it. Secrets are encrypted during transit and at rest. A given secret is only accessible to those services which have been granted explicit access to it via IAM Role, and only while those service tasks are running*** is incorrect because although you can use Docker Secrets to secure the sensitive database credentials, this feature is only applicable in Docker Swarm. In AWS, the recommended way to secure sensitive data is either through the use of Secrets Manager or Systems Manager Parameter Store.

The option that says: ***Store the database credentials in the ECS task definition file of the ECS Cluster and encrypt it with KMS. Store the task definition JSON file in a private S3 bucket and ensure that HTTPS is enabled on the bucket to encrypt the data in-flight. Create an IAM role to the ECS task definition script that allows access to the specific S3 bucket and then pass the `--cli-input-json` parameter when calling the ECS register-task-definition. Reference the task definition JSON file in the S3 bucket which contains the database credentials*** is incorrect because although the solution may work, it is not recommended to store sensitive credentials in S3. This entails a lot of overhead and manual configuration steps which can be simplified by simply using the Secrets Manager or Systems Manager Parameter Store.

The option that says: ***Use the AWS Secrets Manager to store the database credentials and then encrypt them using AWS KMS. Create a resource-based policy for your Amazon ECS task execution role (`taskRoleArn`) and reference it with your task definition which allows access to both KMS and AWS Secrets Manager. Within your container definition, specify secrets with the name of the environment variable to set in the container and the full ARN of the Secrets Manager secret which contains the sensitive data, to present to the container*** is incorrect because although the use of Secrets Manager in securing sensitive data in ECS is valid, using an IAM Role is a more suitable choice over a resource-based policy for the Amazon ECS task execution role.

References:

<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/specifying-sensitive-data.html>

<https://aws.amazon.com/blogs/mt/the-right-way-to-store-secrets-using-parameter-store/>

Check out these Amazon ECS and AWS Systems Manager Cheat Sheets:

<https://tutorialsdojo.com/amazon-elastic-container-service-amazon-ecs/>

<https://tutorialsdojo.com/aws-systems-manager/>

27. QUESTION

Category: CSAA – Design Resilient Architectures

An application consists of multiple EC2 instances in private subnets in different availability zones. The application uses a single NAT Gateway for downloading software patches from the Internet to the instances. There is a requirement to protect the application from a single point of failure when the NAT Gateway encounters a failure or if its availability zone goes down.

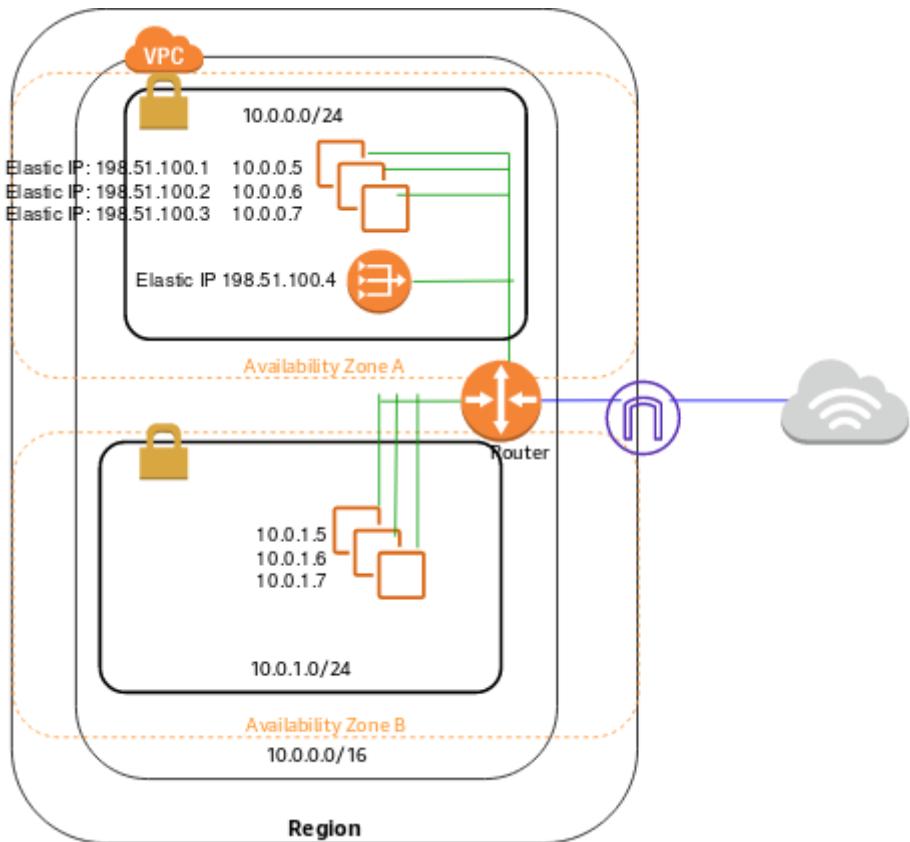
How should the Solutions Architect redesign the architecture to be more highly available and cost-effective?

- Create three NAT Gateways in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone.
- **Create a NAT Gateway in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone**
- Create a NAT Gateway in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone.
- Create two NAT Gateways in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone.

Incorrect

A **NAT Gateway** is a highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet. NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone.

You must create a NAT gateway on a public subnet to enable instances in a private subnet to connect to the Internet or other AWS services, but prevent the Internet from initiating a connection with those instances.



If you have resources in multiple Availability Zones and they share one NAT gateway, and if the NAT gateway's Availability Zone is down, resources in the other Availability Zones lose Internet access. To create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.

Hence, the correct answer is: ***Create a NAT Gateway in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone***.

The option that says: ***Create a NAT Gateway in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone*** is incorrect because you should configure the route table in the private subnet and not the public subnet to associate the right instances in the private subnet.

The options that say: ***Create two NAT Gateways in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone*** and ***Create three NAT Gateways in each availability zone. Configure the route table in each private subnet to ensure that instances use the NAT Gateway in the same availability zone*** are both incorrect because a single NAT Gateway in each availability zone is enough. NAT Gateway is already redundant in nature, meaning, AWS already handles any failures that occur in your NAT Gateway in an availability zone.

References:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

28. QUESTION

Category: CSAA – Design High-Performing Architectures

A Docker application, which is running on an Amazon ECS cluster behind a load balancer, is heavily using DynamoDB. You are instructed to improve the database performance by distributing the workload evenly and using the provisioned throughput efficiently.

Which of the following would you consider to implement for your DynamoDB table?

- Reduce the number of partition keys in the DynamoDB table.
- Use partition keys with high-cardinality attributes, which have a large number of distinct values for each item.
- Use partition keys with low-cardinality attributes, which have a few number of distinct values for each item.
- Avoid using a composite primary key, which is composed of a partition key and a sort key.

Incorrect

The partition key portion of a table's primary key determines the logical partitions in which a table's data is stored. This in turn affects the underlying physical partitions. Provisioned I/O capacity for the table is divided evenly among these physical partitions. Therefore a partition key design that doesn't distribute I/O requests evenly can create "hot" partitions that result in throttling and use your provisioned I/O capacity inefficiently.

The optimal usage of a table's provisioned throughput depends not only on the workload patterns of individual items, but also on the partition-key design. This doesn't mean that you must access all partition key values to achieve an efficient throughput level, or even that the percentage of accessed partition key values must be high. It does mean that the more distinct partition key values that your workload accesses, the more those requests will be spread across the partitioned space. In general, you will use your provisioned throughput more efficiently as the ratio of partition key values accessed to the total number of partition key values increases.

One example for this is the use of ***partition keys with high-cardinality attributes, which have a large number of distinct values for each item***.

Reducing the number of partition keys in the DynamoDB table is incorrect. Instead of doing this, you should actually add more to improve its performance to distribute the I/O requests evenly and not avoid "hot" partitions.

Using partition keys with low-cardinality attributes, which have a few number of distinct values for each item is incorrect because this is the exact opposite of the correct answer. Remember that the more distinct partition key values your workload accesses, the more those requests will be spread across the partitioned space. Conversely, the less distinct partition key values, the less evenly spread it would be across the partitioned space, which effectively slows the performance.

The option that says: ***Avoid using a composite primary key, which is composed of a partition key and a sort key*** is incorrect because as mentioned, a composite primary key will provide more partition for the table and in turn, improves the performance. Hence, it should be used and not avoided.

References:

<https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-uniform-load.html>

<https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Amazon DynamoDB Overview:

<https://youtu.be/3ZOyUNleorU>

29. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application is using CloudFront to distribute their images, videos, and other static contents stored in their S3 bucket to its users around the world. The company has recently introduced a new member-only access to some of its high quality media files. There is a requirement to provide access to multiple private media files only to their paying subscribers without having to change their current URLs.

Which of the following is the most suitable solution that you should implement to satisfy this requirement?

- Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member.
- Create a Signed URL with a custom policy which only allows the members to see the private files.
- Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members.
- **Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required `Set-Cookie` headers to the viewer which will unlock the content only to them.**

Correct

CloudFront signed URLs and signed cookies provide the same basic functionality: they allow you to control who can access your content. If you want to serve private content through CloudFront and you're trying to decide whether to use signed URLs or signed cookies, consider the following:

Use **signed URLs** for the following cases:

- You want to use an RTMP distribution. Signed cookies aren't supported for RTMP distributions.
- You want to restrict access to individual files, for example, an installation download for your application.
- Your users are using a client (for example, a custom HTTP client) that doesn't support cookies.

Use **signed cookies** for the following cases:

- You want to provide access to multiple restricted files, for example, all of the files for a video in HLS format or all of the files in the subscribers' area of a website.
- You don't want to change your current URLs.

Hence, the correct answer for this scenario is the option that says: ***Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required `Set-Cookie` headers to the viewer which will unlock the content only to them.***

The option that says: ***Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member*** is incorrect because a Match Viewer is an Origin Protocol Policy which configures CloudFront to communicate with your origin using HTTP or HTTPS, depending on the protocol of the viewer request. CloudFront caches the object only once even if viewers make requests using both HTTP and HTTPS protocols.

The option that says: ***Create a Signed URL with a custom policy which only allows the members to see the private files*** is incorrect because Signed URLs are primarily used for providing access to individual files, as shown on the above explanation. In addition, the scenario explicitly says that they don't want to change their current URLs which is why implementing Signed Cookies is more suitable than Signed URL.

The option that says: ***Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members*** is incorrect because Field-Level Encryption only allows you to securely upload user-submitted sensitive information to your web servers. It does not provide access to download multiple private files.

Reference:

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-choosing-signed-urls-cookies.html>

<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-signed-cookies.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

30. QUESTION

Category: CSAA – Design Resilient Architectures

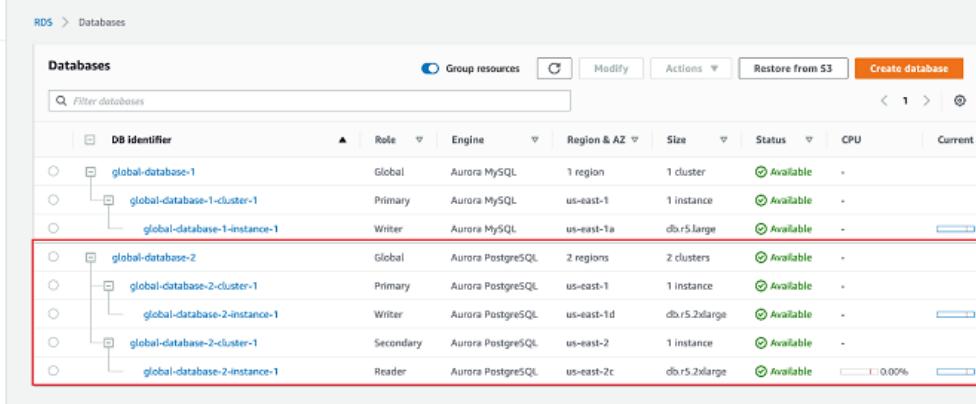
A Solutions Architect needs to set up a relational database and come up with a disaster recovery plan to mitigate multi-region failure. The solution requires a Recovery Point Objective (RPO) of 1 second and a Recovery Time Objective (RTO) of less than 1 minute.

Which of the following AWS services can fulfill this requirement?

- AWS Global Accelerator
- Amazon RDS for PostgreSQL with cross-region read replicas
- Amazon Aurora Global Database
- Amazon DynamoDB global tables

Correct

Amazon Aurora Global Database is designed for globally distributed applications, allowing a single Amazon Aurora database to span multiple AWS regions. It replicates your data with no impact on database performance, enables fast local reads with low latency in each region, and provides disaster recovery from region-wide outages.



The screenshot shows the AWS RDS Databases console. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query Editor, Performance Insights, Snapshots, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Events, Event subscriptions, Recommendations, and Certificate update. The main area is titled 'Databases' and shows two global databases: 'global-database-1' and 'global-database-2'. Each database has multiple instances: 'global-database-1' has 'global-database-1-cluster-1' (Primary, Aurora MySQL, us-east-1) and 'global-database-1-instance-1' (Writer, Aurora MySQL, us-east-1a); 'global-database-2' has 'global-database-2-cluster-1' (Primary, Aurora PostgreSQL, us-east-1) and 'global-database-2-instance-1' (Writer, Aurora PostgreSQL, us-east-1d). There are also 'global-database-1-cluster-1' and 'global-database-2-cluster-1' listed under their respective databases. The table includes columns for DB Identifier, Role, Engine, Region & AZ, Size, Status, CPU, and Current active. A red box highlights the instances under 'global-database-2'.

Aurora Global Database supports storage-based replication that has a latency of less than 1 second. If there is an unplanned outage, one of the secondary regions you assigned can be promoted to read and write capabilities in less than 1 minute. This feature is called Cross-Region Disaster Recovery. An RPO of 1 second and an RTO of less than 1 minute provides you a strong foundation for a global business continuity plan.

Hence, the correct answer is: ***Amazon Aurora Global Database***.

Amazon DynamoDB global tables is incorrect because it is stated in the scenario that the Solutions Architect needs to create a relational database and not a NoSQL database. When you create a DynamoDB global table, it consists of multiple replica tables (one per AWS Region) that DynamoDB treats as a single unit.

Multi-AZ Amazon RDS database with cross-region read replicas is incorrect because a Multi-AZ deployment is only applicable inside a single region and not in a multi-region setup. This database setup is not capable of providing an RPO of 1 second and an RTO of less than 1 minute. Moreover, the replication of cross-region RDS Read Replica is not as fast compared with Amazon Aurora Global Databases.

AWS Global Accelerator is incorrect because this is a networking service that simplifies traffic management and improves application performance. AWS Global Accelerator is not a relational database service; therefore, this is not a suitable service to use in this scenario.

References:

<https://aws.amazon.com/rds/aurora/global-database/>

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html>

Check out this Amazon Aurora Cheat Sheet:

<https://tutorialsdojo.com/amazon-aurora/>

31. QUESTION

Category: CSAA – Design Resilient Architectures

An online cryptocurrency exchange platform is hosted in AWS which uses ECS Cluster and RDS in Multi-AZ Deployments configuration. The application is heavily using the RDS instance to process complex read and write database operations. To maintain the reliability, availability, and performance of your systems, you have to closely monitor how the different processes or threads on a DB instance use the CPU, including the percentage of the CPU bandwidth and total memory consumed by each process.

Which of the following is the most suitable solution to properly monitor your database?

- Use Amazon CloudWatch to monitor the CPU Utilization of your database.
- Create a script that collects and publishes custom metrics to CloudWatch, which tracks the real-time CPU Utilization of the RDS instance, and then set up a custom CloudWatch dashboard to view the metrics.
- Check the `CPU%` and `MEM%` metrics which are readily available in the Amazon RDS console that shows the percentage of the CPU bandwidth and total memory consumed by each database process of your RDS instance.
- **Enable Enhanced Monitoring in RDS.**

Incorrect

Amazon RDS provides metrics in real time for the operating system (OS) that your DB instance runs on. You can view the metrics for your DB instance using the console, or consume the Enhanced Monitoring JSON output from CloudWatch Logs in a monitoring system of your choice. By default, Enhanced Monitoring metrics are stored in the CloudWatch Logs for 30 days. To modify the amount of time the metrics are stored in the CloudWatch Logs, change the retention for the `RDSOSMetrics` log group in the CloudWatch console.

Take note that there are certain differences between CloudWatch and Enhanced Monitoring Metrics. CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance, and Enhanced Monitoring gathers its metrics from an agent on the instance. As a result, you might find differences between the measurements, because the hypervisor layer performs a small amount of work. Hence, ***enabling Enhanced Monitoring in RDS*** is the correct answer in this specific scenario.

The differences can be greater if your DB instances use smaller instance classes, because then there are likely more virtual machines (VMs) that are managed by the hypervisor layer on a single physical instance. Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU.

Process List						
<input type="text"/> Filter process list						
NAME	VIRT	RES	CPU%	MEM%	VMLIMIT	
postgres [3181]`	283.55 MB	17.11 MB	0.02	1.72		
postgres: rdsadmin rdsadmin localhost(40156) idle [2953]`	384.7 MB	9.51 MB	0.02	0.95		

Using Amazon CloudWatch to monitor the CPU Utilization of your database is incorrect because although you can use this to monitor the CPU Utilization of your database instance, it does not provide the percentage of the CPU bandwidth and total memory consumed by each database process in your RDS instance. Take note that CloudWatch gathers metrics about CPU utilization from the hypervisor for a DB instance while RDS Enhanced Monitoring gathers its metrics from an agent on the instance.

The option that says: *Create a script that collects and publishes custom metrics to CloudWatch, which tracks the real-time CPU Utilization of the RDS instance and then set up a custom CloudWatch dashboard to view the metrics* is incorrect because although you can use Amazon CloudWatch Logs and CloudWatch dashboard to monitor the CPU Utilization of the database instance, using CloudWatch alone is still not enough to get the specific percentage of the CPU bandwidth and total memory consumed by each database processes. The data provided by CloudWatch is not as detailed as compared with the Enhanced Monitoring feature in RDS. Take note as well that you do not have direct access to the instances/servers of your RDS database instance, unlike with your EC2 instances where you can install a CloudWatch agent or a custom script to get CPU and memory utilization of your instance.

The option that says: *Check the CPU% and MEM% metrics which are readily available in the Amazon RDS console that shows the percentage of the CPU bandwidth and total memory consumed by each database process of your RDS instance* is incorrect because the CPU% and MEM% metrics are not readily available in the Amazon RDS console, which is contrary to what is being stated in this option.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_Monitoring.OS.html#USER_Monitoring.OS.CloudWatchLogs

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/MonitoringOverview.html#monitoring-cloud_watch

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

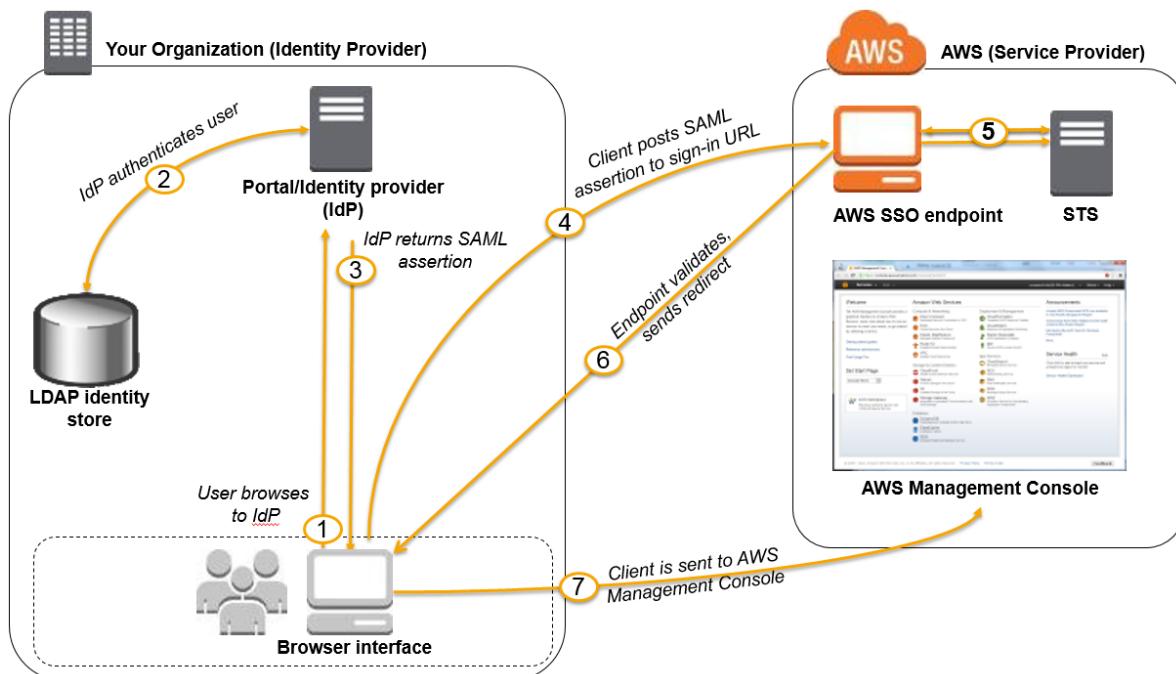
A pharmaceutical company has resources hosted on both their on-premises network and in AWS cloud. They want all of their Software Architects to access resources on both environments using their on-premises credentials, which is stored in Active Directory.

In this scenario, which of the following can be used to fulfill this requirement?

- Set up SAML 2.0-Based Federation by using a Web Identity Federation.
- **Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).**
- Use Amazon VPC
- Use IAM users

Correct

Since the company is using Microsoft Active Directory which implements Security Assertion Markup Language (SAML), you can set up a SAML-Based Federation for API Access to your AWS cloud. In this way, you can easily connect to AWS using the login credentials of your on-premises network.



AWS supports identity federation with SAML 2.0, an open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS APIs without you having to create an IAM user for everyone in your organization. By using SAML, you can simplify the process of configuring federation with AWS, because you can use the IdP's service instead of writing custom identity proxy code.

Before you can use SAML 2.0-based federation as described in the preceding scenario and diagram, you must configure your organization's IdP and your AWS account to trust each other. The general process for configuring this trust is described in the following steps. Inside your organization, you must have an IdP that supports SAML 2.0, like Microsoft Active Directory Federation Service (AD FS, part of Windows Server), Shibboleth, or another compatible SAML 2.0 provider.

Hence, the correct answer is: ***Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).***

Setting up SAML 2.0-Based Federation by using a Web Identity Federation is incorrect because this is primarily used to let users sign in via a well-known external identity provider (IdP), such as Login with Amazon, Facebook, Google. It does not utilize Active Directory.

Using IAM users is incorrect because the situation requires you to use the existing credentials stored in their Active Directory, and not user accounts that will be generated by IAM.

Using Amazon VPC is incorrect because this only lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. This has nothing to do with user authentication or Active Directory.

References:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

33. QUESTION

Category: CSAA – Design High-Performing Architectures

An organization needs to provision a new Amazon EC2 instance with a persistent block storage volume to migrate data from its on-premises network to AWS. The required maximum performance for the storage volume is 64,000 IOPS.

In this scenario, which of the following can be used to fulfill this requirement?

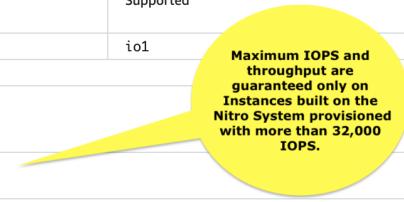
- Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.
- Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance.
- Launch any type of Amazon EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.
- Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O.

Correct

An **Amazon EBS volume** is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible.

The **AWS Nitro System** is the underlying platform for the latest generation of EC2 instances that enables AWS to innovate faster, further reduce the cost of the customers, and deliver added benefits like increased security and new instance types.

	Solid-state drives (SSD)		
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD io2	io1
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	<ul style="list-style-type: none"> Recommended for most workloads System boot volumes Virtual desktops Low-latency interactive apps Development and test environments 		<ul style="list-style-type: none"> Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume Large database workloads, such as: <ul style="list-style-type: none"> MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle
Amazon EBS Multi-attach	Not supported	Not Supported	Supported
API name	gp2	io2	io1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	
Dominant performance attribute	IOPS	IOPS	
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	
Max throughput per volume	250 MiB/s *	1,000 MiB/s †	
Max IOPS per instance ‡‡	160,000		
Max throughput per instance ‡‡	4,750 MB/s		



Amazon EBS is a persistent block storage volume. It can persist independently from the life of an instance. Since the scenario requires you to have an EBS volume with up to 64,000 IOPS, you have to launch a Nitro-based EC2 instance.

Hence, the correct answer in this scenario is: ***Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.***

The option that says: ***Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance*** is incorrect. Although an Instance Store is a block storage volume, it is not persistent and the data will be gone if the instance is restarted from the stopped state (*note that this is different from the OS-level reboot. In OS-level reboot, data still persists in the instance store*). **An instance store only provides temporary block-level storage for your instance. It means that the data in the instance store can be lost if the underlying disk drive fails, if the instance stops, and if the instance terminates.**

The option that says: ***Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O*** is incorrect. Although Amazon EFS can provide over 64,000 IOPS, this solution uses a file system and not a block storage volume which is what is asked in the scenario.

The option that says: ***Launch an EC2 instance and attach an io1 EBS volume with 64,000 IOPS*** is incorrect. In order to achieve the 64,000 IOPS for a provisioned IOPS SSD, you must provision a Nitro-based EC2 instance. The maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System provisioned with more than 32,000 IOPS. Other instances guarantee up to 32,000 IOPS only.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#EBSVolumeTypes_piops

<https://aws.amazon.com/s3/storage-classes/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

Amazon S3 vs EFS vs EBS Cheat Sheet:

34. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

An application is hosted in an AWS Fargate cluster that runs a batch job whenever an object is loaded on an Amazon S3 bucket. The minimum number of ECS Tasks is initially set to 1 to save on costs, and it will only increase the task count based on the new objects uploaded on the S3 bucket. Once processing is done, the bucket becomes empty and the ECS Task count should be back to 1.

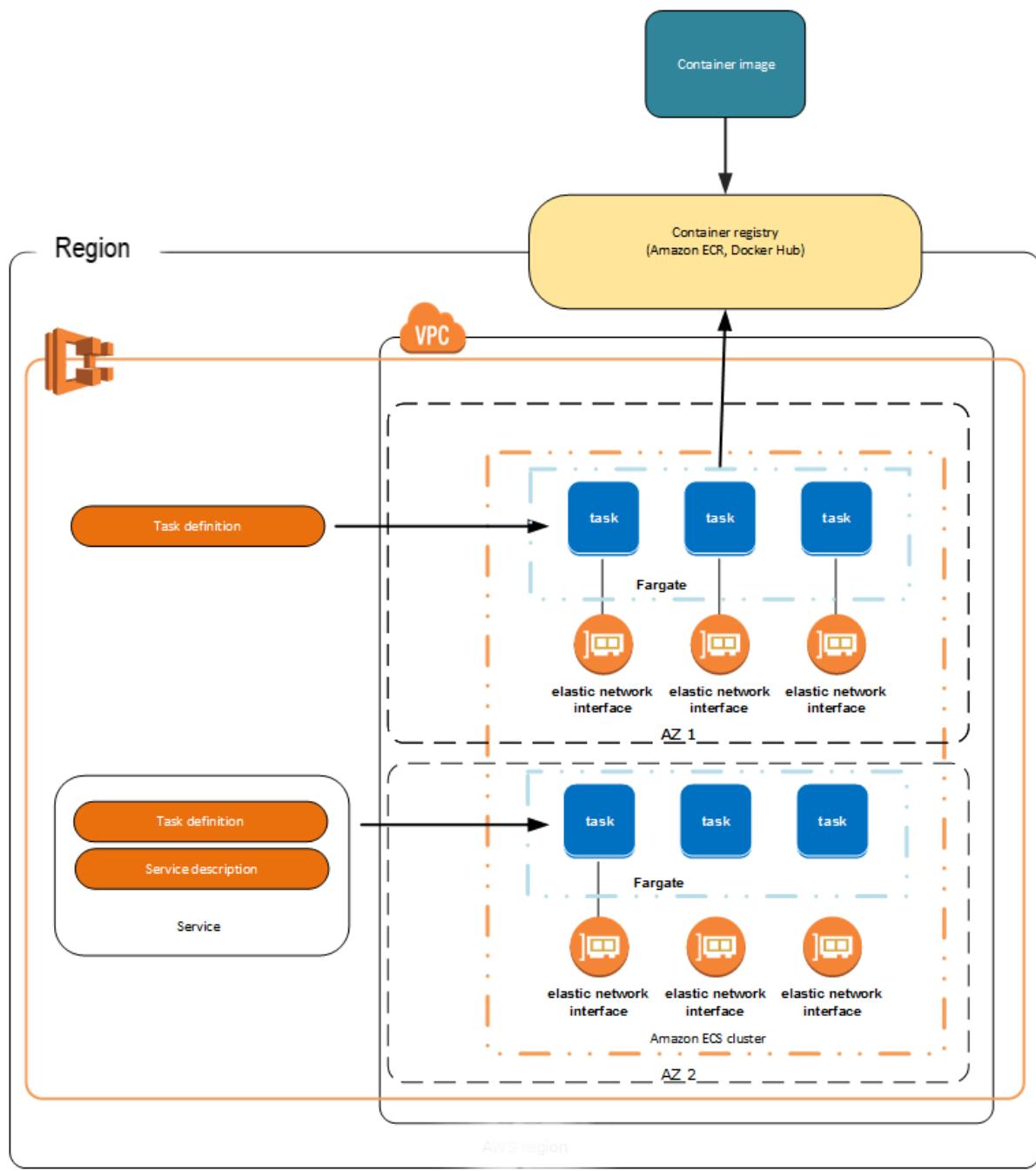
Which is the most suitable option to implement with the LEAST amount of effort?

- Set up an alarm in CloudWatch to monitor CloudTrail since the S3 object-level operations are recorded on CloudTrail. Create two Lambda functions for increasing/decreasing the ECS task count. Set these as respective targets for the CloudWatch Alarm depending on the S3 event.
- Set up a CloudWatch Event rule to detect S3 object PUT operations and set the target to a Lambda function that will run Amazon ECS API command to increase the number of tasks on ECS. Create another rule to detect S3 DELETE operations and run the Lambda function to reduce the number of ECS tasks.
- Set up a CloudWatch Event rule to detect S3 object PUT operations and set the target to the ECS cluster with the increased number of tasks. Create another rule to detect S3 DELETE operations and set the target to the ECS Cluster with 1 as the Task count.
- Set up an alarm in CloudWatch to monitor CloudTrail since this S3 object-level operations are recorded on CloudTrail. Set two alarm actions to update ECS task count to scale-out/scale-in depending on the S3 event.

Incorrect

You can use **CloudWatch Events** to run Amazon ECS tasks when certain AWS events occur. You can set up a CloudWatch Events rule that runs an Amazon ECS task whenever a file is uploaded to a certain Amazon S3 bucket using the Amazon S3 PUT operation. You can also declare a reduced number of ECS tasks whenever a file is deleted on the S3 bucket using the DELETE operation.

First, you must create a CloudWatch Events rule for the S3 service that will watch for object-level operations – PUT and DELETE objects. For object-level operations, it is required to create a CloudTrail trail first. On the Targets section, select the “ECS task” and input the needed values such as the cluster name, task definition and the task count. You need two rules – one for the scale-up and another for the scale-down of the ECS task count.



Hence, the correct answer is: ***Set up a CloudWatch Event rule to detect S3 object PUT operations and set the target to the ECS cluster with the increased number of tasks. Create another rule to detect S3 DELETE operations and set the target to the ECS Cluster with 1 as the Task count.***

The option that says: ***Set up a CloudWatch Event rule to detect S3 object PUT operations and set the target to a Lambda function that will run Amazon ECS API command to increase the number of tasks on ECS. Create another rule to detect S3 DELETE operations and run the Lambda function to reduce the number of ECS tasks*** is incorrect. Although this solution meets the requirement, creating your own Lambda function for this scenario is not really necessary. It is much simpler to control ECS task directly as target for the CloudWatch Event rule. Take note that the scenario asks for a solution that is the easiest to implement.

The option that says: ***Set up an alarm in CloudWatch to monitor CloudTrail since the S3 object-level operations are recorded on CloudTrail. Create two Lambda functions for increasing/decreasing the ECS task count. Set these as respective targets for the CloudWatch Alarm depending on the S3 event*** is incorrect because using CloudTrail, CloudWatch Alarm, and two Lambda functions creates an unnecessary complexity to what you want to achieve. CloudWatch Events can directly target an ECS task on the Targets section when you create a new rule.

The option that says: ***Set up an alarm in CloudWatch to monitor CloudTrail since this S3 object-level operations are recorded on CloudTrail. Set two alarm actions to update ECS task count to scale-out/scale-in depending on the S3 event*** is incorrect because you can't directly set CloudWatch Alarms to update the ECS task count. You have to use CloudWatch Events instead.

References:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/CloudWatch-Events-tutorial-ECS.html>

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/Create-CloudWatch-Events-Rule.html>

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

Amazon CloudWatch Overview:

<https://youtu.be/q0DmxfyGkeU>

35. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An online medical system hosted in AWS stores sensitive Personally Identifiable Information (PII) of the users in an Amazon S3 bucket. Both the master keys and the unencrypted data should never be sent to AWS to comply with the strict compliance and regulatory requirements of the company.

Which S3 encryption technique should the Architect use?

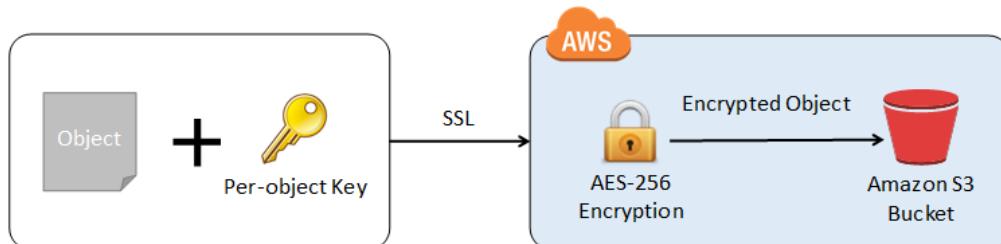
- Use S3 server-side encryption with a KMS managed key.
- Use S3 client-side encryption with a client-side master key.
- Use S3 server-side encryption with customer provided key.
- Use S3 client-side encryption with a KMS-managed customer master key.

Correct

Client-side encryption is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options:

- Use an AWS KMS-managed customer master key.
- Use a client-side master key.

When using an AWS KMS-managed customer master key to enable client-side data encryption, you provide an AWS KMS customer master key ID (CMK ID) to AWS. On the other hand, when you use client-side master key for client-side data encryption, **your client-side master keys and your unencrypted data are never sent to AWS**. It's important that you safely manage your encryption keys because if you lose them, you can't decrypt your data.



This is how client-side encryption using client-side master key works:

When uploading an object – You provide a client-side master key to the Amazon S3 encryption client. The client uses the master key only to encrypt the data encryption key that it generates randomly. The process works like this:

- \1. The Amazon S3 encryption client generates a one-time-use symmetric key (also known as a data encryption key or data key) locally. It uses the data key to encrypt the data of a single Amazon S3 object. The client generates a separate data key for each object.
- \2. The client encrypts the data encryption key using the master key that you provide. The client uploads the encrypted data key and its material description as part of the object metadata. The client uses the material description to determine which client-side master key to use for decryption.
- \3. The client uploads the encrypted data to Amazon S3 and saves the encrypted data key as object metadata (`x-amz-meta-x-amz-key`) in Amazon S3.

When downloading an object – The client downloads the encrypted object from Amazon S3. Using the material description from the object's metadata, the client determines which master key to use to decrypt the data key. The client uses that master key to decrypt the data key and then uses the data key to decrypt the object.

Hence, the correct answer is to ***use S3 client-side encryption with a client-side master key***.

Using S3 client-side encryption with a KMS-managed customer master key is incorrect because in client-side encryption with a KMS-managed customer master key, you provide an AWS KMS customer master key ID (CMK ID) to AWS. The scenario clearly indicates that both the master keys and the unencrypted data should never be sent to AWS.

Using S3 server-side encryption with a KMS managed key is incorrect because the scenario mentioned that the unencrypted data should never be sent to AWS, which means that you have to use client-side encryption in order to encrypt the data first before sending to AWS. In this way, you can ensure that there is no unencrypted data being uploaded to AWS. In addition, the master key used by Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS) is uploaded and managed by AWS, which directly violates the requirement of not uploading the master key.

Using S3 server-side encryption with customer provided key is incorrect because just as mentioned above, you have to use client-side encryption in this scenario instead of server-side encryption. For the S3 server-side encryption with customer-provided key (SSE-C), you actually provide the encryption key as part of your request to upload the object to S3. Using this key, Amazon S3 manages both the encryption (as it writes to disks) and decryption (when you access your objects).

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

36. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company is designing a banking portal that uses Amazon ElastiCache for Redis as its distributed session management component. Since the other Cloud Engineers in your department have access to your ElastiCache cluster, you have to secure the session data in the portal by requiring them to enter a password before they are granted permission to execute Redis commands.

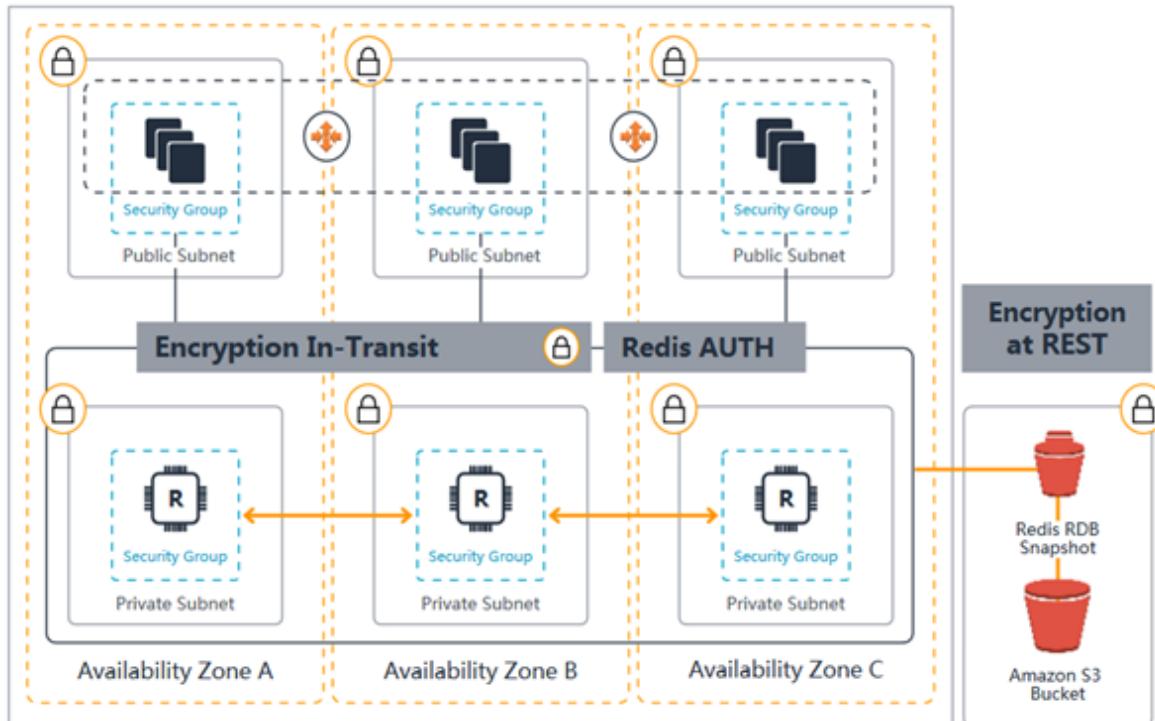
As the Solutions Architect, which of the following should you do to meet the above requirement?

- Set up an IAM Policy and MFA which requires the Cloud Engineers to enter their IAM credentials and token before they can access the ElastiCache cluster.
- Set up a Redis replication group and enable the `AtRestEncryptionEnabled` parameter.
- Enable the in-transit encryption for Redis replication groups.
- **Authenticate the users using Redis AUTH by creating a new Redis Cluster with both the `--transit-encryption-enabled` and `--auth-token` parameters enabled.**

Incorrect

Using `*Redis AUTH**` command can improve data security by requiring the user to enter a password before they are granted permission to execute Redis commands on a password-protected Redis server. Hence, the correct answer is: ****Authenticate the users using Redis AUTH by creating a new Redis Cluster with both the `--transit-encryption-enabled` and `--auth-token` parameters enabled.****

To require that users enter a password on a password-protected Redis server, include the parameter `*--auth-token**` with the correct password when you create your replication group or cluster and on all subsequent commands to the replication group or cluster.



Setting up an IAM Policy and MFA which requires the Cloud Engineers to enter their IAM credentials and token before they can access the ElastiCache cluster is incorrect because this is not possible in IAM. You have to use the Redis AUTH option instead.

Setting up a Redis replication group and enabling the* `AtRestEncryptionEnabled` **parameter* is incorrect because the Redis At-Rest Encryption feature only secures the data inside the in-memory data store. You have to use Redis AUTH option instead.

Enabling the in-transit encryption for Redis replication groups is incorrect. Although in-transit encryption is part of the solution, it is missing the most important thing which is the Redis AUTH option.

References:

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html>

<https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/encryption.html>

Check out this Amazon Elasticache Cheat Sheet:

<https://tutorialsdojo.com/amazon-elasticache/>

Redis (cluster mode enabled vs disabled) vs Memcached:

<https://tutorialsdojo.com/redis-cluster-mode-enabled-vs-disabled-vs-memcached/>

37. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A software development company is using serverless computing with AWS Lambda to build and run applications without having to set up or manage servers. They have a Lambda function that connects to a MongoDB Atlas, which is a popular Database as a Service (DBaaS) platform and also uses a third party API to fetch certain data for their application. One of the developers was instructed to create the environment variables for the MongoDB database hostname, username, and password as well as the API credentials that will be used by the Lambda function for DEV, SIT, UAT, and PROD environments.

Considering that the Lambda function is storing sensitive database and API credentials, how can this information be secured to prevent other developers in the team, or anyone, from seeing these credentials in plain text? Select the best option that provides maximum security.

- AWS Lambda does not provide encryption for the environment variables. Deploy your code to an EC2 instance instead.
- **Create a new KMS key and use it to enable encryption helpers that leverage on AWS Key Management Service to store and encrypt the sensitive information.**
- There is no need to do anything because, by default, AWS Lambda already encrypts the environment variables using the AWS Key Management Service.
- Enable SSL encryption that leverages on AWS CloudHSM to store and encrypt the sensitive information.

Correct

When you create or update Lambda functions that use environment variables, AWS Lambda encrypts them using the AWS Key Management Service. When your Lambda function is invoked, those values are decrypted and made available to the Lambda code.

The first time you create or update Lambda functions that use environment variables in a region, a default service key is created for you automatically within AWS KMS. This key is used to encrypt environment variables. However, if you wish to use encryption helpers and use KMS to encrypt environment variables after your Lambda function is created, you must create your own AWS KMS key and choose it instead of the default key. The default key will give errors when chosen. Creating your own key gives you more flexibility, including the ability to create, rotate, disable, and define access controls, and to audit the encryption keys used to protect your data.

The screenshot shows the AWS Lambda Environment Variables configuration page. In the 'Encryption configuration' section, there is a checked checkbox for 'Enable helpers for encryption in transit'. Below it, a dropdown menu for 'AWS KMS key to encrypt in transit' is open, showing a search bar with 'arn:aws:kms:us-east-1:8420' and a dropdown item '7:key/2defc6c2-ab8a-499f-87de-'. A red error message at the bottom of the dropdown says 'AWS KMS call failed for reason: User: arn:aws:iam::84205 7:user/koko is not authorized to perform: kms:Encrypt on resource: arn:aws:kms:us-east-1:84205'. At the bottom of the page, there is another dropdown for 'AWS KMS key to encrypt at rest' with options '(default) aws/lambda' and 'Use a customer master key'.

The option that says: ***There is no need to do anything because, by default, AWS Lambda already encrypts the environment variables using the AWS Key Management Service*** is incorrect. Although Lambda encrypts the environment variables in your function by default, the sensitive information would still be visible to other users who have access to the Lambda console. This is because Lambda uses a default KMS key to encrypt the variables, which is usually accessible by other users. The best option in this scenario is to use encryption helpers to secure your environment variables.

The option that says: ***Enable SSL encryption that leverages on AWS CloudHSM to store and encrypt the sensitive information*** is also incorrect since enabling SSL would encrypt data only when in-transit. Your other teams would still be able to view the plaintext at-rest. Use AWS KMS instead.

The option that says: ***AWS Lambda does not provide encryption for the environment variables. Deploy your code to an EC2 instance instead*** is incorrect since, as mentioned, Lambda does provide encryption functionality of environment variables.

References:

https://docs.aws.amazon.com/lambda/latest/dg/env_variables.html#env_encrypt

https://docs.aws.amazon.com/lambda/latest/dg/tutorial-env_console.html

Check out this AWS Lambda Cheat Sheet:

<https://tutorialsdojo.com/aws-lambda/>

AWS Lambda Overview – Serverless Computing in AWS:

<https://youtu.be/bPVX1zHwAnY>

38. QUESTION

Category: CSAA – Design Resilient Architectures

A Solutions Architect is working for a company which has multiple VPCs in various AWS regions. The Architect is assigned to set up a logging system which will track all of the changes made to their AWS resources in all regions, including the configurations made in IAM, CloudFront, AWS WAF, and Route 53. In order to pass the compliance requirements, the solution must ensure the security, integrity, and durability of the log data. It should also provide an event history of all API calls made in AWS Management Console and AWS CLI.

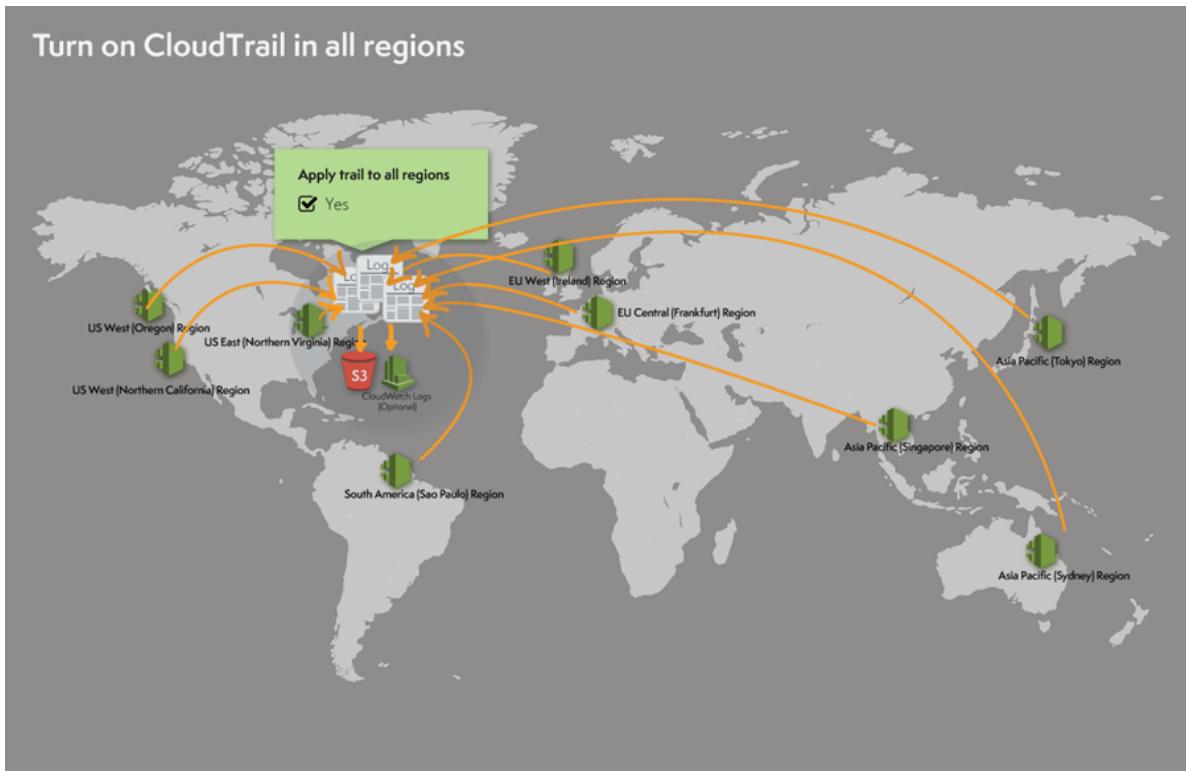
Which of the following solutions is the best fit for this scenario?

- Set up a new CloudTrail trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `--no-include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- Set up a new CloudWatch trail in a new S3 bucket using the CloudTrail console and also pass the `--is-multi-region-trail` parameter then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- Set up a new CloudWatch trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `-include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.
- Set up a new CloudTrail trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `--include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies.

Incorrect

An **event** in CloudTrail is the record of an activity in an AWS account. This activity can be an action taken by a user, role, or service that is monitorable by CloudTrail. CloudTrail events provide a history of both API and non-API account activity made through the AWS Management Console, AWS SDKs, command-line tools, and other AWS services. There are two types of events that can be logged in CloudTrail: management events and data events. By default, trails log management events, but not data events.

Turn on CloudTrail in all regions



A trail can be applied to all regions or a single region. As a best practice, create a trail that applies to all regions in the AWS partition in which you are working. This is the default setting when you create a trail in the CloudTrail console.

For most services, events are recorded in the region where the action occurred. For global services such as AWS Identity and Access Management (IAM), AWS STS, Amazon CloudFront, and Route 53, events are delivered to any trail that includes global services, and are logged as occurring in US East (N. Virginia) Region.

In this scenario, the company requires a secure and durable logging solution that will track all of the activities of all AWS resources in all regions. CloudTrail can be used for this case with multi-region trail enabled, however, it will only cover the activities of the regional services (EC2, S3, RDS etc.) and not for global services such as IAM, CloudFront, AWS WAF, and Route 53. In order to satisfy the requirement, you have to add the `--include-global-service-events` parameter in your AWS CLI command.

The option that says: ***Set up a new CloudTrail trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `--include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies*** is correct because it provides security, integrity, and durability to your log data and in addition, it has the `-include-global-service-events` parameter enabled which will also include activity from global services such as IAM, Route 53, AWS WAF, and CloudFront.

The option that says: ***Set up a new CloudWatch trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `--include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies*** is incorrect because you need to use CloudTrail instead of CloudWatch.

The option that says: ***Set up a new CloudWatch trail in a new S3 bucket using the CloudTrail console and also pass the `--is-multi-region-trail` parameter then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies*** is incorrect because you need to use CloudTrail instead of CloudWatch. In addition, the `-include-global-service-events` parameter is also missing in this setup.

The option that says: ***Set up a new CloudTrail trail in a new S3 bucket using the AWS CLI and also pass both the `--is-multi-region-trail` and `--no-include-global-service-events` parameters then encrypt log files using KMS encryption. Apply Multi Factor Authentication (MFA) Delete on the S3 bucket and ensure that only authorized users can access the logs by configuring the bucket policies*** is incorrect because the `--is-multi-region-trail` is not enough as you also need to add the `--include-global-service-events` parameter and not `--no-include-global-service-events`.

References:

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html#cloudtrail-concepts-global-service-events>

<http://docs.aws.amazon.com/IAM/latest/UserGuide/cloudtrail-integration.html>

<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-create-and-update-a-trail-by-using-the-aws-cli.html>

Check out this AWS CloudTrail Cheat Sheet:

<https://tutorialsdojo.com/aws-cloudtrail/>

39. QUESTION

Category: CSAA – Design High-Performing Architectures

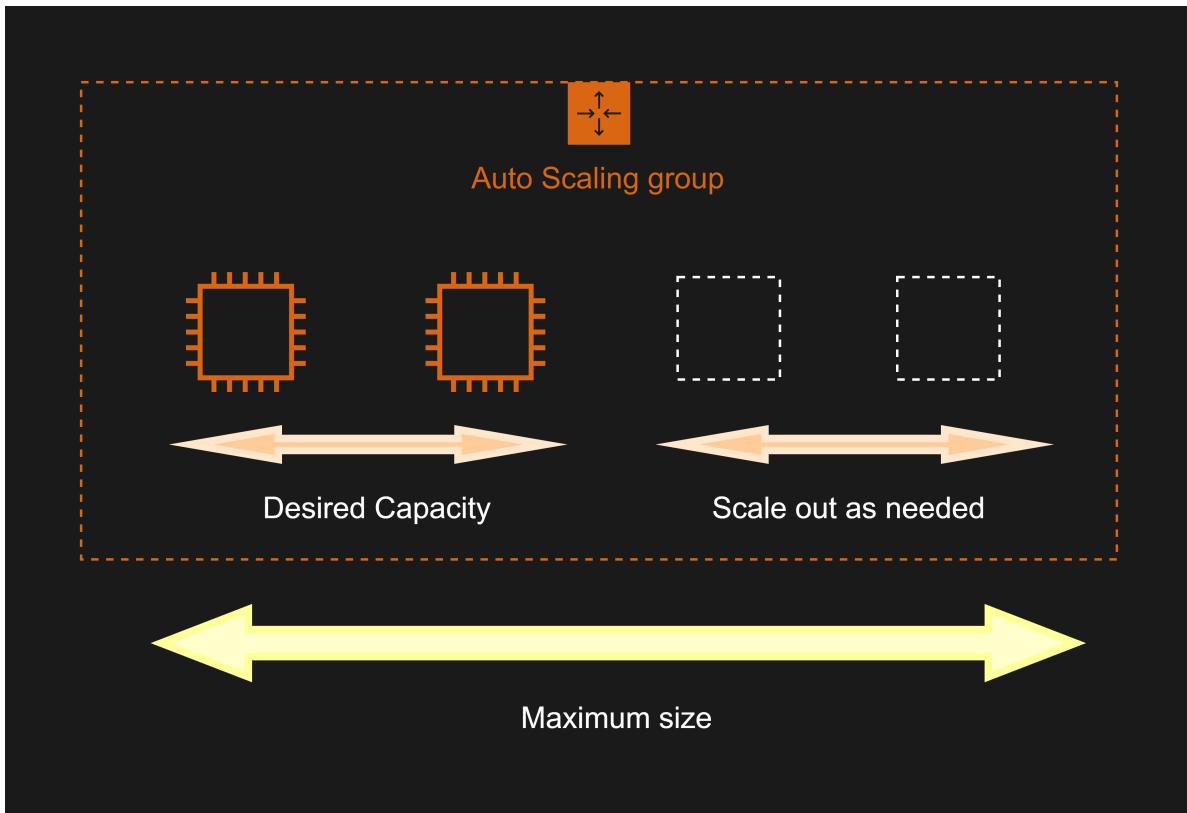
A tech company has a CRM application hosted on an Auto Scaling group of On-Demand EC2 instances. The application is extensively used during office hours from 9 in the morning till 5 in the afternoon. Their users are complaining that the performance of the application is slow during the start of the day but then works normally after a couple of hours.

Which of the following can be done to ensure that the application works properly at the beginning of the day?

- Set up an Application Load Balancer (ALB) to your architecture to ensure that the traffic is properly distributed on the instances.
- Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization.
- Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the Memory utilization.
- **Configure a Scheduled scaling policy for the Auto Scaling group to launch new instances before the start of the day.**

Correct

Scaling based on a schedule allows you to scale your application in response to predictable load changes. For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling activities based on the predictable traffic patterns of your web application.



To configure your Auto Scaling group to scale based on a schedule, you create a scheduled action. The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times. To create a scheduled scaling action, you specify the start time when the scaling action should take effect, and the new minimum, maximum, and desired sizes for the scaling action. At the specified time, Amazon EC2 Auto Scaling updates the group with the values for minimum, maximum, and desired size specified by the scaling action. You can create scheduled actions for scaling one time only or for scaling on a recurring schedule.

Hence, ***configuring a Scheduled scaling policy for the Auto Scaling group to launch new instances before the start of the day*** is the correct answer. You need to configure a Scheduled scaling policy. This will ensure that the instances are already scaled up and ready before the start of the day since this is when the application is used the most.

Configuring a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization and ***configuring a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the Memory utilization*** are both incorrect because although these are valid solutions, it is still better to configure a Scheduled scaling policy as you already know the exact peak hours of your application. By the time either the CPU or Memory hits a peak, the application already has performance issues, so you need to ensure the scaling is done beforehand using a Scheduled scaling policy.

Setting up an Application Load Balancer (ALB) to your architecture to ensure that the traffic is properly distributed on the instances is incorrect. Although the Application load balancer can also balance the traffic, it cannot increase the instances based on demand.

Reference:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/schedule_time.html

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

40. QUESTION

Category: CSAA – Design Resilient Architectures

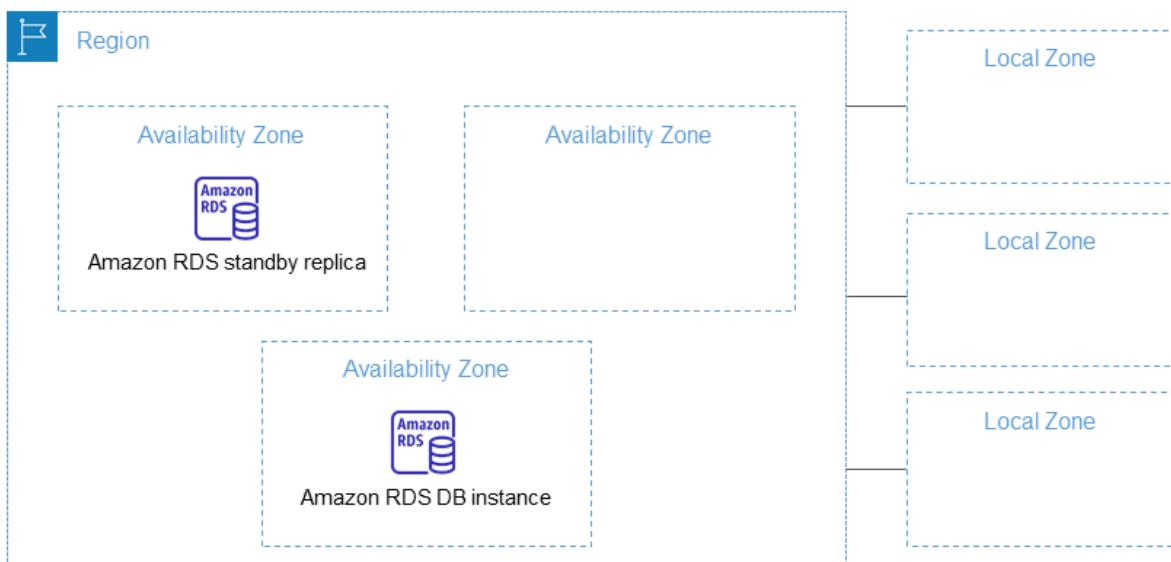
A Forex trading platform, which frequently processes and stores global financial data every minute, is hosted in your on-premises data center and uses an Oracle database. Due to a recent cooling problem in their data center, the company urgently needs to migrate their infrastructure to AWS to improve the performance of their applications. As the Solutions Architect, you are responsible in ensuring that the database is properly migrated and should remain available in case of database server failure in the future.

Which of the following is the most suitable solution to meet the requirement?

- Create an Oracle database in RDS with Multi-AZ deployments.
- Launch an Oracle Real Application Clusters (RAC) in RDS.
- Launch an Oracle database instance in RDS with Recovery Manager (RMAN) enabled.
- Convert the database schema using the AWS Schema Conversion Tool and AWS Database Migration Service. Migrate the Oracle database to a non-cluster Amazon Aurora with a single instance.

Correct

Amazon RDS Multi-AZ deployments provide enhanced availability and durability for Database (DB) Instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB Instance, Amazon RDS automatically creates a primary DB Instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ). Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.



In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora), so that you can resume database operations as soon as the failover is complete. Since the endpoint for your DB Instance remains the same after a failover, your application can resume database operation without the need for manual administrative intervention.

In this scenario, the best RDS configuration to use is an Oracle database in RDS with Multi-AZ deployments to ensure high availability even if the primary database instance goes down. Hence, ***creating an Oracle database in RDS with Multi-AZ deployments*** is the correct answer.

Launching an Oracle database instance in RDS with Recovery Manager (RMAN) enabled and ***launching an Oracle Real Application Clusters (RAC) in RDS*** are incorrect because Oracle RMAN and RAC are not supported in RDS.

The option that says: ***Convert the database schema using the AWS Schema Conversion Tool and AWS Database Migration Service. Migrate the Oracle database to a non-cluster Amazon Aurora with a single instance*** is incorrect because although this solution is feasible, it takes time to migrate your Oracle database to Aurora, which is not acceptable. Based on this option, the Aurora database is only using a

single instance with no Read Replica and is not configured as an Amazon Aurora DB cluster, which could have improved the availability of the database.

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

41. QUESTION

Category: CSAA – Design High-Performing Architectures

A content management system (CMS) is hosted on a fleet of auto-scaled, On-Demand EC2 instances that use Amazon Aurora as its database. Currently, the system stores the file documents that the users upload in one of the attached EBS Volumes. Your manager noticed that the system performance is quite slow and he has instructed you to improve the architecture of the system.

In this scenario, what will you do to implement a scalable, high-available POSIX-compliant shared file system?

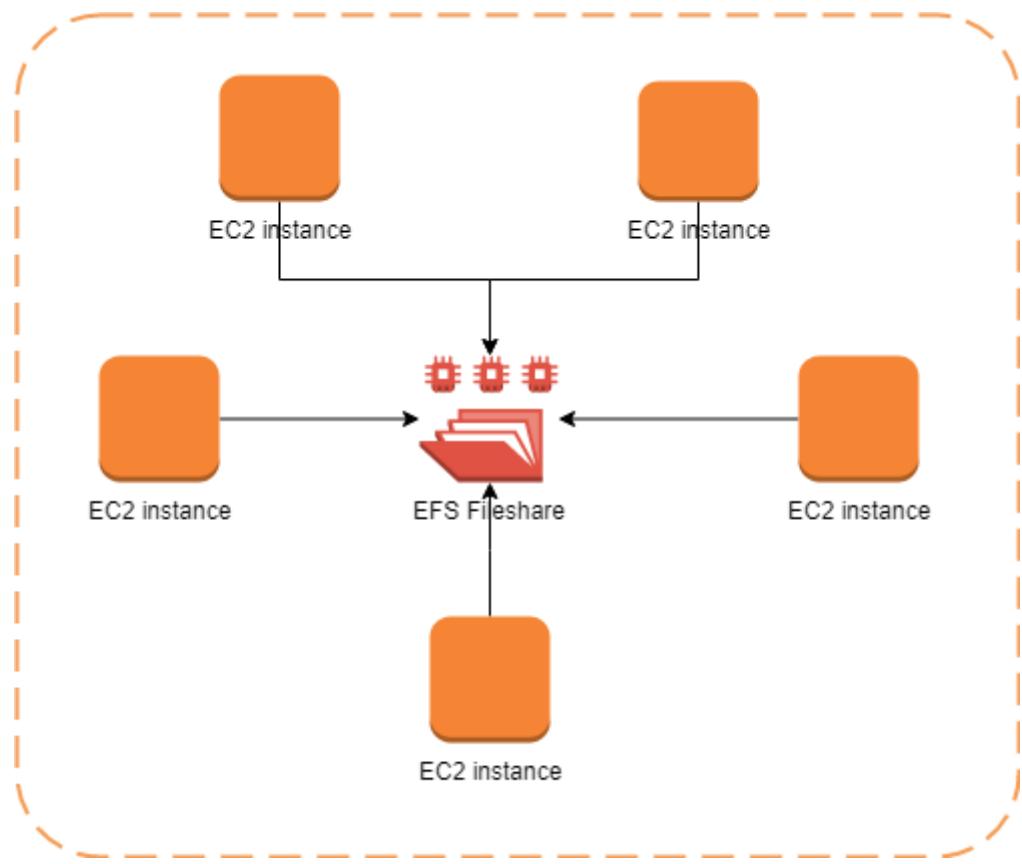
- Create an S3 bucket and use this as the storage for the CMS
- **Use EFS**
- Use ElastiCache
- Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes

Correct

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. When mounted on Amazon EC2 instances, an Amazon EFS file system provides a standard file system interface and file system access semantics, allowing you to seamlessly integrate Amazon EFS with your existing applications and tools. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, allowing Amazon EFS to provide a common data source for workloads and applications running on more than one Amazon EC2 instance.

This particular scenario tests your understanding of EBS, EFS, and S3. In this scenario, there is a fleet of On-Demand EC2 instances that store file documents from the users to one of the attached EBS Volumes. The system performance is quite slow because the architecture doesn't provide the EC2 instances parallel shared access to the file documents.

Although an EBS Volume can be attached to multiple EC2 instances, you can only do so on instances within an availability zone. What we need is high-available storage that can span multiple availability zones. Take note as well that the type of storage needed here is "file storage" which means that ***S3*** is not the best service to use because it is mainly used for "object storage", and S3 does not provide the notion of "folders" too. This is why ***using EFS*** is the correct answer.



Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes is incorrect because an EBS volume is a storage area network (SAN) storage and not a POSIX-compliant shared file system. You have to use EFS instead.

Using ElastiCache is incorrect because this is an in-memory data store that improves the performance of your applications, which is not what you need since it is not a file storage.

Reference:

<https://aws.amazon.com/efs/>

Check out this Amazon EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-efs/>

Check out this Amazon S3 vs EBS vs EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/>

42. QUESTION

Category: CSAA – Design Resilient Architectures

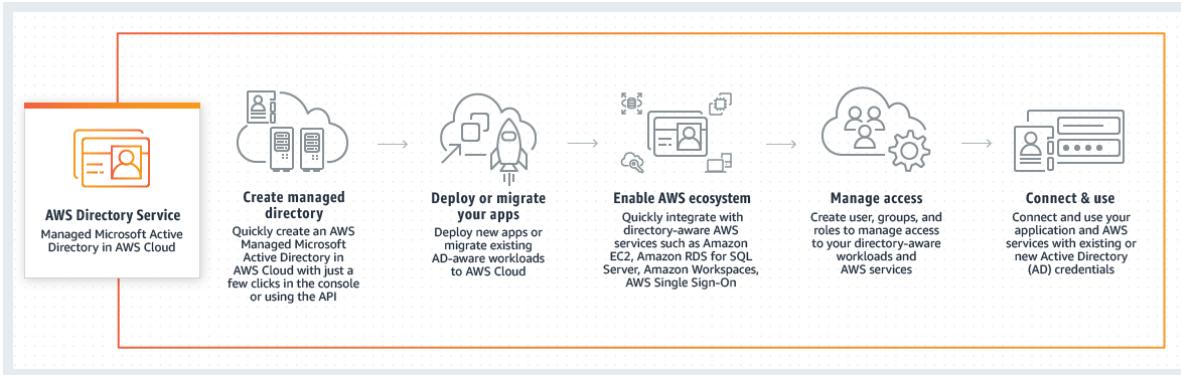
A telecommunications company is planning to give AWS Console access to developers. Company policy mandates the use of identity federation and role-based access control. Currently, the roles are already assigned using groups in the corporate Active Directory.

In this scenario, what combination of the following services can provide developers access to the AWS console? (Select TWO.)

- AWS Directory Service AD Connector
- AWS Directory Service Simple AD
- IAM Groups
- Lambda
- IAM Roles

Incorrect

Considering that the company is using a corporate Active Directory, it is best to use ***AWS Directory Service AD Connector*** for easier integration. In addition, since the roles are already assigned using groups in the corporate Active Directory, it would be better to also use ***IAM Roles***. Take note that you can assign an IAM Role to the users or groups from your Active Directory once it is integrated with your VPC via the AWS Directory Service AD Connector.



AWS Directory Service provides multiple ways to use Amazon Cloud Directory and Microsoft Active Directory (AD) with other AWS services. Directories store information about users, groups, and devices, and administrators use them to manage access to information and resources. AWS Directory Service provides multiple directory choices for customers who want to use existing Microsoft AD or Lightweight Directory Access Protocol (LDAP)-aware applications in the cloud. It also offers those same choices to developers who need a directory to manage users, groups, devices, and access.

AWS Directory Service Simple AD is incorrect because this just provides a **subset** of the features offered by AWS Managed Microsoft AD, including the ability to manage user accounts and group memberships, create and apply group policies, securely connect to Amazon EC2 instances, and provide Kerberos-based single sign-on (SSO). In this scenario, the more suitable component to use is the AD Connector since it is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory.

IAM Groups is incorrect because this is just a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. In this scenario, the more suitable one to use is IAM Roles in order for permissions to create AWS Directory Service resources.

Lambda is incorrect because this is primarily used for serverless computing.

Reference:

<https://aws.amazon.com/blogs/security/how-to-connect-your-on-premises-active-directory-to-aws-using-a-d-connector/>

Check out these AWS IAM and Directory Service Cheat Sheets:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

<https://tutorialsdojo.com/aws-directory-service/>

Here is a video tutorial on AWS Directory Service:

<https://youtu.be/4XeqotTYBtY>

43. QUESTION

Category: CSAA – Design Resilient Architectures

A company has a cloud architecture that is composed of Linux and Windows EC2 instances that process high volumes of financial data 24 hours a day, 7 days a week. To ensure high availability of the systems, the Solutions Architect needs to create a solution that allows them to monitor the memory and disk utilization metrics of all the instances.

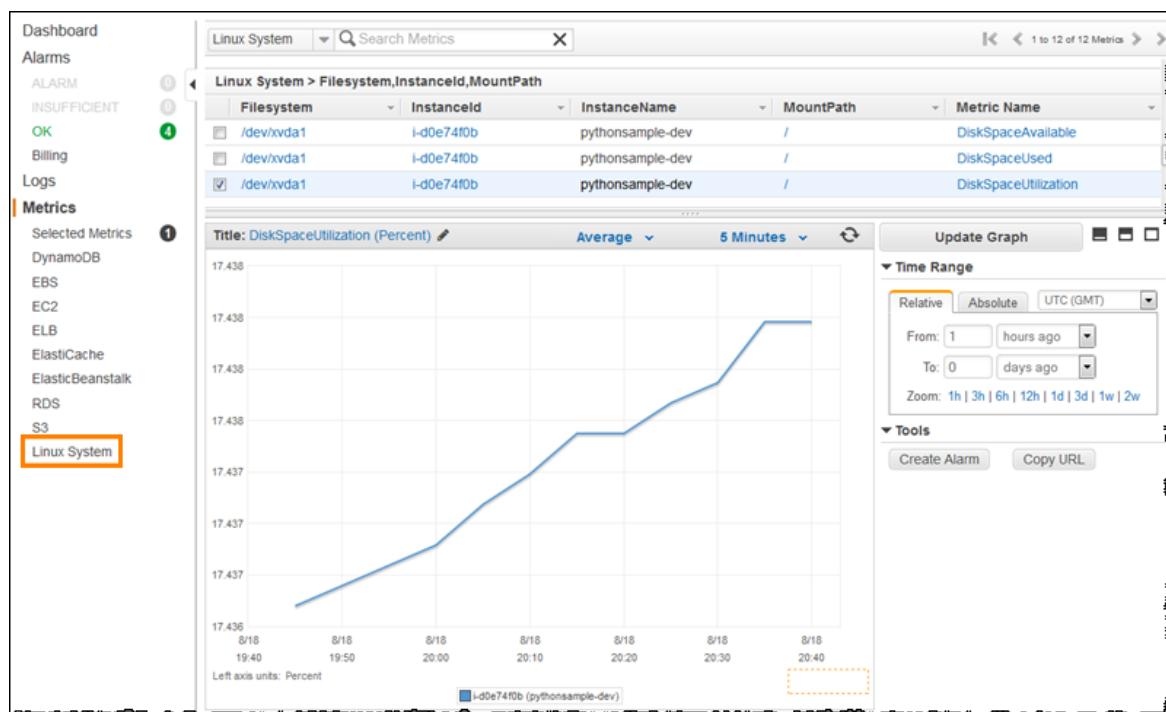
Which of the following is the most suitable monitoring solution to implement?

- Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances.
- **Install the CloudWatch agent to all the EC2 instances that gather the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.**
- Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard.
- Use Amazon Inspector and install the Inspector agent to all EC2 instances.

Incorrect

Amazon CloudWatch has available Amazon EC2 Metrics for you to use for monitoring CPU utilization, Network utilization, Disk performance, and Disk Reads/Writes. In case you need to monitor the below items, you need to prepare a custom metric using a Perl or other shell script, as there are no ready to use metrics for:

1. Memory utilization
2. Disk swap utilization
3. Disk space utilization
4. Page file utilization
5. Log collection



Take note that there is a multi-platform CloudWatch agent which can be installed on both Linux and Windows-based instances. You can use a single agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers. This agent supports both Windows Server and Linux and enables you to select the metrics to be collected, including sub-resource metrics such as per-CPU core. It is recommended that you use the new agent instead of the older monitoring scripts to collect metrics and logs.

Hence, the correct answer is: ***Install the CloudWatch agent to all the EC2 instances that gathers the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.***

The option that says: ***Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances*** is incorrect because, by default, CloudWatch does not automatically provide memory and disk utilization metrics of your instances. You have to set up custom CloudWatch metrics to monitor the memory, disk swap, disk space, and page file utilization of your instances.

The option that says: ***Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard*** is incorrect because Enhanced Monitoring is a feature of Amazon RDS. By default, Enhanced Monitoring metrics are stored for 30 days in the CloudWatch Logs.

The option that says: ***Use Amazon Inspector and install the Inspector agent to all EC2 instances*** is incorrect because Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. It does not provide a custom metric to track the memory and disk utilization of each and every EC2 instance in your VPC.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html#using_put_script

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

CloudWatch Agent vs SSM Agent vs Custom Daemon Scripts:

<https://tutorialsdojo.com/cloudwatch-agent-vs-ssm-agent-vs-custom-daemon-scripts/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

44. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A company hosted a web application in an Auto Scaling group of EC2 instances. The IT manager is concerned about the over-provisioning of the resources that can cause higher operating costs. A Solutions Architect has been instructed to create a cost-effective solution without affecting the performance of the application.

Which dynamic scaling policy should be used to satisfy this requirement?

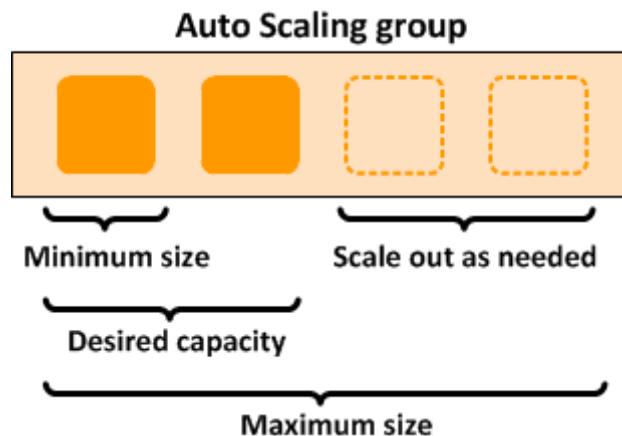
- Use suspend and resume scaling.
- Use scheduled scaling.
- **Use target tracking scaling.**
- Use simple scaling.

Correct

An **Auto Scaling group** contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and management. An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core

functionality of the Amazon EC2 Auto Scaling service. The size of an Auto Scaling group depends on the number of instances that you set as the desired capacity. You can adjust its size to meet demand, either manually or by using automatic scaling.

Step scaling policies and simple scaling policies are two of the dynamic scaling options available for you to use. Both require you to create CloudWatch alarms for the scaling policies. Both require you to specify the high and low thresholds for the alarms. Both require you to define whether to add or remove instances, and how many, or set the group to an exact size. The main difference between the policy types is the step adjustments that you get with step scaling policies. When step adjustments are applied, and they increase or decrease the current capacity of your Auto Scaling group, the adjustments vary based on the size of the alarm breach.



The primary issue with simple scaling is that after a scaling activity is started, the policy must wait for the scaling activity or health check replacement to complete and the cooldown period to expire before responding to additional alarms. Cooldown periods help to prevent the initiation of additional scaling activities before the effects of previous activities are visible.

With a target tracking scaling policy, you can increase or decrease the current capacity of the group based on a target value for a specific metric. This policy will help resolve the over-provisioning of your resources. The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value. In addition to keeping the metric close to the target value, a target tracking scaling policy also adjusts to changes in the metric due to a changing load pattern.

Hence, the correct answer is: ***Use target tracking scaling.***

The option that says: ***Use simple scaling*** is incorrect because you need to wait for the cooldown period to complete before initiating additional scaling activities. Target tracking or step scaling policies can trigger a scaling activity immediately without waiting for the cooldown period to expire.

The option that says: ***Use scheduled scaling*** is incorrect because this policy is mainly used for predictable traffic patterns. You need to use the target tracking scaling policy to optimize the cost of your infrastructure without affecting the performance.

The option that says: ***Use suspend and resume scaling*** is incorrect because this type is used to temporarily pause scaling activities triggered by your scaling policies and scheduled actions.

References:

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-target-tracking.html>

<https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

45. QUESTION

Category: CSAA – Design High-Performing Architectures

The company that you are working for has a highly available architecture consisting of an elastic load balancer and several EC2 instances configured with auto-scaling in three Availability Zones. You want to monitor your EC2 instances based on a particular metric, which is not readily available in CloudWatch.

Which of the following is a custom metric in CloudWatch which you have to manually set up?

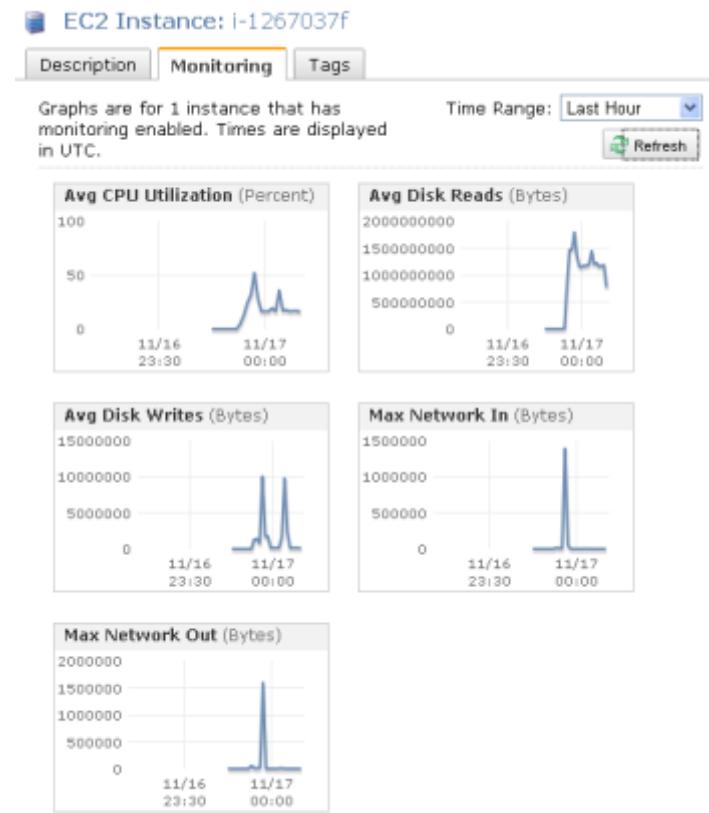
- Network packets out of an EC2 instance
- Memory Utilization of an EC2 instance
- CPU Utilization of an EC2 instance
- Disk Reads activity of an EC2 instance

Incorrect

CloudWatch has available Amazon EC2 Metrics for you to use for monitoring. CPU Utilization identifies the processing power required to run an application upon a selected instance. Network Utilization identifies the volume of incoming and outgoing network traffic to a single instance. Disk Reads metric is used to determine the volume of the data the application reads from the hard disk of the instance. This can be used to determine the speed of the application. However, there are certain metrics that are not readily available in CloudWatch such as memory utilization, disk space utilization, and many others which can be collected by setting up a custom metric.

You need to prepare a custom metric using CloudWatch Monitoring Scripts which is written in Perl. You can also install CloudWatch Agent to collect more system-level metrics from Amazon EC2 instances. Here's the list of custom metrics that you can set up:

- Memory utilization
- Disk swap utilization
- Disk space utilization
- Page file utilization
- Log collection



CPU Utilization of an EC2 instance, *Disk Reads activity of an EC2 instance*, and *Network packets out of an EC2 instance* are all incorrect because these metrics are readily available in CloudWatch by default.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html#using_put_script

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

46. QUESTION

Category: CSAA – Design High-Performing Architectures

A cryptocurrency trading platform is using an API built in AWS Lambda and API Gateway. Due to the recent news and rumors about the upcoming price surge of Bitcoin, Ethereum and other cryptocurrencies, it is expected that the trading platform would have a significant increase in site visitors and new users in the coming days ahead.

In this scenario, how can you protect the backend systems of the platform from traffic spikes?

- Move the Lambda function in a VPC.
- Enable throttling limits and result caching in API Gateway.
- Use CloudFront in front of the API Gateway to act as a cache.

- Switch from using AWS Lambda and API Gateway to a more scalable and highly available architecture using EC2 instances, ELB, and Auto Scaling.

Correct

Amazon API Gateway provides throttling at multiple levels including global and by service call. Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds. Amazon API Gateway tracks the number of requests per second. Any request over the limit will receive a 429 HTTP response. The client SDKs generated by Amazon API Gateway retry calls automatically when met with this response. Hence, ***enabling throttling limits and result caching in API Gateway*** is the correct answer.

You can add caching to API calls by provisioning an Amazon API Gateway cache and specifying its size in gigabytes. The cache is provisioned for a specific stage of your APIs. This improves performance and reduces the traffic sent to your back end. Cache settings allow you to control the way the cache key is built and the time-to-live (TTL) of the data stored for each method. Amazon API Gateway also exposes management APIs that help you invalidate the cache for each stage.

The screenshot shows the AWS API Gateway Stage Editor for the 'prod' stage. The left sidebar lists various API components: APIs, ConcurrencyTest, Resources, **Stages** (which is selected and highlighted with an orange circle), Authorizers, Gateway Responses, Models, Resource Policy, Documentation, Dashboard, Settings, PetStore, Usage Plans, API Keys, Custom Domain Names, and Client Certificates. The main area shows the 'prod Stage Editor' with the 'Invoke URL' as <https://execute-api.us-east-2.amazonaws.com/prod>. The top navigation bar includes 'Delete St', 'Create', and tabs for 'Settings' (which is selected and highlighted with an orange circle), Logs, Stage Variables, SDK Generation, Export, and Deployment History. Below the tabs are sub-tabs for Documentation History and Canary. The 'Settings' tab contains sections for Cache Settings (with an 'Enable API cache' checkbox), Default Method Throttling (with an 'Enable throttling' checkbox checked, highlighted with an orange circle, and settings for Rate: 20 requests per second and Burst: 10 requests), and Client Certificate.

The option that says: ***Switch from using AWS Lambda and API Gateway to a more scalable and highly available architecture using EC2 instances, ELB, and Auto Scaling*** is incorrect since there is no need to transfer your applications to other services.

Using CloudFront in front of the API Gateway to act as a cache is incorrect because CloudFront only speeds up content delivery which provides a better latency experience for your users. It does not help much for the backend.

Moving the Lambda function in a VPC is incorrect because this answer is irrelevant to what is being asked. A VPC is your own virtual private cloud where you can launch AWS services.

Reference:

<https://aws.amazon.com/api-gateway/faqs/>

Check out this Amazon API Gateway Cheat Sheet:

<https://tutorialsdojo.com/amazon-api-gateway/>

Here is an in-depth tutorial on Amazon API Gateway:

<https://youtu.be/XwfpPEFHkTQ>

47. QUESTION

Category: CSAA – Design Resilient Architectures

A multi-tiered application hosted in your on-premises data center is scheduled to be migrated to AWS. The application has a message broker service which uses industry standard messaging APIs and protocols that must be migrated as well, without rewriting the messaging code in your application.

Which of the following is the most suitable service that you should use to move your messaging service to AWS?

- **Amazon MQ**
- Amazon SQS
- Amazon SNS
- Amazon SWF

Incorrect

Amazon MQ, Amazon SQS, and Amazon SNS are messaging services that are suitable for anyone from startups to enterprises. If you're using messaging with existing applications and want to move your messaging service to the cloud quickly and easily, it is recommended that you consider Amazon MQ. It supports industry-standard APIs and protocols so you can switch from any standards-based message broker to Amazon MQ without rewriting the messaging code in your applications.

Hence, ***Amazon MQ*** is the correct answer.

Amazon MQ

managed message broker service for Apache ActiveMQ

Amazon MQ is a managed message broker service for ActiveMQ that makes it easy to set up and operate message brokers in the cloud, so you can migrate your messaging and applications without rewriting code.

Benefits

Accelerate migration

Amazon MQ supports industry-standard APIs and protocols so you can migrate messaging and applications without rewriting code.

Offload operations

Amazon MQ manages the administration and maintenance of ActiveMQ brokers and automatically provisions infrastructure for high availability.

Reduce cost

Amazon MQ provides cost-efficient and flexible messaging capacity - you pay for broker instance and storage usage as you go.

Related services

[Amazon SQS](#)

Amazon SQS is a fully managed and highly scalable message queuing service for distributed applications and systems.

[Amazon SNS](#)

Amazon SNS is a fully managed pub/sub messaging and mobile notification service with nearly unlimited throughput.

Create a broker

Broker name

MyBroker

[Next step](#)

Pricing & costs (US)

mq.t2.micro	\$0.03 per hour
-------------	-----------------

mq.m4.large	\$0.3 per hour
-------------	----------------

Storage	\$0.3 per GB-month
---------	--------------------

[Learn more](#)

Documentation

[Developer Guide](#)

[API Reference](#)

[FAQs](#)

[Support forums](#)

If you are building brand new applications in the cloud, then it is highly recommended that you consider [Amazon SQS](#) and [Amazon SNS](#). Amazon SQS and SNS are lightweight, fully managed message queue and topic services that scale almost infinitely and provide simple, easy-to-use APIs. You can use Amazon SQS and SNS to decouple and scale microservices, distributed systems, and serverless applications, and improve reliability.

Amazon SQS is incorrect because although this is a fully managed message queuing service, it does not support an extensive list of industry-standard messaging APIs and protocol, unlike Amazon MQ. Moreover, using Amazon SQS requires you to do additional changes in the messaging code of applications to make it compatible.

Amazon SNS is incorrect because SNS is more suitable as a pub/sub messaging service instead of a message broker service.

Amazon SWF is incorrect because this is a fully-managed state tracker and task coordinator service and not a messaging service, unlike Amazon MQ, AmazonSQS and Amazon SNS.

References:

<https://aws.amazon.com/amazon-mq/faqs/>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html#sqsdifference-from-amazon-mq-sns>

Check out this Amazon MQ Cheat Sheet:

<https://tutorialsdojo.com/amazon-mq/>

48. QUESTION

Category: CSAA – Design High-Performing Architectures

An application hosted in EC2 consumes messages from an SQS queue and is integrated with SNS to send out an email to you once the process is complete. The Operations team received 5 orders but after a few hours, they saw 20 email notifications in their inbox.

Which of the following could be the possible culprit for this issue?

- The web application is set for long polling so the messages are being sent twice.
- **The web application is not deleting the messages in the SQS queue after it has processed them.**
- The web application does not have permission to consume messages in the SQS queue.
- The web application is set to short polling so some messages are not being picked up.

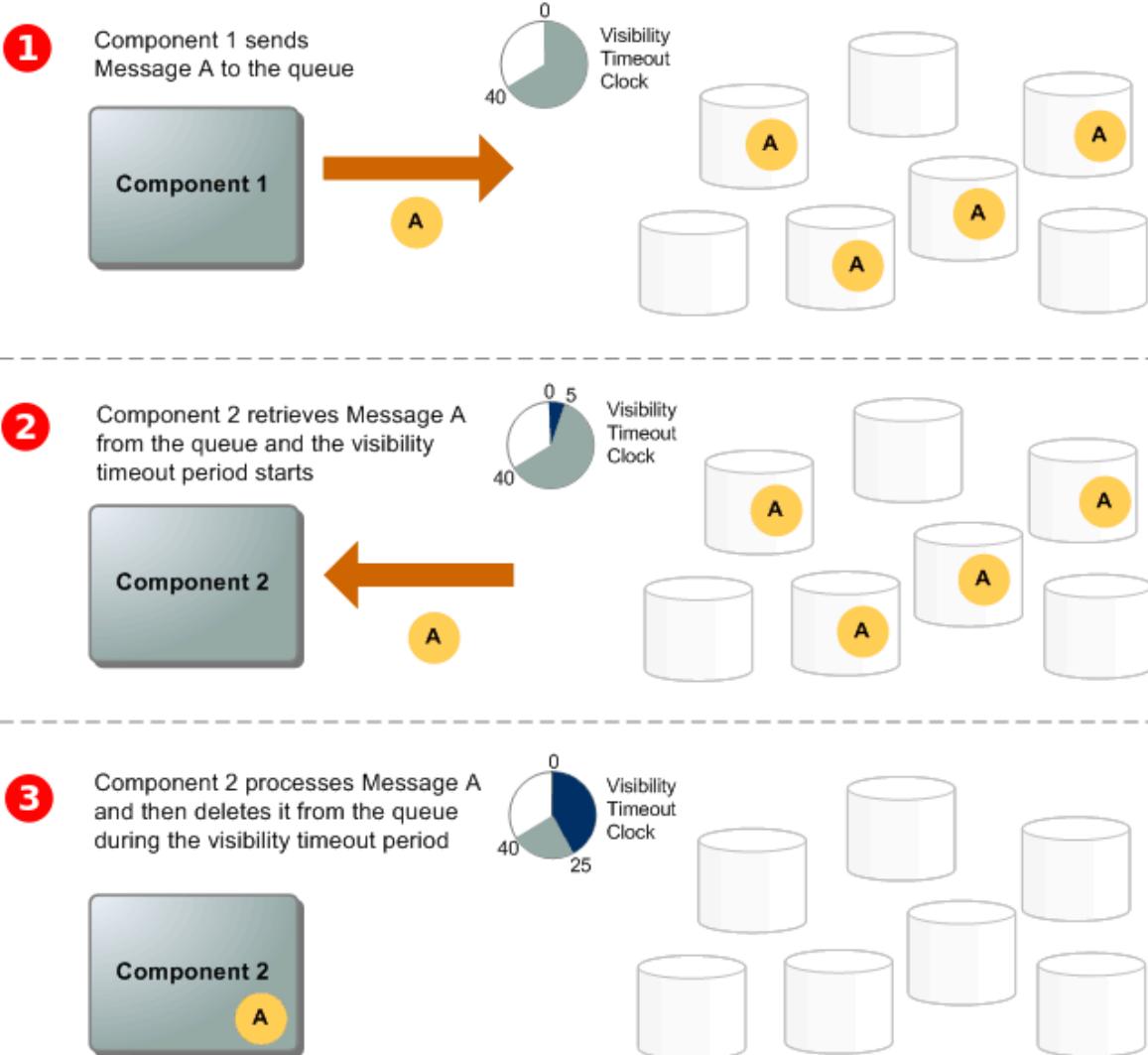
Correct

Always remember that the messages in the SQS queue will continue to exist even after the EC2 instance has processed it, until you delete that message. You have to ensure that you delete the message after processing to prevent the message from being received and processed again once the visibility timeout expires.

There are three main parts in a distributed messaging system:

- \1. The components of your distributed system (EC2 instances)
- \2. Your queue (distributed on Amazon SQS servers)
- \3. Messages in the queue.

You can set up a system which has several components that send messages to the queue and receive messages from the queue. The queue redundantly stores the messages across multiple Amazon SQS servers.



Refer to the third step of the SQS Message Lifecycle:

1. Component 1 sends Message A to a queue, and the message is distributed across the Amazon SQS servers redundantly.
2. When Component 2 is ready to process a message, it consumes messages from the queue, and Message A is returned. While Message A is being processed, it remains in the queue and isn't returned to subsequent receive requests for the duration of the visibility timeout.
3. Component 2 **deletes** Message A from the queue to prevent the message from being received and processed again once the visibility timeout expires.

The option that says: ***The web application is set for long polling so the messages are being sent twice*** is incorrect because long polling helps reduce the cost of using SQS by eliminating the number of empty responses (when there are no messages available for a ReceiveMessage request) and false empty responses (when messages are available but aren't included in a response). Messages being sent twice in an SQS queue configured with long polling is quite unlikely.

The option that says: ***The web application is set to short polling so some messages are not being picked up*** is incorrect since you are receiving emails from SNS where messages are certainly being processed. Following the scenario, messages not being picked up won't result into 20 messages being sent to your inbox.

The option that says: ***The web application does not have permission to consume messages in the SQS queue*** is incorrect because not having the correct permissions would have resulted in a different response. The scenario says that messages were properly processed but there were over 20 messages that were sent, hence, there is no problem with the accessing the queue.

References:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-message-lifecycle.html>

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-basic-architecture.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

49. QUESTION

Category: CSAA – Design High-Performing Architectures

A company collects atmospheric data such as temperature, air pressure, and humidity from different countries. Each site location is equipped with various weather instruments and a high-speed Internet connection. The average collected data in each location is around 500 GB and will be analyzed by a weather forecasting application hosted in Northern Virginia. As the Solutions Architect, you need to aggregate all the data in the fastest way.

Which of the following options can satisfy the given requirement?

- Set up a Site-to-Site VPN connection.
- Use AWS Snowball Edge to transfer large amounts of data.
- Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket.
- **Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.**

Correct

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers industry-leading durability, availability, performance, security, and virtually unlimited scalability at very low costs. Amazon S3 is also designed to be highly flexible. Store any type and amount of data that you want; read the same piece of data a million times or only for emergency disaster recovery; build a simple FTP application or a sophisticated web application.



Amazon S3 Transfer Acceleration

Speed Comparison

Upload speed comparison in the selected region
(Based on the location of bucket: jbarr-public)

N. Virginia
(US-EAST-1)

539% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

This speed checker uses multipart uploads to transfer a file from your browser to various Amazon S3 regions with and without Amazon S3 Transfer Acceleration. It compares the speed results and shows the percentage difference for every region.

Note: In general, the farther away you are from an Amazon S3 region, the higher the speed improvement you can expect from using Amazon S3 Transfer Acceleration. If you see similar speed results with and without the acceleration, your upload bandwidth or a system constraint might be limiting your speed.

Upload speed comparison in other regions

N. California
(US-WEST-1)

73% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Oregon
(US-WEST-2)

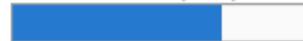
17% slower

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Ireland
(EU-WEST-1)

919% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Frankfurt
(EU-CENTRAL-1)

928% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Tokyo
(AP-NORTHEAST-1)

680% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Seoul
(AP-NORTHEAST-2)

822% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Singapore
(AP-SOUTHEAST-1)

1261% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Sydney
(AP-SOUTHEAST-2)

1226% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

São Paulo
(SA-EAST-1)

1000% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Since the weather forecasting application is located in N.Virginia, you need to transfer all the data in the same AWS Region. With Amazon S3 Transfer Acceleration, you can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Multipart upload allows you to upload a single object as a set of parts. After all the parts of your object are uploaded, Amazon S3 then presents the data as a single object. This approach is the fastest way to aggregate all the data.

Hence, the correct answer is: ***Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.***

The option that says: ***Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket*** is incorrect because replicating the objects to the destination bucket takes about 15 minutes. Take note that the requirement in the scenario is to aggregate the data in the fastest way.

The option that says: ***Use AWS Snowball Edge to transfer large amounts of data*** is incorrect because the end-to-end time to transfer up to 80 TB of data into AWS Snowball Edge is approximately one week.

The option that says: ***Set up a Site-to-Site VPN connection*** is incorrect because setting up a VPN connection is not needed in this scenario. Site-to-Site VPN is just used for establishing secure connections between an on-premises network and Amazon VPC. Also, this approach is not the fastest way to transfer your data. You must use Amazon S3 Transfer Acceleration.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/replication.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

50. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A travel photo sharing website is using Amazon S3 to serve high-quality photos to visitors of your website. After a few days, you found out that there are other travel websites linking and using your photos. This resulted in financial losses for your business.

What is the MOST effective method to mitigate this issue?

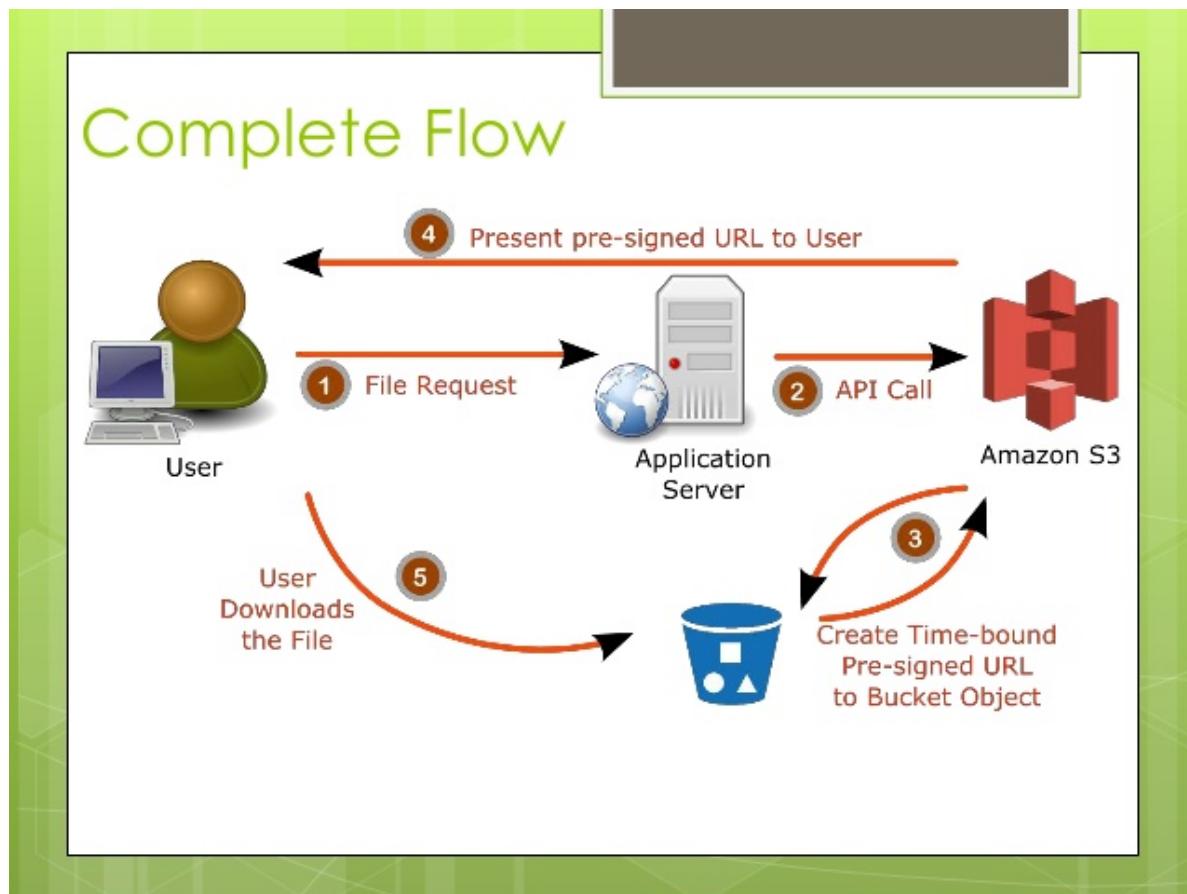
- Use CloudFront distributions for your photos.
- Block the IP addresses of the offending websites using NACL.
- Store and privately serve the high-quality photos on Amazon WorkDocs instead.
- **Configure your S3 bucket to remove public read access and use pre-signed URLs with expiry dates.**

Correct

In Amazon S3, all objects are private by default. Only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a pre-signed URL, using their own security credentials, to grant time-limited permission to download the objects.

When you create a pre-signed URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The pre-signed URLs are valid only for the specified duration.

Anyone who receives the pre-signed URL can then access the object. For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a pre-signed URL.



Using CloudFront distributions for your photos is incorrect. CloudFront is a content delivery network service that speeds up delivery of content to your customers.

Blocking the IP addresses of the offending websites using NACL is also incorrect. Blocking IP address using NACLs is not a very efficient method because a quick change in IP address would easily bypass this configuration.

Storing and privately serving the high-quality photos on Amazon WorkDocs instead is incorrect as WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service. It is not a suitable service for storing static content. Amazon WorkDocs is more often used to easily create, edit, and share documents for collaboration and not for serving object data like Amazon S3.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectOperations.html>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

<https://tutorialsdojo.com/s3-pre-signed-urls-vs-cloudfront-signed-urls-vs-origin-access-identity-oai/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

51. QUESTION

Category: CSAA – Design High-Performing Architectures

A company plans to build a data analytics application in AWS which will be deployed in an Auto Scaling group of On-Demand EC2 instances and a MongoDB database. It is expected that the database will have high-throughput workloads performing small, random I/O operations. As the Solutions Architect, you are required to properly set up and launch the required resources in AWS.

Which of the following is the most suitable EBS type to use for your database?

- Throughput Optimized HDD (st1)
- Provisioned IOPS SSD (io1)
- Cold HDD (sc1)
- General Purpose SSD (gp2)

Incorrect

On a given volume configuration, certain I/O characteristics drive the performance behavior for your EBS volumes. SSD-backed volumes, such as General Purpose SSD (gp2) and Provisioned IOPS SSD (io1), deliver consistent performance whether an I/O operation is random or sequential. HDD-backed volumes like Throughput Optimized HDD (st1) and Cold HDD (sc1) deliver optimal performance only when I/O operations are large and sequential.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has ***small, random*** I/O operations or ***large, sequential*** I/O operations.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

Provisioned IOPS SSD (`io1`) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike `gp2`, which uses a bucket and credit model to calculate performance, an `io1` volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers within 10 percent of the provisioned IOPS performance 99.9 percent of the time over a given year.

Volume Type	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
	General Purpose SSD (<code>gp2</code>)*	Provisioned IOPS SSD (<code>io1</code>)	Throughput Optimized HDD (<code>st1</code>)	Cold HDD (<code>sc1</code>)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none"> • Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: <ul style="list-style-type: none"> ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle 	<ul style="list-style-type: none"> • Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume 	<ul style="list-style-type: none"> • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume
API Name	<code>gp2</code>	<code>io1</code>	<code>st1</code>	<code>sc1</code>
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance††	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

General Purpose SSD (`gp2`) is incorrect because although General Purpose is a type of SSD that can handle small, random I/O operations, the Provisioned IOPS SSD volumes are much more suitable to meet the needs of I/O-intensive database workloads such as MongoDB, Oracle, MySQL, and many others.

Throughput Optimized HDD (`st1`) and ***Cold HDD (`sc1`)*** are incorrect because HDD volumes (such as Throughput Optimized HDD and Cold HDD volumes) are more suitable for workloads with large, sequential I/O operations instead of small, random I/O operations.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html#EBSVolumeTypes_piops

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-io-characteristics.html>

Amazon EBS Overview – SSD vs HDD:

<https://www.youtube.com/embed/LW7x8wyLFvw>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

52. QUESTION

Category: CSAA – Design Resilient Architectures

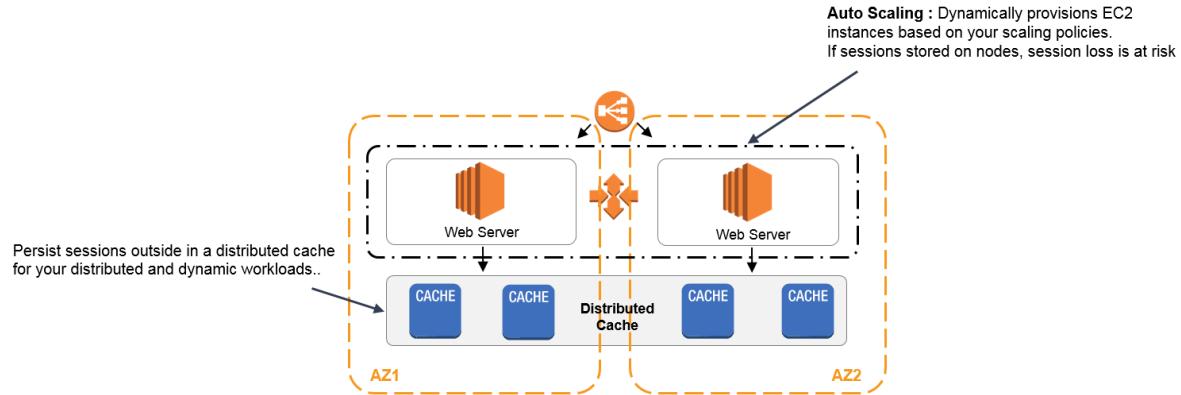
An IT consultant is working for a large financial company. The role of the consultant is to help the development team build a highly available web application using stateless web servers.

In this scenario, which AWS services are suitable for storing session state data? (Select TWO.)

- Redshift Spectrum
- DynamoDB
- RDS
- ElastiCache
- S3 Glacier

Correct

DynamoDB and ***ElastiCache*** are the correct answers. You can store session state data on both DynamoDB and ElastiCache. These AWS services provide high-performance storage of key-value pairs which can be used to build a highly available web application.



Redshift Spectrum is incorrect since this is a data warehousing solution where you can directly query data from your data warehouse. Redshift is not suitable for storing session state, but more on analytics and OLAP processes.

RDS is incorrect as well since this is a relational database solution of AWS. This relational storage type might not be the best fit for session states, and it might not provide the performance you need compared to DynamoDB for the same cost.

S3 Glacier is incorrect since this is a low-cost cloud storage service for data archiving and long-term backup. The archival and retrieval speeds of Glacier is too slow for handling session states.

References:

<https://aws.amazon.com/caching/database-caching/>

<https://aws.amazon.com/caching/session-management/>

Check out this Amazon ElastiCache Cheat Sheet:

<https://tutorialsdojo.com/amazon-elasticache/>

53. QUESTION

Category: CSAA – Design High-Performing Architectures

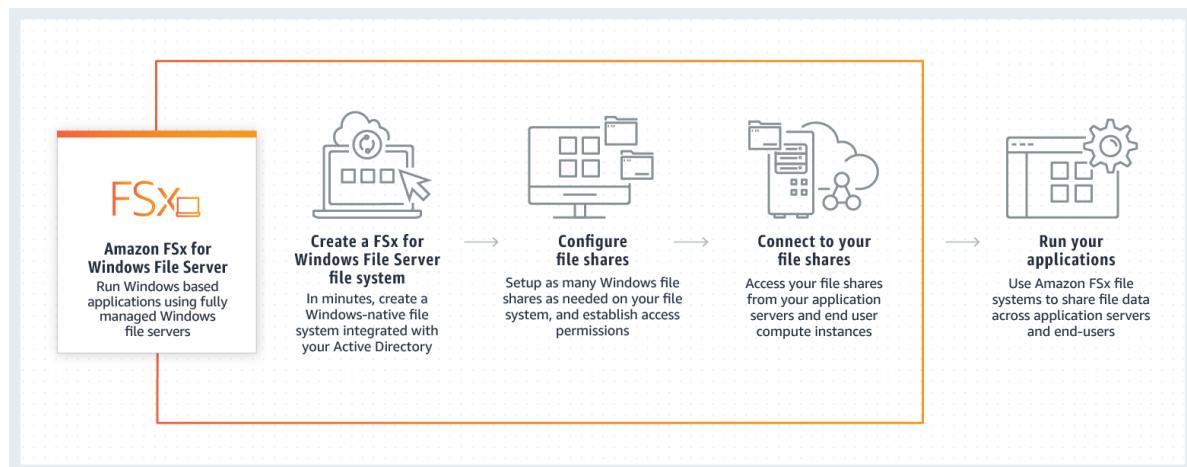
A company has a web application that uses Internet Information Services (IIS) for Windows Server. A file share is used to store the application data on the network-attached storage of the company's on-premises data center. To achieve a highly available system, they plan to migrate the application and file share to AWS.

Which of the following can be used to fulfill this requirement?

- Migrate the existing file share configuration to Amazon FSx for Windows File Server.
- Migrate the existing file share configuration to Amazon EBS.
- Migrate the existing file share configuration to Amazon EFS.
- Migrate the existing file share configuration to AWS Storage Gateway.

Correct

Amazon FSx for Windows File Server provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system. Amazon FSx for Windows File Server has the features, performance, and compatibility to easily lift and shift enterprise applications to the AWS Cloud. It is accessible from Windows, Linux, and macOS compute instances and devices. Thousands of compute instances and devices can access a file system concurrently.



In this scenario, you need to migrate your existing file share configuration to the cloud. Among the options given, the best possible answer is Amazon FSx. A file share is a specific folder in your file system, including the folder's subfolders, which you make accessible to your compute instances via the SMB protocol. To migrate file share configurations from your on-premises file system, you must migrate your

files first to Amazon FSx before migrating your file share configuration.

Hence, the correct answer is: ***Migrate the existing file share configuration to Amazon FSx for Windows File Server***.

The option that says: ***Migrate the existing file share configuration to AWS Storage Gateway*** is incorrect because AWS Storage Gateway is primarily used to integrate your on-premises network to AWS but not for migrating your applications. Using a file share in Storage Gateway implies that you will still keep your on-premises systems, and not entirely migrate it.

The option that says: ***Migrate the existing file share configuration to Amazon EFS*** is incorrect because it is stated in the scenario that the company is using a file share that runs on a Windows server. Remember that Amazon EFS only supports Linux workloads.

The option that says: ***Migrate the existing file share configuration to Amazon EBS*** is incorrect because EBS is primarily used as block storage for EC2 instances and not as a shared file system. A file share is a specific folder in a file system that you can access using a server message block (SMB) protocol. Amazon EBS does not support SMB protocol.

References:

<https://aws.amazon.com/fsx/windows/faqs/>

<https://docs.aws.amazon.com/fsx/latest/WindowsGuide/migrate-file-share-config-to-fsx.html>

Check out this Amazon FSx Cheat Sheet:

<https://tutorialsdojo.com/amazon-fsx/>

54. QUESTION

Category: CSAA – Design Resilient Architectures

An online shopping platform is hosted on an Auto Scaling group of Spot EC2 instances and uses Amazon Aurora PostgreSQL as its database. There is a requirement to optimize your database workloads in your cluster where you have to direct the production traffic to your high-capacity instances and point the reporting queries sent by your internal staff to the low-capacity instances.

Which is the most suitable configuration for your application as well as your Aurora database cluster to achieve this requirement?

- Create a custom endpoint in Aurora based on the specified criteria for the production traffic and another custom endpoint to handle the reporting queries.
- Do nothing since by default, Aurora will automatically direct the production traffic to your high-capacity instances and the reporting queries to your low-capacity instances.
- In your application, use the instance endpoint of your Aurora database to handle the incoming production traffic and use the cluster endpoint to handle reporting queries.
- Configure your application to use the reader endpoint for both production traffic and reporting queries, which will enable your Aurora database to automatically perform load-balancing among all the Aurora Replicas.

Incorrect

Amazon Aurora typically involves a cluster of DB instances instead of a single instance. Each connection is handled by a specific DB instance. When you connect to an Aurora cluster, the host name and port that you specify point to an intermediate handler called an *endpoint*. Aurora uses the endpoint mechanism to abstract these connections. Thus, you don't have to hardcode all the hostnames or write your own logic for load-balancing and rerouting connections when some DB instances aren't available.

For certain Aurora tasks, different instances or groups of instances perform different roles. For example, the primary instance handles all data definition language (DDL) and data manipulation language (DML) statements. Up to 15 Aurora Replicas handle read-only query traffic.

The screenshot shows the AWS Aurora console interface. At the top, there is a table listing three DB instances under the cluster 'tutorialsdojo-1'. The columns include DB Identifier, Role, Engine, Class, Status, and CPU usage. Below this, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The 'Connectivity & security' tab is selected. Under this tab, there is a section titled 'Endpoints (2)' with a 'Create custom endpoint' button. A search bar labeled 'Filter endpoint' is also present. Below this, another table lists two endpoints, each with details like Endpoint name, Status, Type, and Port.

DB Identifier	Role	Engine	Class	Status	CPU
tutorialsdojo-1	Cluster	Aurora MySQL	-	Available	
tutorialsdojo	Writer	Aurora MySQL	db.t2.small	Available	9.33%
tutorialsdojo-ap-southeast-2b	Reader	Aurora MySQL	db.t2.small	Available	7.70%

Endpoint name	Status	Type	Port
tutorialsdojo-1.cluster-ro-cerpn6ov4bsw.ap-southeast-2.rds.amazonaws.com	Available	Reader	3306
tutorialsdojo-1.cluster-cerpn6ov4bsw.ap-southeast-2.rds.amazonaws.com	Available	Writer	3306

Using endpoints, you can map each connection to the appropriate instance or group of instances based on your use case. For example, to perform DDL statements you can connect to whichever instance is the primary instance. To perform queries, you can connect to the reader endpoint, with Aurora automatically performing load-balancing among all the Aurora Replicas. For clusters with DB instances of different capacities or configurations, you can connect to custom endpoints associated with different subsets of DB instances. For diagnosis or tuning, you can connect to a specific instance endpoint to examine details about a specific DB instance.

The custom endpoint provides load-balanced database connections based on criteria other than the read-only or read-write capability of the DB instances. For example, you might define a custom endpoint to connect to instances that use a particular AWS instance class or a particular DB parameter group. Then you might tell particular groups of users about this custom endpoint. For example, you might direct internal users to low-capacity instances for report generation or ad hoc (one-time) querying, and direct production traffic to high-capacity instances. Hence, ***creating a custom endpoint in Aurora based on the specified criteria for the production traffic and another custom endpoint to handle the reporting queries*** is the correct answer.

Configuring your application to use the reader endpoint for both production traffic and reporting queries, which will enable your Aurora database to automatically perform load-balancing among all the Aurora Replicas is incorrect because although it is true that a reader endpoint enables your Aurora database to automatically perform load-balancing among all the Aurora Replicas, it is quite limited to doing read operations only. You still need to use a custom endpoint to load-balance the database connections based on the specified criteria.

The option that says: ***In your application, use the instance endpoint of your Aurora database to handle the incoming production traffic and use the cluster endpoint to handle reporting queries*** is incorrect because a cluster endpoint (also known as a writer endpoint) for an Aurora DB cluster simply connects to the current primary DB instance for that DB cluster. This endpoint can perform write operations in the database such as DDL statements, which is perfect for handling production traffic but not suitable for handling queries for reporting since there will be no write database operations that will be sent. Moreover, the endpoint does not point to lower-capacity or high-capacity instances as per the requirement. A better solution for this is to use a custom endpoint.

The option that says: ***Do nothing since by default, Aurora will automatically direct the production traffic to your high-capacity instances and the reporting queries to your low-capacity instances*** is incorrect because Aurora does not do this by default. You have to create custom endpoints in order to accomplish this requirement.

Reference:

<https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.Endpoints.html>

Check out this Amazon Aurora Cheat Sheet:

<https://tutorialsdojo.com/amazon-aurora/>

55. QUESTION

Category: CSAA – Design High-Performing Architectures

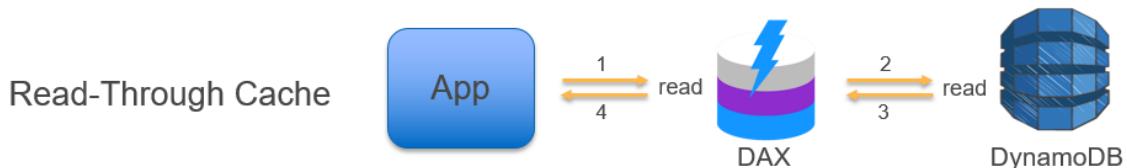
A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time.

To improve the game's performance, which AWS service can you use to reduce DynamoDB response times from milliseconds to microseconds?

- Amazon ElastiCache
- AWS Device Farm
- DynamoDB Auto Scaling
- **Amazon DynamoDB Accelerator (DAX)**

Correct

Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache that can reduce Amazon DynamoDB response times from milliseconds to microseconds, even at millions of requests per second.



Amazon ElastiCache is incorrect because although you may use ElastiCache as your database cache, it will not reduce the DynamoDB response time from milliseconds to microseconds as compared with DynamoDB DAX.

AWS Device Farm is incorrect because this is an app testing service that lets you test and interact with your Android, iOS, and web apps on many devices at once, or reproduce issues on a device in real time.

DynamoDB Auto Scaling is incorrect because this is primarily used to automate capacity management for your tables and global secondary indexes.

References:

<https://aws.amazon.com/dynamodb/dax>

<https://aws.amazon.com/device-farm>

Check out this Amazon DynamoDB Cheat Sheet:

56. QUESTION

Category: CSAA – Design Secure Applications and Architectures

In a government agency that you are working for, you have been assigned to put confidential tax documents on AWS cloud. However, there is a concern from a security perspective on what can be put on AWS.

What are the features in AWS that can ensure data security for your confidential documents? (Select TWO.)

- EBS On-Premises Data Encryption
- S3 Server-Side Encryption
- S3 On-Premises Data Encryption
- S3 Client-Side Encryption
- Public Data Set Volume Encryption

Incorrect

You can secure the privacy of your data in AWS, both at rest and in-transit, through encryption. If your data is stored in EBS Volumes, you can enable EBS Encryption and if it is stored on Amazon S3, you can enable ***client-side*** and ***server-side encryption***.

Public Data Set Volume Encryption is incorrect as public data sets are designed to be publicly accessible.

EBS On-Premises Data Encryption and ***S3 On-Premises Data Encryption*** are both incorrect as there is no such thing as On-Premises Data Encryption for S3 and EBS as these services are in the AWS cloud and not on your on-premises network.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-public-data-sets.html>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

57. QUESTION

Category: CSAA – Design Resilient Architectures

An organization needs a persistent block storage volume that will be used for mission-critical workloads. The backup data will be stored in an object storage service and after 30 days, the data will be stored in a data archiving storage service.

What should you do to meet the above requirement?

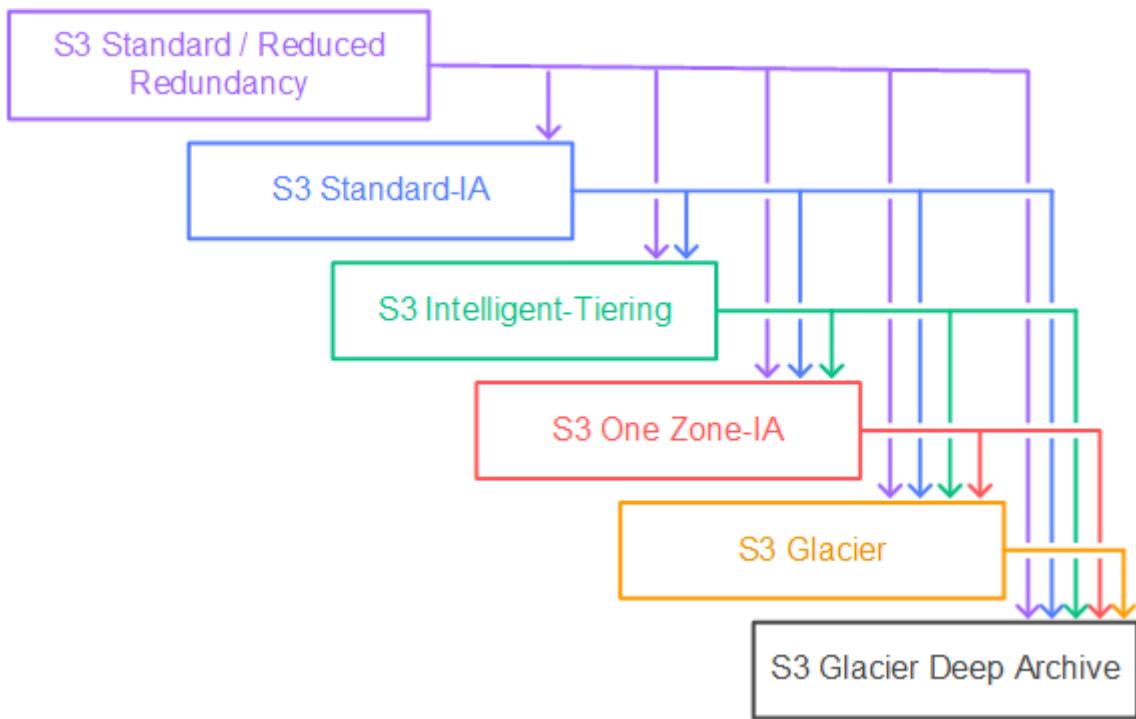
- Attach an instance store volume in your existing EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 Glacier.
- Attach an instance store volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA.
- Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA.
- **Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 Glacier.**

Incorrect

Amazon Elastic Block Store (EBS) is an easy-to-use, high-performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale. A broad range of workloads, such as relational and non-relational databases, enterprise applications, containerized applications, big data analytics engines, file systems, and media workflows are widely deployed on Amazon EBS.

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics.

In an **S3 Lifecycle configuration**, you can define rules to transition objects from one storage class to another to save on storage costs. Amazon S3 supports a waterfall model for transitioning between storage classes, as shown in the diagram below:



In this scenario, three services are required to implement this solution. The mission-critical workloads mean that you need to have a persistent block storage volume and the designed service for this is Amazon EBS volumes. The second workload needs to have an object storage service, such as Amazon S3, to store your backup data. Amazon S3 enables you to configure the lifecycle policy from S3 Standard to different storage classes. For the last one, it needs archive storage such as Amazon S3 Glacier.

Hence, the correct answer in this scenario is: ***Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 Glacier.***

The option that says: ***Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA*** is incorrect because this lifecycle policy will transition your objects into an infrequently accessed storage class and not a storage class for data archiving.

The option that says: ***Attach an instance store volume in your existing EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 Glacier*** is incorrect because an Instance Store volume is simply a temporary block-level storage for EC2 instances. Also, you can't attach instance store volumes to an instance after you've launched it. You can specify the instance store volumes for your instance only when you launch it.

The option that says: ***Attach an instance store volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA*** is incorrect. Just like the previous option, the use of instance store volume is not suitable for mission-critical workloads because the data can be lost if the underlying disk drive fails, the instance stops, or if the instance is terminated. In addition, Amazon S3 Glacier is a more suitable option for data archival instead of Amazon S3 One Zone-IA.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>

<https://aws.amazon.com/s3/storage-classes/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Storage Services Cheat Sheets:

<https://tutorialsdojo.com/aws-cheat-sheets-storage-services/>

58. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect needs to make sure that the On-Demand EC2 instance can only be accessed from this IP address (110.238.98.71) via an SSH connection. Which configuration below will satisfy this requirement?

- Security Group Inbound Rule: Protocol – UDP, Port Range – 22, Source 110.238.98.71/0
- Security Group Inbound Rule: Protocol – TCP, Port Range – 22, Source 110.238.98.71/0
- Security Group Inbound Rule: Protocol – UDP, Port Range – 22, Source 110.238.98.71/32
- **Security Group Inbound Rule: Protocol – TCP, Port Range – 22, Source 110.238.98.71/32**

Incorrect

A **security group** acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign up to five security groups to the instance. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC can be assigned to a different set of security groups.

Type	Protocol	Port range	Source
HTTP	TCP	80	Custom 0.0.0.0/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0
SSH	TCP	22	Custom 110.238.98.71/32

The requirement is to only allow the individual IP of the client and not the entire network. Therefore, the proper CIDR notation should be used. The /32 denotes one IP address and the /0 refers to the entire network. Take note that the SSH protocol uses TCP and port 22.

Hence, the correct answer is: ***Protocol – TCP, Port Range – 22, Source 110.238.98.71/32***

Protocol – UDP, Port Range – 22, Source 110.238.98.71/32 and ***Protocol – UDP, Port Range – 22, Source 110.238.98.71/0*** are incorrect as they are using UDP.

Protocol – TCP, Port Range – 22, Source 110.238.98.71/0 is incorrect because it uses a /0 CIDR notation.

Protocol – TCP, Port Range – 22, Source 110.238.98.71/0 is incorrect because it allows the entire network instead of a single IP.

Reference:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html#security-group-rule>
S

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

59. QUESTION

Category: CSAA – Design Secure Applications and Architectures

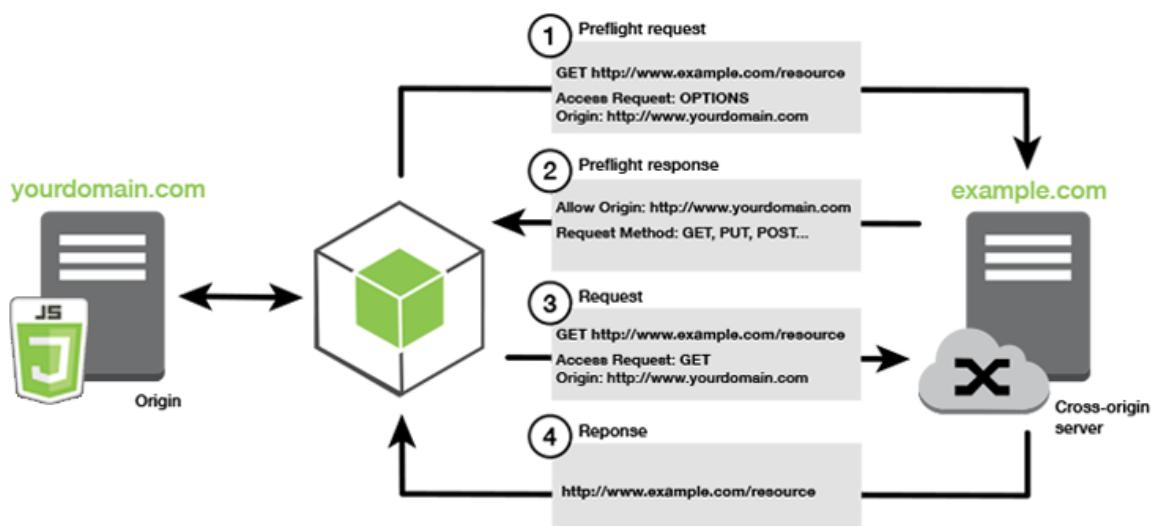
A Solutions Architect is hosting a website in an Amazon S3 bucket named `tutorialsdojo`. The users load the website using the following URL: `http://tutorialsdojo.s3-website-us-east-1.amazonaws.com` and there is a new requirement to add a JavaScript on the webpages in order to make authenticated HTTP `GET` requests against the same bucket by using the Amazon S3 API endpoint (`tutorialsdojo.s3.amazonaws.com`). Upon testing, you noticed that the web browser blocks JavaScript from allowing those requests.

Which of the following options is the MOST suitable solution that you should implement for this scenario?

- Enable cross-account access.
- Enable Cross-Region Replication (CRR).
- Enable Cross-Zone Load Balancing.
- Enable Cross-origin resource sharing (CORS) configuration in the bucket.

Correct

Cross-origin resource sharing (CORS) defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. With CORS support, you can build rich client-side web applications with Amazon S3 and selectively allow cross-origin access to your Amazon S3 resources.



Suppose that you are hosting a website in an Amazon S3 bucket named `your-website` and your users load the website endpoint `http://your-website.s3-website-us-east-1.amazonaws.com`. Now you want to use JavaScript on the webpages that are stored in this bucket to be able to make authenticated GET and PUT requests against the same bucket by using the Amazon S3 API endpoint for the bucket, `your-website.s3.amazonaws.com`. A browser would normally block JavaScript from allowing those requests,

but with CORS you can configure your bucket to explicitly enable cross-origin requests from `your-website.s3-website-us-east-1.amazonaws.com`.

In this scenario, you can solve the issue by enabling the CORS in the S3 bucket. Hence, ***enabling Cross-origin resource sharing (CORS) configuration in the bucket*** is the correct answer.

Enabling cross-account access is incorrect because cross-account access is a feature in IAM and not in Amazon S3.

Enabling Cross-Zone Load Balancing is incorrect because Cross-Zone Load Balancing is only used in ELB and not in S3.

Enabling Cross-Region Replication (CRR) is incorrect because CRR is a bucket-level configuration that enables automatic, asynchronous copying of objects across buckets in different AWS Regions.

References:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ManageCorsUsing.html>

60. QUESTION

Category: CSAA – Design High-Performing Architectures

An AI-powered Forex trading application consumes thousands of data sets to train its machine learning model. The application's workload requires a high-performance, parallel hot storage to process the training datasets concurrently. It also needs cost-effective cold storage to archive those datasets that yield low profit.

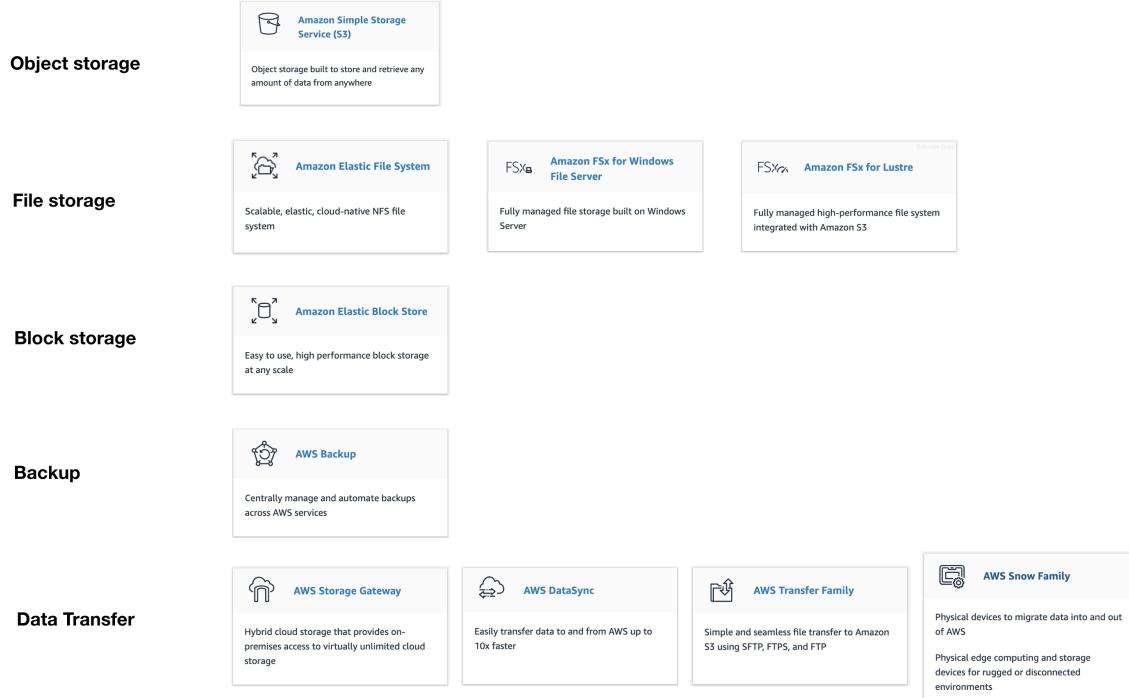
Which of the following Amazon storage services should the developer use?

- Use Amazon FSx For Lustre and Amazon S3 for hot and cold storage respectively.
- Use Amazon FSx For Lustre and Amazon EBS Provisioned IOPS SSD (io1) volumes for hot and cold storage respectively.
- Use Amazon FSx For Windows File Server and Amazon S3 for hot and cold storage respectively.
- Use Amazon Elastic File System and Amazon S3 for hot and cold storage respectively.

Correct

Hot storage refers to the storage that keeps frequently accessed data (hot data). **Warm storage** refers to the storage that keeps less frequently accessed data (warm data). **Cold storage** refers to the storage that keeps rarely accessed data (cold data). In terms of pricing, the colder the data, the cheaper it is to store, and the costlier it is to access when needed.

aws STORAGE SERVICES



Amazon FSx For Lustre is a high-performance file system for fast processing of workloads. Lustre is a popular open-source **parallel file system** which stores data across multiple network file servers to maximize performance and reduce bottlenecks.

Amazon FSx for Windows File Server is a fully managed Microsoft Windows file system with full support for the SMB protocol, Windows NTFS, Microsoft Active Directory (AD) Integration.

Amazon Elastic File System is a fully-managed file storage service that makes it easy to set up and scale file storage in the Amazon Cloud.

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. S3 offers different storage tiers for different use cases (frequently accessed data, infrequently accessed data, and rarely accessed data).

The question has two requirements:

1. High-performance, parallel hot storage to process the training datasets concurrently.
2. Cost-effective cold storage to keep the archived datasets that are accessed infrequently

In this case, we can use **Amazon FSx For Lustre** for the first requirement, as it provides a high-performance, parallel file system for hot data. On the second requirement, we can use Amazon S3 for storing the cold data. Amazon S3 supports a cold storage system via Amazon S3 Glacier / Glacier Deep Archive.

Hence, the correct answer is ***Use Amazon FSx For Lustre and Amazon S3 for hot and cold storage respectively***.

Using Amazon FSx For Lustre and Amazon EBS Provisioned IOPS SSD (io1) volumes for hot and cold storage respectively is incorrect because the Provisioned IOPS SSD (io1) volumes are designed as a hot storage to meet the needs of I/O-intensive workloads. EBS has a storage option called Cold HDD but it is not used for storing cold data. In addition, EBS Cold HDD is a lot more expensive than using Amazon S3 Glacier / Glacier Deep Archive.

Using Amazon Elastic File System and Amazon S3 for hot and cold storage respectively is incorrect because although EFS supports concurrent access to data, it does not have the high-performance ability that is required for machine learning workloads.

Using Amazon FSx For Windows File Server and Amazon S3 for hot and cold storage respectively is incorrect because **Amazon FSx For Windows File Server does not have a parallel file system, unlike Lustre.**

References:

<https://aws.amazon.com/fsx/>

<https://docs.aws.amazon.com/whitepapers/latest/cost-optimization-storage-optimization/aws-storage-services.html>

<https://aws.amazon.com/blogs/startups/picking-the-right-data-store-for-your-workload/>

Check out this Amazon FSx Cheat Sheet:

<https://tutorialsdojo.com/amazon-fsx/>

61. QUESTION

Category: CSAA – Design Resilient Architectures

An application that records weather data every minute is deployed in a fleet of Spot EC2 instances and uses a MySQL RDS database instance. Currently, there is only one RDS instance running in one Availability Zone. You plan to improve the database to ensure high availability by synchronous data replication to another RDS instance.

Which of the following performs synchronous data replication in RDS?

- **RDS DB instance running as a Multi-AZ deployment**
- RDS Read Replica
- DynamoDB Read Replica
- CloudFront running as a Multi-AZ deployment

Correct

When you create or modify your DB instance to run as a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous **standby** replica in a different Availability Zone. Updates to your DB Instance are synchronously replicated across Availability Zones to the standby in order to keep both in sync and protect your latest database updates against DB instance failure.

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance

RDS Read Replica is incorrect as a **Read Replica provides an asynchronous replication instead of synchronous.**

DynamoDB Read Replica and ***CloudFront running as a Multi-AZ deployment*** are incorrect as both DynamoDB and CloudFront do not have a Read Replica feature.

Reference:

<https://aws.amazon.com/rds/details/multi-az/>

Amazon RDS Overview:

<https://youtu.be/aZmpLI8K1UU>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

62. QUESTION

Category: CSAA – Design Resilient Architectures

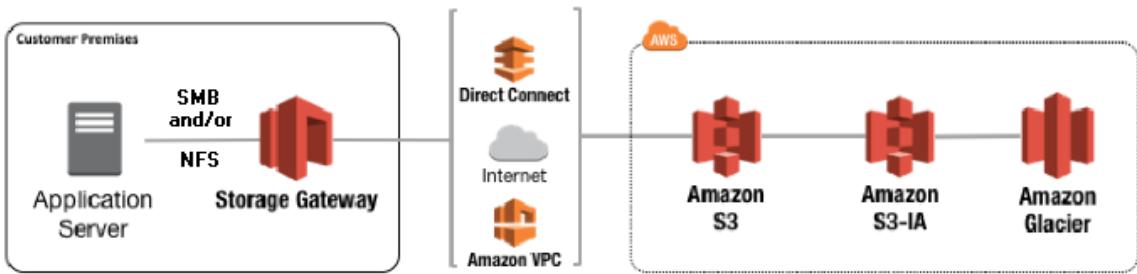
A company has a hybrid cloud architecture that connects their on-premises data center and cloud infrastructure in AWS. They require a durable storage backup for their corporate documents stored on-premises and a local cache that provides low latency access to their recently accessed data to reduce data egress charges. The documents must be stored to and retrieved from AWS via the Server Message Block (SMB) protocol. These files must immediately be accessible within minutes for six months and archived for another decade to meet the data compliance.

Which of the following is the best and most cost-effective approach to implement in this scenario?

- Launch a new tape gateway that connects to your on-premises data center using AWS Storage Gateway. Upload the documents to the tape gateway and set up a lifecycle policy to move the data into Glacier for archival.
- **Launch a new file gateway that connects to your on-premises data center using AWS Storage Gateway. Upload the documents to the file gateway and set up a lifecycle policy to move the data into Glacier for data archival.**
- Use AWS Snowmobile to migrate all of the files from the on-premises network. Upload the documents to an S3 bucket and set up a lifecycle policy to move the data into Glacier for archival.
- Establish a Direct Connect connection to integrate your on-premises network to your VPC. Upload the documents on Amazon EBS Volumes and use a lifecycle policy to automatically move the EBS snapshots to an S3 bucket, and then later to Glacier for archival.

Correct

A file gateway supports a file interface into Amazon Simple Storage Service (Amazon S3) and combines a service and a virtual software appliance. By using this combination, you can store and retrieve objects in Amazon S3 using industry-standard file protocols such as Network File System (NFS) and Server Message Block (SMB). The software appliance, or gateway, is deployed into your on-premises environment as a virtual machine (VM) running on VMware ESXi, Microsoft Hyper-V, or Linux Kernel-based Virtual Machine (KVM) hypervisor.



The gateway provides access to objects in S3 as files or file share mount points. With a file gateway, you can do the following:

- You can store and retrieve files directly using the NFS version 3 or 4.1 protocol.
 - You can store and retrieve files directly using the SMB file system version, 2 and 3 protocol.
 - You can access your data directly in Amazon S3 from any AWS Cloud application or service.
 - You can manage your Amazon S3 data using lifecycle policies, cross-region replication, and versioning.
- You can think of a file gateway as a file system mount on S3.

AWS Storage Gateway supports the Amazon S3 Standard, Amazon S3 Standard-Infrequent Access, Amazon S3 One Zone-Infrequent Access and Amazon Glacier storage classes. When you create or update a file share, you have the option to select a storage class for your objects. You can either choose the Amazon S3 Standard or any of the infrequent access storage classes such as S3 Standard IA or S3 One Zone IA. Objects stored in any of these storage classes can be transitioned to Amazon Glacier using a Lifecycle Policy.

Although you can write objects directly from a file share to the S3-Standard-IA or S3-One Zone-IA storage class, it is recommended that you use a Lifecycle Policy to transition your objects rather than write directly from the file share, especially if you're expecting to update or delete the object within 30 days of archiving it.

Therefore, the correct answer is: ***Launch a new file gateway that connects to your on-premises data center using AWS Storage Gateway. Upload the documents to the file gateway and set up a lifecycle policy to move the data into Glacier for data archival.***

The option that says: ***Launch a new tape gateway that connects to your on-premises data center using AWS Storage Gateway. Upload the documents to the tape gateway and set up a lifecycle policy to move the data into Glacier for archival*** is incorrect because although tape gateways provide cost-effective and durable archive backup data in Amazon Glacier, **it does not meet the criteria of being retrievable immediately within minutes. It also doesn't maintain a local cache that provides low latency access to the recently accessed data and reduce data egress charges.** Thus, it is still better to set up a file gateway instead.

The option that says: ***Establish a Direct Connect connection to integrate your on-premises network to your VPC. Upload the documents on Amazon EBS Volumes and use a lifecycle policy to automatically move the EBS snapshots to an S3 bucket, and then later to Glacier for archival*** is incorrect because EBS Volumes are not as durable compared with S3 and it would be more cost-efficient if you directly store the documents to an S3 bucket. An alternative solution is to use AWS Direct Connect with AWS Storage Gateway to create a connection for high-throughput workload needs, providing a dedicated network connection between your on-premises file gateway and AWS. But this solution is using EBS, hence, this option is still wrong.

The option that says: ***Use AWS Snowmobile to migrate all of the files from the on-premises network. Upload the documents to an S3 bucket and set up a lifecycle policy to move the data into Glacier for archival*** is incorrect because Snowmobile is mainly used to migrate the entire data of an on-premises data center to AWS. This is not a suitable approach as the company still has a hybrid cloud architecture.

which means that they will still use their on-premises data center along with their AWS cloud infrastructure.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html>

<https://docs.aws.amazon.com/storagegateway/latest/userguide/StorageGatewayConcepts.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate-saa-c02/>

63. QUESTION

Category: CSAA – Design Resilient Architectures

There was an incident in your production environment where the user data stored in the S3 bucket has been accidentally deleted by one of the Junior DevOps Engineers. The issue was escalated to your manager and after a few days, you were instructed to improve the security and protection of your AWS resources.

What combination of the following options will protect the S3 objects in your bucket from both accidental deletion and overwriting? (Select TWO.)

- **Enable Versioning**
- Provide access to S3 data strictly through pre-signed URL only
- Disallow S3 Delete using an IAM bucket policy
- Enable Amazon S3 Intelligent-Tiering
- **Enable Multi-Factor Authentication Delete**

Correct

By using Versioning and enabling MFA (Multi-Factor Authentication) Delete, you can secure and recover your S3 objects from accidental deletion or overwrite.

Versioning is a means of keeping multiple variants of an object in the same bucket. Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

You can also optionally add another layer of security by configuring a bucket to enable MFA (Multi-Factor Authentication) Delete, which requires additional authentication for either of the following operations:

- Change the versioning state of your bucket
- Permanently delete an object version

MFA Delete requires two forms of authentication together:

- Your security credentials
- The concatenation of a valid serial number, a space, and the six-digit code displayed on an approved authentication device

Providing access to S3 data strictly through pre-signed URL only is incorrect since a pre-signed URL gives access to the object identified in the URL. **Pre-signed URLs are useful when customers perform an object upload to your S3 bucket, but does not help in preventing accidental deletes.**

Disallowing S3 Delete using an IAM bucket policy is incorrect since you still want users to be able to delete objects in the bucket, and you just want to prevent accidental deletions. Disallowing S3 Delete using an IAM bucket policy will restrict all delete operations to your bucket.

Enabling Amazon S3 Intelligent-Tiering is incorrect since S3 intelligent tiering does not help in this situation.

Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

64. QUESTION

Category: CSAA – Design High-Performing Architectures

A company is using Amazon S3 to store frequently accessed data. When an object is created or deleted, the S3 bucket will send an event notification to the Amazon SQS queue. A solutions architect needs to create a solution that will notify the development and operations team about the created or deleted objects.

Which of the following would satisfy this requirement?

- Set up another Amazon SQS queue for the other team. Grant Amazon S3 permission to send a notification to the second SQS queue.
- Create a new Amazon SNS FIFO topic for the other team. Grant Amazon S3 permission to send the notification to the second SNS topic.
- Set up an Amazon SNS topic and configure two Amazon SQS queues to poll the SNS topic. Grant Amazon S3 permission to send notifications to Amazon SNS and update the bucket to use the new SNS topic.
- Create an Amazon SNS topic and configure two Amazon SOS queues to subscribe to the topic. Grant Amazon S3 permission to send notifications to Amazon SNS and update the bucket to use the new SNS topic.

Incorrect

The **Amazon S3** notification feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration that identifies the events you want Amazon S3 to publish and the destinations where you want Amazon S3 to send the notifications. You store this configuration in the notification subresource that is associated with a bucket.

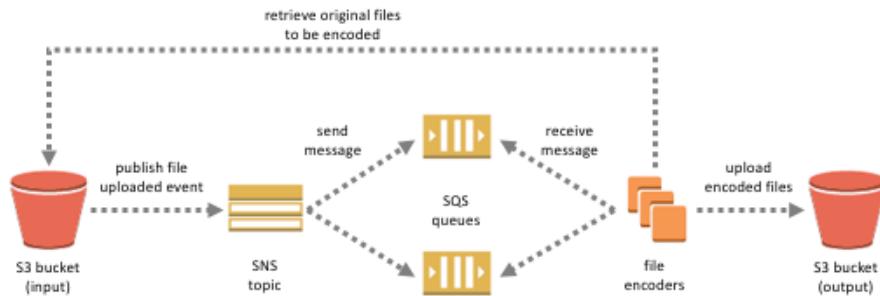
Amazon S3 supports the following destinations where it can publish events:

- Amazon Simple Notification Service (Amazon SNS) topic

- Amazon Simple Queue Service (Amazon SQS) queue

- AWS Lambda

In Amazon SNS, the *fanout* scenario is when a message published to an SNS topic is replicated and pushed to multiple endpoints, such as Amazon SQS queues, HTTP(S) endpoints, and Lambda functions. This allows for parallel asynchronous processing.



For example, you can develop an application that publishes a message to an SNS topic whenever an order is placed for a product. Then, SQS queues that are subscribed to the SNS topic receive identical notifications for the new order. An Amazon Elastic Compute Cloud (Amazon EC2) server instance attached to one of the SQS queues can handle the processing or fulfillment of the order. And you can attach another Amazon EC2 server instance to a data warehouse for analysis of all orders received.

Based on the given scenario, the existing setup sends the event notification to an SQS queue. Since you need to send the notification to the development and operations team, you can use a combination of Amazon SNS and SQS. By using the message fanout pattern, you can create a topic and use two Amazon SQS queues to subscribe to the topic. If Amazon SNS receives an event notification, it will publish the message to both subscribers.

Take note that Amazon S3 event notifications are designed to be delivered at least once and to one destination only. You cannot attach two or more SNS topics or SQS queues for S3 event notification. Therefore, you must send the event notification to Amazon SNS.

Hence, the correct answer is: ***Create an Amazon SNS topic and configure two Amazon SQS queues to subscribe to the topic. Grant Amazon S3 permission to send notifications to Amazon SNS and update the bucket to use the new SNS topic.***

The option that says: ***Set up another Amazon SQS queue for the other team. Grant Amazon S3 permission to send a notification to the second SQS queue*** is incorrect because you can only add 1 SQS or SNS at a time for Amazon S3 events notification. If you need to send the events to multiple subscribers, you should implement a message fanout pattern with Amazon SNS and Amazon SQS.

The option that says: ***Create a new Amazon SNS FIFO topic for the other team. Grant Amazon S3 permission to send the notification to the second SNS topic*** is incorrect. Just as mentioned in the previous option, you can only add 1 SQS or SNS at a time for Amazon S3 events notification. In addition, neither Amazon SNS FIFO topic nor Amazon SQS FIFO queue is warranted in this scenario. Both of them can be used together to provide strict message ordering and message deduplication. The FIFO capabilities of each of these services work together to act as a fully managed service to integrate distributed applications that require data consistency in near-real-time.

The option that says: ***Set up an Amazon SNS topic and configure two Amazon SQS queues to poll the SNS topic. Grant Amazon S3 permission to send notifications to Amazon SNS and update the bucket to use the new SNS topic*** is incorrect because you can't poll Amazon SNS. Instead of configuring queues to poll Amazon SNS, you should configure each Amazon SQS queue to subscribe to the SNS topic.

References:

- <https://docs.aws.amazon.com/AmazonS3/latest/dev/ways-to-add-notification-config-to-bucket.html>
- <https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html#notification-how-to-overview>
- <https://docs.aws.amazon.com/sns/latest/dg/welcome.html>

Check out this Amazon S3 Cheat Sheet:

- <https://tutorialsdojo.com/amazon-s3/>

Amazon SNS Overview:

- <https://youtu.be/ft5R45IEUj8>

65. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A tech company that you are working for has undertaken a Total Cost Of Ownership (TCO) analysis evaluating the use of Amazon S3 versus acquiring more storage hardware. The result was that all 1200 employees would be granted access to use Amazon S3 for storage of their personal documents.

Which of the following will you need to consider so you can set up a solution that incorporates single sign-on feature from your corporate AD or LDAP directory and also restricts access for each individual user to a designated user folder in an S3 bucket? (Select TWO.)

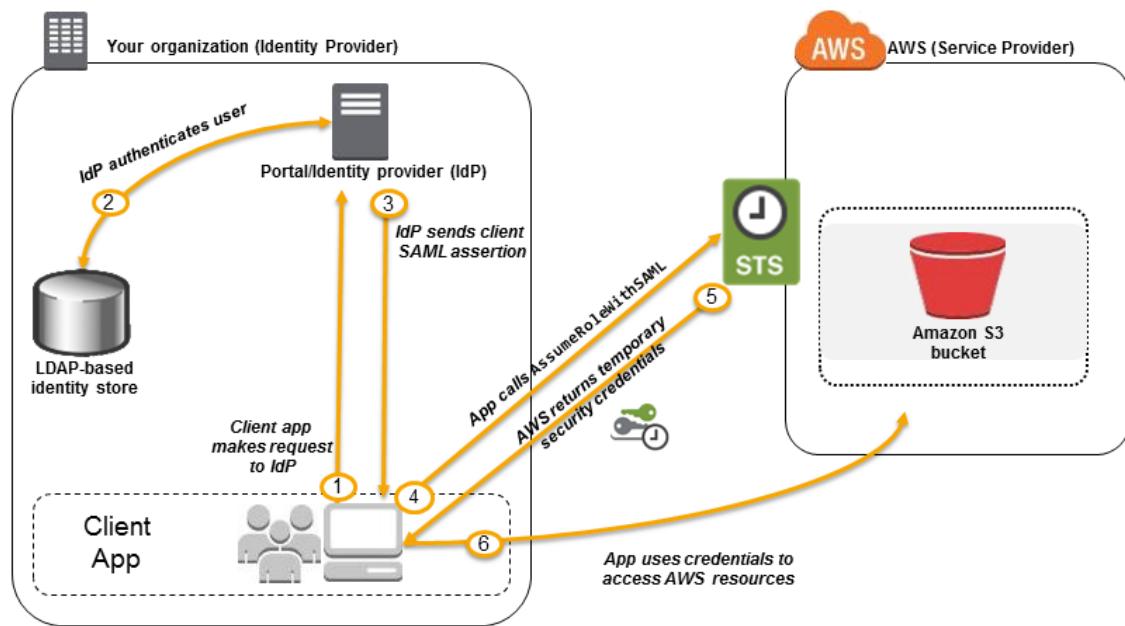
- Set up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket.
- **Configure an IAM role and an IAM Policy to access the bucket.**
- Map each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents.
- **Set up a Federation proxy or an Identity provider, and use AWS Security Token Service to generate temporary tokens.**
- Use 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others.

Incorrect

The question refers to one of the common scenarios for temporary credentials in AWS. Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles. In this example, it is called **enterprise identity federation** considering that you also need to set up a single sign-on (SSO) capability.

The correct answers are:

- *– Setup a Federation proxy or an Identity provider***
- *– Setup an AWS Security Token Service to generate temporary tokens***
- *– Configure an IAM role and an IAM Policy to access the bucket.***



In an enterprise identity federation, you can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with a separate user name and password. This is known as the *single sign-on* (SSO) approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your own solution for federating user identities.

Using 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others is incorrect since you don't have to use 3rd party solutions to provide the access. AWS already provides the necessary tools that you can use in this situation.

Mapping each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents is incorrect as there is no direct way of integrating Amazon S3 with Amazon WorkDocs for this particular scenario. **Amazon WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service.** With Amazon WorkDocs, you can easily create, edit, and share content. And because it's stored centrally on AWS, you can access it from anywhere on any device.

Setting up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket is incorrect since creating that many IAM users would be unnecessary. Also, you want the account to integrate with your AD or LDAP directory, hence, IAM Users does not fit these criteria.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc.html

<https://aws.amazon.com/blogs/security/writing-iam-policies-grant-access-to-user-specific-folders-in-an-amazon-s3-bucket/>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

1. QUESTION

Category: CSAA – Design Resilient Architectures

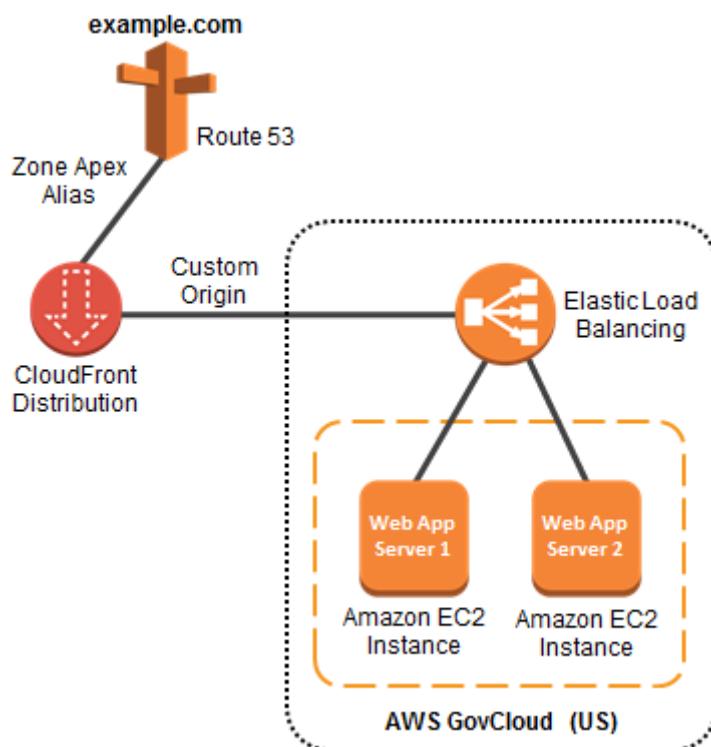
A start-up company has an EC2 instance that is hosting a web application. The volume of users is expected to grow in the coming months and hence, you need to add more elasticity and scalability in your AWS architecture to cope with the demand.

Which of the following options can satisfy the above requirement for the given scenario? (Select TWO.)

- Set up an S3 Cache in front of the EC2 instance.
- Set up two EC2 instances and use Route 53 to route traffic based on a Weighted Routing Policy.
- Set up an AWS WAF behind your EC2 Instance.
- Set up two EC2 instances deployed using Launch Templates and integrated with AWS Glue.
- Set up two EC2 instances and then put them behind an Elastic Load balancer (ELB).

Incorrect

Using an Elastic Load Balancer is an ideal solution for adding elasticity to your application. Alternatively, you can also create a policy in Route 53, such as a Weighted routing policy, to evenly distribute the traffic to 2 or more EC2 instances. Hence, ***setting up two EC2 instances and then put them behind an Elastic Load balancer (ELB)*** and ***setting up two EC2 instances and using Route 53 to route traffic based on a Weighted Routing Policy*** are the correct answers.



Setting up an S3 Cache in front of the EC2 instance is incorrect because doing so does not provide elasticity and scalability to your EC2 instances.

Setting up an AWS WAF behind your EC2 Instance is incorrect because AWS WAF is a web application firewall that helps protect your web applications from common web exploits. This service is more on providing security to your applications.

Setting up two EC2 instances deployed using Launch Templates and integrated with AWS Glue is incorrect because AWS Glue is a fully managed extract, transform, and load (ETL) service that makes it easy for customers to prepare and load their data for analytics. It does not provide scalability or elasticity to your instances.

References:

<https://aws.amazon.com/elasticloadbalancing>

[## Check out this AWS Elastic Load Balancing Cheat Sheet:](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide>Welcome.html</p></div><div data-bbox=)

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

2. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A website that consists of HTML, CSS, and other client-side Javascript will be hosted on the AWS environment. Several high-resolution images will be displayed on the webpage. The website and the photos should have the optimal loading response times as possible, and should also be able to scale to high request rates.

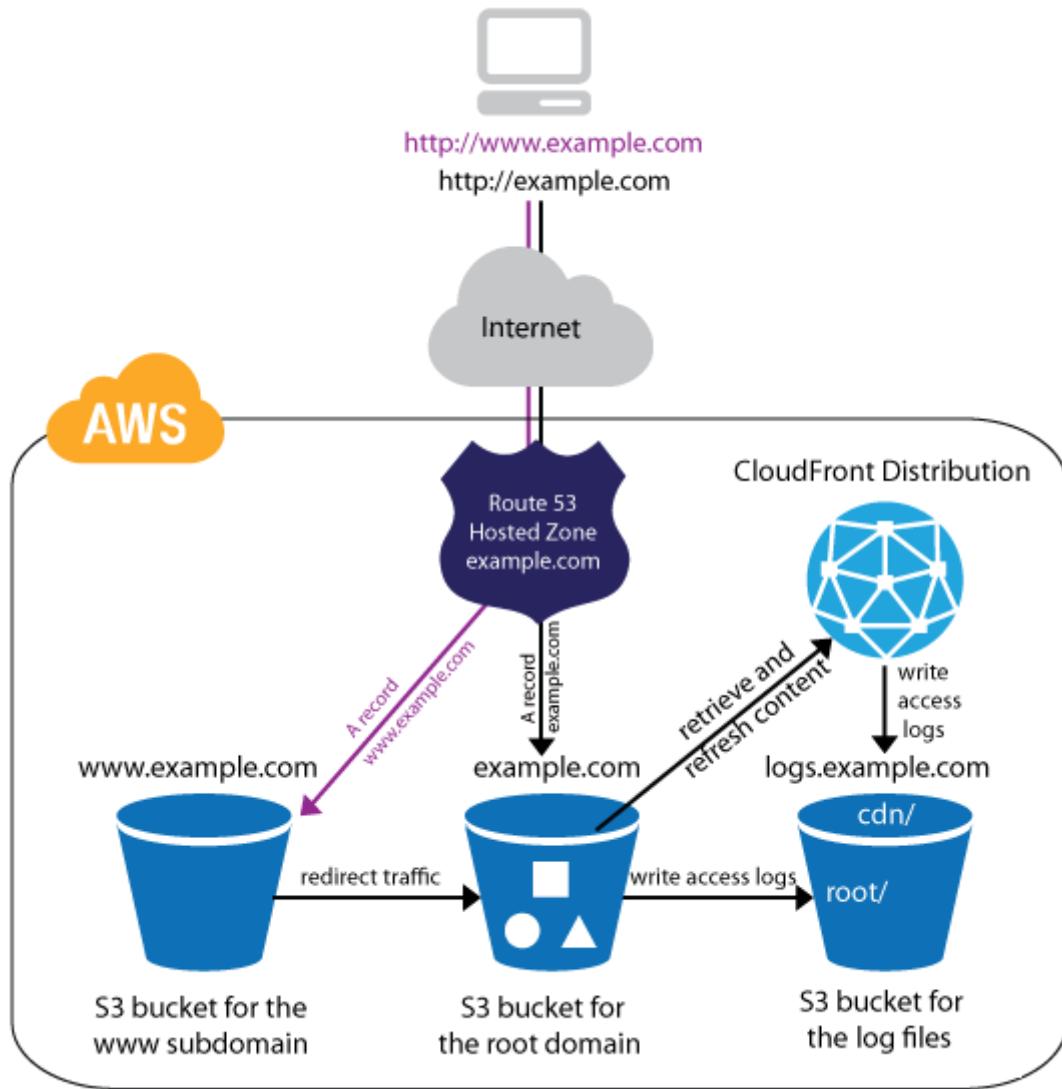
Which of the following architectures can provide the most cost-effective and fastest loading experience?

- Create a Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.
- Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator.
- Create a Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.
- **Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.**

Correct

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Additionally, You can use Amazon S3 to host a static website. On a static website, individual webpages include static content. Amazon S3 is **highly scalable and you only pay for what you use**, you can start small and grow your application as you wish, with no compromise on performance or reliability.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. CloudFront can be integrated with Amazon S3 for fast delivery of data originating from an S3 bucket to your end-users. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.



The scenario given is about storing and hosting images and a static website respectively. Since we are just dealing with static content, we can leverage the web hosting feature of S3. Then we can improve the architecture further by integrating it with CloudFront. This way, users will be able to load both the web pages and images faster than if we are serving them from a standard webserver.

Hence, the correct answer is: ***Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.***

The option that says: ***Create an Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in a S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users*** is incorrect. Creating your own web server just to host a static website in AWS is a costly solution. Web Servers on an EC2 instance is usually used for hosting dynamic web applications. Since static websites contain web pages with fixed content, we should use S3 website hosting instead.

The option that says: ***Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator*** is incorrect. This is how we serve static websites in the old days. Now, with the help of S3 website hosting, we can host our static contents from a durable, high-availability, and highly scalable environment without managing any servers. Hosting static websites in S3 is cheaper than hosting it in an EC2 instance. In addition, Using ASG for scaling instances that host a static website is an over-engineered solution that carries unnecessary costs. S3 automatically scales to high requests and you only pay for what you use.

The option that says: ***Create an Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users*** is incorrect because although LightSail is cheaper than EC2, creating your own LightSail web server for hosting static websites is still a relatively expensive solution when compared to hosting it on S3. In addition, S3 automatically scales to high request rates.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>

<https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/>

Check out these Amazon S3 and CloudFront Cheat Sheets:

<https://tutorialsdojo.com/amazon-s3/>

<https://tutorialsdojo.com/amazon-cloudfront/>

3. QUESTION

Category: CSAA – Design High-Performing Architectures

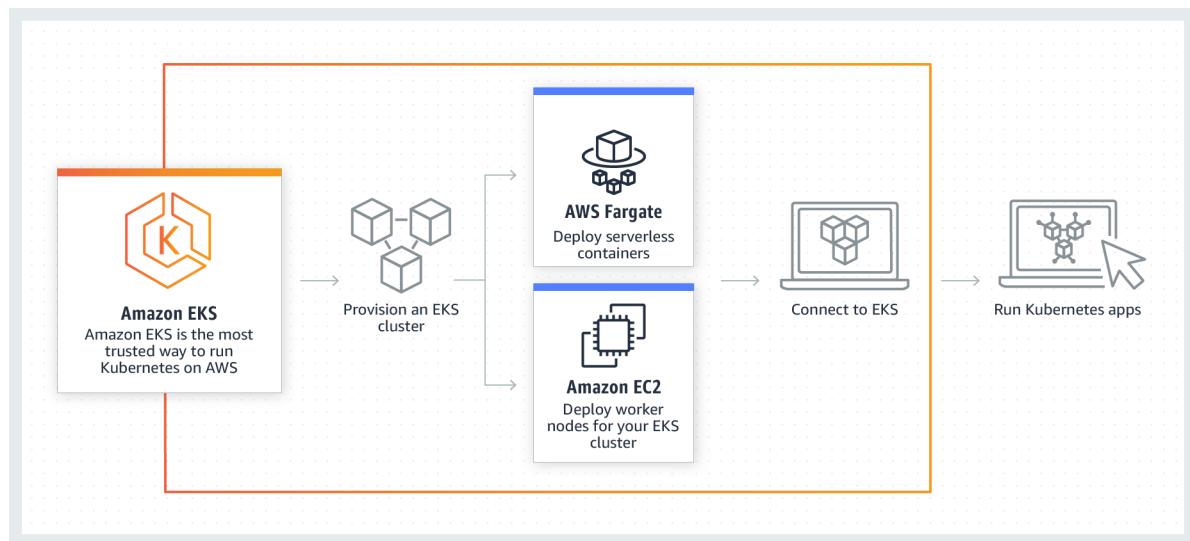
A company plans to migrate its suite of containerized applications running on-premises to a container service in AWS. The solution must be cloud-agnostic and use an open-source platform that can automatically manage containerized workloads and services. It should also use the same configuration and tools across various production environments.

What should the Solution Architect do to properly migrate and satisfy the given requirement?

- Migrate the application to Amazon Container Registry (ECR) with Amazon EC2 instance worker nodes.
- **Migrate the application to Amazon Elastic Kubernetes Service with EKS worker nodes.**
- Migrate the application to Amazon Elastic Container Service with ECS tasks that use the AWS Fargate launch type.
- Migrate the application to Amazon Elastic Container Service with ECS tasks that use the Amazon EC2 launch type.

Correct

Amazon EKS provisions and scales the Kubernetes control plane, including the API servers and backend persistence layer, across multiple AWS availability zones for high availability and fault tolerance. Amazon EKS automatically detects and replaces unhealthy control plane nodes and provides patching for the control plane. Amazon EKS is integrated with many AWS services to provide scalability and security for your applications. These services include Elastic Load Balancing for load distribution, IAM for authentication, Amazon VPC for isolation, and AWS CloudTrail for logging.



To migrate the application to a container service, you can use Amazon ECS or Amazon EKS. But the key point in this scenario is cloud-agnostic and open-source platform. Take note that Amazon ECS is an AWS proprietary container service. This means that it is not an open-source platform. Amazon EKS is a portable, extensible, and open-source platform for managing containerized workloads and services. Kubernetes is considered cloud-agnostic because it allows you to move your containers to other cloud service providers.

Amazon EKS runs up-to-date versions of the open-source Kubernetes software, so you can use all of the existing plugins and tools from the Kubernetes community. Applications running on Amazon EKS are fully compatible with applications running on any standard Kubernetes environment, whether running in on-premises data centers or public clouds. This means that you can easily migrate any standard Kubernetes application to Amazon EKS without any code modification required.

Hence, the correct answer is: ***Migrate the application to Amazon Elastic Kubernetes Service with EKS worker nodes.***

The option that says: ***Migrate the application to Amazon Container Registry (ECR) with Amazon EC2 instance worker nodes*** is incorrect because Amazon ECR is just a fully-managed Docker container registry. Also, this option is not an open-source platform that can manage containerized workloads and services.

The option that says: ***Migrate the application to Amazon Elastic Container Service with ECS tasks that use the AWS Fargate launch type*** is incorrect because it is stated in the scenario that you have to migrate the application suite to an open-source platform. AWS Fargate is just a serverless compute engine for containers. It is not cloud-agnostic since you cannot use the same configuration and tools if you moved it to another cloud service provider such as Microsoft Azure or Google Cloud Platform (GCP).

The option that says: ***Migrate the application to Amazon Elastic Container Service with ECS tasks that use the Amazon EC2 launch type.*** is incorrect because Amazon ECS is an AWS proprietary managed container orchestration service. You should use Amazon EKS since Kubernetes is an open-source platform and is considered cloud-agnostic. With Kubernetes, you can use the same configuration and tools that you're currently using in AWS even if you move your containers to another cloud service provider.

References:

<https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html>

<https://aws.amazon.com/eks/faqs/>

Check out our library of AWS Cheat Sheets:

<https://tutorialsdojo.com/links-to-all-aws-cheat-sheets/>

4. QUESTION

Category: CSAA – Design Resilient Architectures

You have built a web application that checks for new items in an S3 bucket once every hour. If new items exist, a message is added to an SQS queue. You have a fleet of EC2 instances which retrieve messages from the SQS queue, process the file, and finally, send you and the user an email confirmation that the item has been successfully processed. Your officemate uploaded one test file to the S3 bucket and after a couple of hours, you noticed that you and your officemate have 50 emails from your application with the same message.

Which of the following is most likely the root cause why the application has sent you and the user multiple emails?

- The sqsSendMessage attribute of the SQS queue is configured to 50.
- There is a bug in the application.
- By default, SQS automatically deletes the messages that were processed by the consumers. It might be possible that your officemate has submitted the request 50 times which is why you received a lot of emails.
- **Your application does not issue a delete command to the SQS queue after processing the message, which is why this message went back to the queue and was processed multiple times.**

Correct

In this scenario, the main culprit is that your application does not issue a delete command to the SQS queue after processing the message, which is why this message went back to the queue and was processed multiple times.

The option that says: ***The sqsSendMessage attribute of the SQS queue is configured to 50*** is incorrect as there is no sqsSendMessage attribute in SQS.

The option that says: ***There is a bug in the application*** is a valid answer but since the scenario did not mention that the EC2 instances deleted the processed messages, the most likely cause of the problem is that the application does not issue a delete command to the SQS queue as mentioned above.

The option that says: ***By default, SQS automatically deletes the messages that were processed by the consumers. It might be possible that your officemate has submitted the request 50 times which is why you received a lot of emails*** is incorrect as SQS does not automatically delete the messages.

Reference:

<https://aws.amazon.com/sqs/faqs/>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

5. QUESTION

Category: CSAA – Design Resilient Architectures

A company is planning to launch an application which requires a data warehouse that will be used for their infrequently accessed data. You need to use an EBS Volume that can handle large, sequential I/O operations.

Which of the following is the most cost-effective storage type that you should use to meet the requirement?

- EBS General Purpose SSD (gp2)
- Provisioned IOPS SSD (io1)
- Throughput Optimized HDD (st1)
- **Cold HDD (sc1)**

Correct

Cold HDD volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than Throughput Optimized HDD, this is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, Cold HDD provides inexpensive block storage. Take note that bootable Cold HDD volumes are not supported.

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none">• Recommended for most workloads• System boot volumes• Virtual desktops• Low-latency interactive apps• Development and test environments	<ul style="list-style-type: none">• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume• Large database workloads, such as:<ul style="list-style-type: none">◦ MongoDB◦ Cassandra◦ Microsoft SQL Server◦ MySQL◦ PostgreSQL◦ Oracle	<ul style="list-style-type: none">• Streaming workloads requiring consistent, fast throughput at a low price• Big data• Data warehouses• Log processing• Cannot be a boot volume	<ul style="list-style-type: none">• Throughput-oriented storage for large volumes of data that is infrequently accessed• Scenarios where the lowest storage cost is important• Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS*/Volume	16,000***	64,000***	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

Cold HDD provides the lowest cost HDD volume and is designed for less frequently accessed workloads. Hence, ***Cold HDD (sc1)*** is the correct answer.

In the exam, always consider the difference between SSD and HDD as shown on the table below. This will allow you to easily eliminate specific EBS-types in the options which are not SSD or not HDD, depending on whether the question asks for a storage type which has ***small, random*** I/O operations or ***large, sequential*** I/O operations.

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	small, random I/O operations	large, sequential I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	<ul style="list-style-type: none"> - Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others... 	<ul style="list-style-type: none"> - Best for large streaming workloads requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is infrequently accessed
Cost	moderate / high 	low 
Dominant Performance Attribute	IOPS	Throughput (MiB/s)



TutorialsDojo

EBS General Purpose SSD (gp2) is incorrect because a General purpose SSD volume costs more and it is mainly used for a wide variety of workloads. It is recommended to be used as system boot volumes, virtual desktops, low-latency interactive apps, and many more.

Provisioned IOPS SSD (io1) is incorrect because this costs more than Cold HDD and thus, not cost-effective for this scenario. It provides the highest performance SSD volume for mission-critical low-latency or high-throughput workloads, which is not needed in the scenario.

Throughput Optimized HDD (st1) is incorrect because this is primarily used for **frequently** accessed, throughput-intensive workloads. In this scenario, Cold HDD perfectly fits the requirement as it is used for their infrequently accessed data and provides the lowest cost, unlike Throughput Optimized HDD.

References:

<https://aws.amazon.com/ebs/details/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

6. QUESTION

Category: CSAA – Design Resilient Architectures

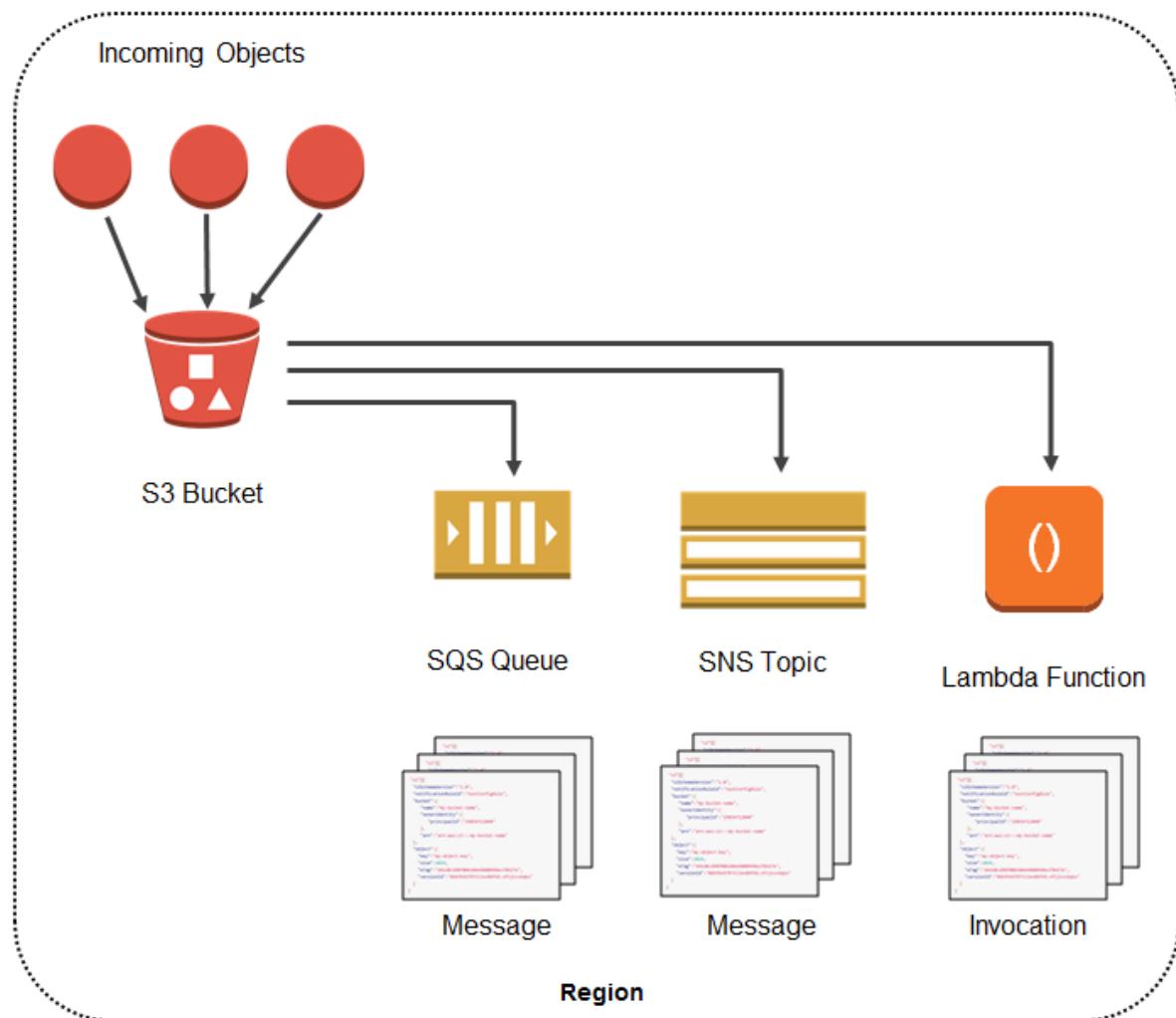
A Data Engineer is working for a litigation firm for their case history application. The engineer needs to keep track of all the cases that the firm has handled. The static assets like .jpg, .png, and .pdf files are stored in S3 for cost efficiency and high durability. As these files are critical to the business, the engineer wants to keep track of what's happening in the S3 bucket. The engineer found out that S3 has an event notification whenever a delete or write operation happens within the S3 bucket.

What are the possible Event Notification destinations available for S3 buckets? (Select TWO.)

- Kinesis
- SES
- **SQS**
- **Lambda function**
- SWF

Correct

The **Amazon S3 notification** feature enables you to receive notifications when certain events happen in your bucket. To enable notifications, you must first add a notification configuration identifying the events you want Amazon S3 to publish, and the destinations where you want Amazon S3 to send the event notifications.



Amazon S3 supports the following destinations where it can publish events:

Amazon Simple Notification Service (Amazon SNS) topic – A web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients.

Amazon Simple Queue Service (Amazon SQS) queue – Offers reliable and scalable hosted queues for storing messages as they travel between computer.

AWS Lambda – AWS Lambda is a compute service where you can upload your code and the service can run the code on your behalf using the AWS infrastructure. You package up and upload your custom code to AWS Lambda when you create a Lambda function

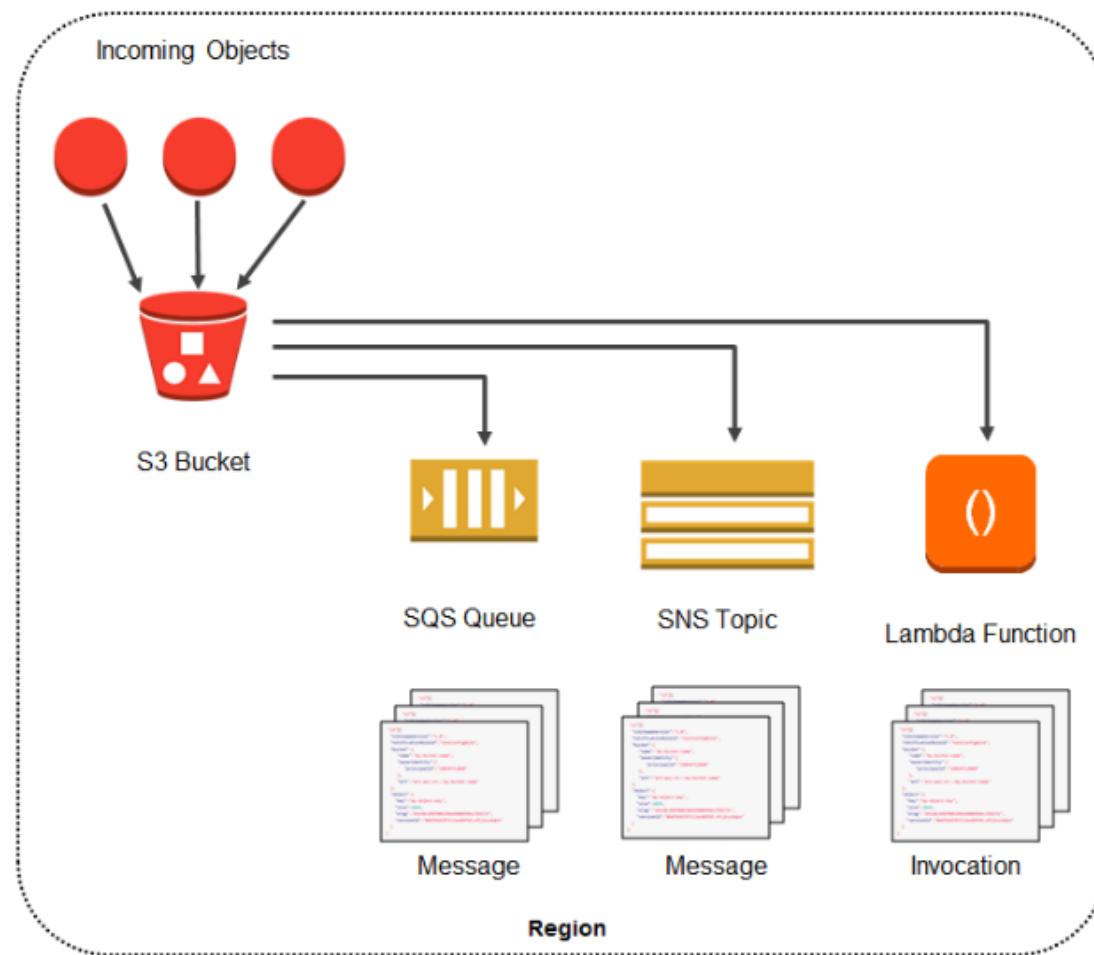
Kinesis is incorrect because this is used to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information, and not used for event notifications. You have to use SNS, SQS or Lambda.

SES is incorrect because this is mainly used for sending emails designed to help digital marketers and application developers send marketing, notification, and transactional emails, and not for sending event notifications from S3. You have to use SNS, SQS or Lambda.

SWF is incorrect because this is mainly used to build applications that use Amazon's cloud to coordinate work across distributed components and not used as a way to trigger event notifications from S3. You have to use SNS, SQS or Lambda.

Here's what you need to do in order to start using this new feature with your application:

1. Create the queue, topic, or Lambda function (which I'll call the target for brevity) if necessary.
2. Grant S3 permission to publish to the target or invoke the Lambda function. For SNS or SQS, you do this by applying an appropriate policy to the topic or the queue. For Lambda, you must create and supply an IAM role, then associate it with the Lambda function.
3. Arrange for your application to be invoked in response to activity on the target. As you will see in a moment, you have several options here.
4. Set the bucket's Notification Configuration to point to the target.



Reference:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/NotificationHowTo.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

7. QUESTION

Category: CSAA – Design Resilient Architectures

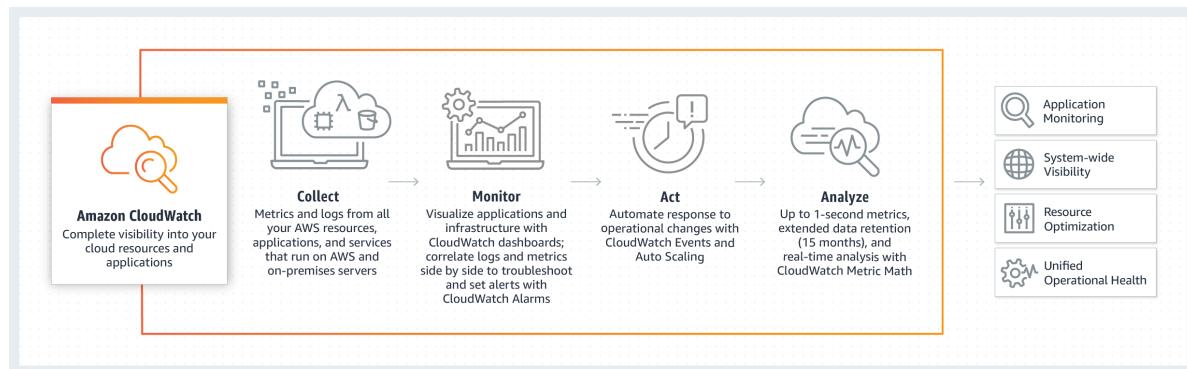
A company is running a custom application in an Auto Scaling group of Amazon EC2 instances. Several instances are failing due to insufficient swap space. The Solutions Architect has been instructed to troubleshoot the issue and effectively monitor the available swap space of each EC2 instance.

Which of the following options fulfills this requirement?

- Create a new trail in AWS CloudTrail and configure Amazon CloudWatch Logs to monitor your trail logs.
- Create a CloudWatch dashboard and monitor the `Swapused` metric.
- Enable detailed monitoring on each instance and monitor the `Swaputilization` metric.
- **Install the CloudWatch agent on each instance and monitor the `Swaputilization` metric.**

Correct

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, and set alarms. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health.



The main requirement in the scenario is to monitor the `Swaputilization` metric. Take note that you can't use the default metrics of CloudWatch to monitor the `Swaputilization` metric. To monitor custom metrics, you must install the CloudWatch agent on the EC2 instance. After installing the CloudWatch agent, you can now collect system metrics and log files of an EC2 instance.

Hence, the correct answer is: ***Install the CloudWatch agent on each instance and monitor the `SwapUtilization` metric.***

The option that says: ***Enable detailed monitoring on each instance and monitor the `Swaputilization` metric**** is incorrect because you can't monitor the `Swaputilization` metric by just enabling the detailed monitoring option. You must install the CloudWatch agent on the instance.

The option that says: ***Create a CloudWatch dashboard and monitor the `Swapused` metric**** is incorrect because you must install the CloudWatch agent first to add the custom metric in the dashboard.

The option that says: ***Create a new trail in AWS CloudTrail and configure Amazon CloudWatch Logs to monitor your trail logs*** is incorrect because CloudTrail won't help you monitor custom metrics. CloudTrail is specifically used for monitoring API activities in an AWS account.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html>

<https://aws.amazon.com/cloudwatch/faqs/>

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

Amazon CloudWatch Overview:

<https://youtu.be/q0DmxfyGkeU>

8. QUESTION

Category: CSAA – Design High-Performing Architectures

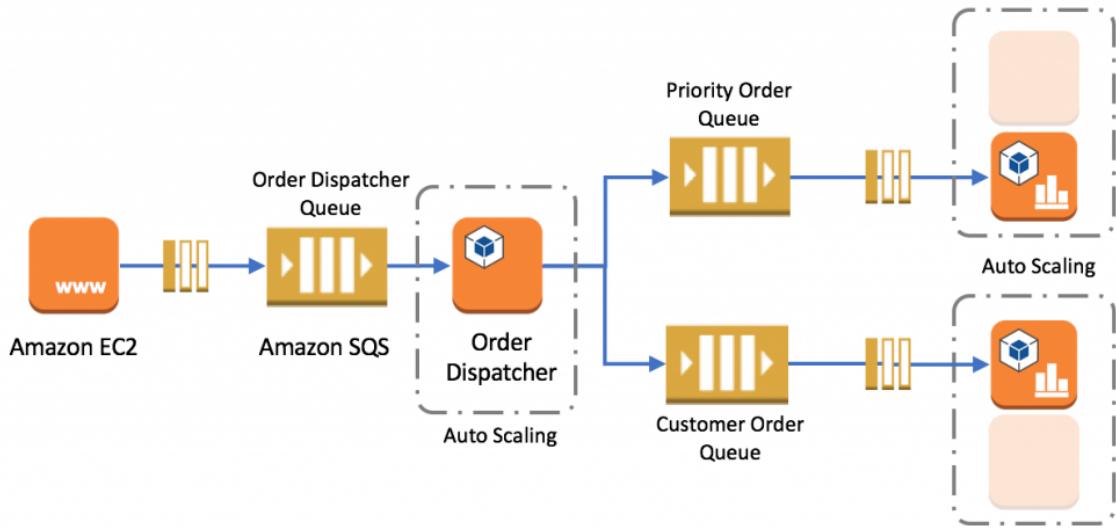
A company launched a website that accepts high-quality photos and turns them into a downloadable video montage. The website offers a free and a premium account that guarantees faster processing. All requests by both free and premium members go through a single SQS queue and then processed by a group of EC2 instances that generate the videos. The company needs to ensure that the premium users who paid for the service have higher priority than the free members.

How should the company re-design its architecture to address this requirement?

- Use Amazon Kinesis to process the photos and generate the video montage in real-time.
- For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members.
- Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue.
- Use Amazon S3 to store and process the photos and then generate the video montage afterward.

Incorrect

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available.



In this scenario, it is best to create 2 separate SQS queues for each type of members. The SQS queues for the premium members can be polled first by the EC2 Instances and once completed, the messages from the free members can be processed next.

Hence, the correct answer is: ***Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue.***

The option that says: ***For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members*** is incorrect as **you cannot set a priority to individual items in the SQS queue.**

The option that says: ***Using Amazon Kinesis to process the photos and generate the video montage in real time*** is incorrect as Amazon Kinesis is used to process streaming data and it is not applicable in this scenario.

The option that says: ***Using Amazon S3 to store and process the photos and then generating the video montage afterwards*** is incorrect as Amazon S3 is used for durable storage and not for processing data.

Reference:

<https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-best-practices.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

9. QUESTION

Category: CSAA – Design High-Performing Architectures

A company has developed public APIs hosted in Amazon EC2 instances behind an Elastic Load Balancer. The APIs will be used by various clients from their respective on-premises data centers. A Solutions Architect received a report that the web service clients can only access trusted IP addresses whitelisted on their firewalls.

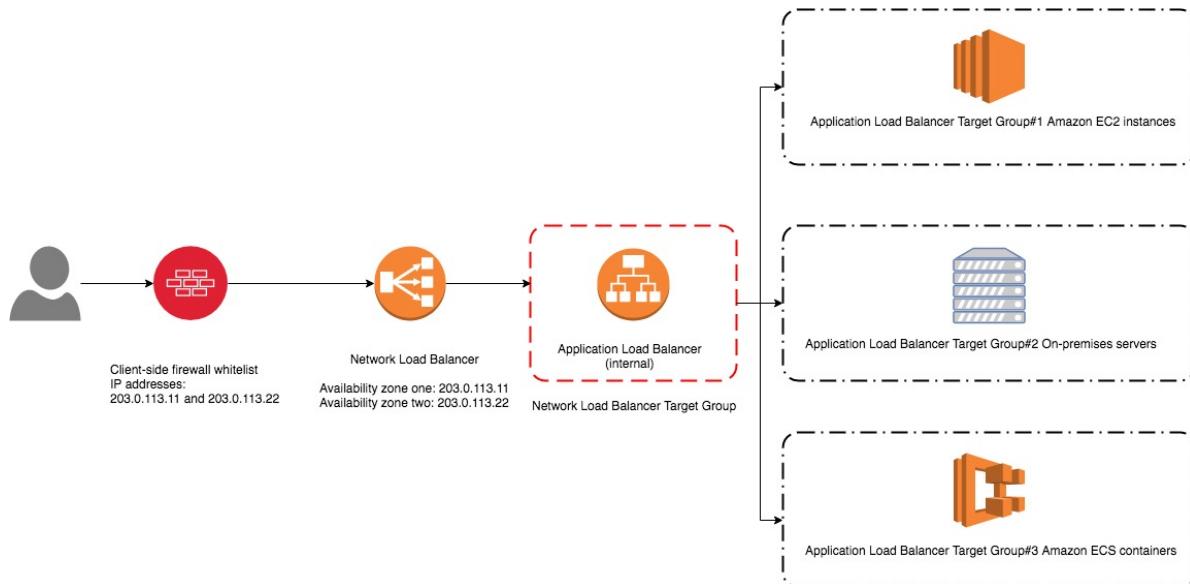
What should you do to accomplish the above requirement?

- Create an Alias Record in Route 53 which maps to the DNS name of the load balancer.

- Create a CloudFront distribution whose origin points to the private IP addresses of your web servers.
- Associate an Elastic IP address to an Application Load Balancer.
- **Associate an Elastic IP address to a Network Load Balancer.**

Incorrect

A **Network Load Balancer** functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the default rule's target group. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.



Based on the given scenario, web service clients can only access trusted IP addresses. To resolve this requirement, you can use the Bring Your Own IP (BYOIP) feature to use the trusted IPs as Elastic IP addresses (EIP) to a Network Load Balancer (NLB). This way, there's no need to re-establish the whitelists with new IP addresses.

Hence, the correct answer is: ***Associate an Elastic IP address to a Network Load Balancer.***

The option that says: ***Associate an Elastic IP address to an Application Load Balancer*** is incorrect because you can't assign an Elastic IP address to an Application Load Balancer. The alternative method you can do is assign an Elastic IP address to a Network Load Balancer in front of the Application Load Balancer.

The option that says: ***Create a CloudFront distribution whose origin points to the private IP addresses of your web servers*** is incorrect because web service clients can only access trusted IP addresses. The fastest way to resolve this requirement is to attach an Elastic IP address to a Network Load Balancer.

The option that says: ***Create an Alias Record in Route 53 which maps to the DNS name of the load balancer*** is incorrect. This approach won't still allow them to access the application because of trusted IP addresses on their firewalls.

References:

- <https://aws.amazon.com/premiumsupport/knowledge-center/elb-attach-elastic-ip-to-public-nlb/>
- <https://aws.amazon.com/blogs/networking-and-content-delivery/using-static-ip-addresses-for-application-load-balancers/>
- <https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html>

Check out this AWS Elastic Load Balancing Cheat Sheet:

10. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A solutions architect is designing a cost-efficient, highly available storage solution for company data. One of the requirements is to ensure that the previous state of a file is preserved and retrievable if a modified version of it is uploaded. Also, to meet regulatory compliance, data over 3 years must be retained in an archive and will only be accessible once a year.

How should the solutions architect build the solution?

- Create an S3 Standard bucket and enable S3 Object Lock in governance mode.
- Create a One-Zone-IA bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.
- **Create an S3 Standard bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.**
- Create an S3 Standard bucket with S3 Object Lock in compliance mode enabled then configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.

Correct

Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket. You can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in your buckets. With versioning, you can recover more easily from both unintended user actions and application failures. After versioning is enabled for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of those objects.

Hence, the correct answer is: ***Create an S3 Standard bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.***

The S3 Object Lock feature allows you to store objects using a write-once-read-many (WORM) model. In the scenario, changes to objects are allowed but their previous versions should be preserved and remain retrievable. If you enable the S3 Object Lock feature, you won't be able to upload new versions of an object. This feature is only helpful when you want to prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely.

Therefore, the following options are incorrect:

- *- Create an S3 Standard bucket and enable S3 Object Lock in governance mode.***
- *- Create an S3 Standard bucket with S3 Object Lock in compliance mode enabled then configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years.***

The option that says: ***Create a One-Zone-IA bucket with object-level versioning enabled and configure a lifecycle rule that transfers files to Amazon S3 Glacier Deep Archive after 3 years*** is incorrect. One-Zone-IA is not highly available as it only relies on one availability zone for storing data.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html>

<https://aws.amazon.com/blogs/aws/new-amazon-s3-storage-class-glacier-deep-archive/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

11. QUESTION

Category: CSAA – Design Secure Applications and Architectures

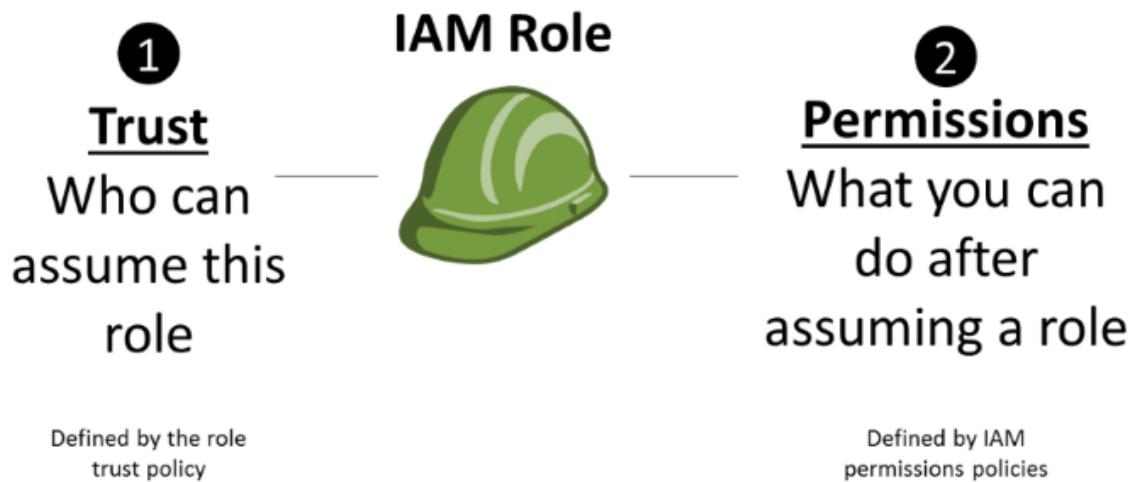
A company developed a meal planning application that provides meal recommendations for the week as well as the food consumption of the users. The application resides on an EC2 instance which requires access to various AWS services for its day-to-day operations.

Which of the following is the best way to allow the EC2 instance to access the S3 bucket and other AWS services?

- Store the API credentials in the EC2 instance.
- Add the API Credentials in the Security Group and assign it to the EC2 instance.
- Store the API credentials in a bastion host.
- **Create a role in IAM and assign it to the EC2 instance.**

Correct

The best practice in handling API Credentials is to create a new role in the Identity Access Management (IAM) service and then assign it to a specific EC2 instance. In this way, you have a secure and centralized way of storing and managing your credentials.



Storing the API credentials in the EC2 instance, **adding the API Credentials in the Security Group and assigning it to the EC2 instance**, and **storing the API credentials in a bastion host** are incorrect because it is not secure to store nor use the API credentials from an EC2 instance. You should use IAM service instead.

Reference:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>

Check out this AWS IAM Cheat Sheet:

12. QUESTION

Category: CSAA – Design Resilient Architectures

A Solutions Architect is working for a company that uses Chef Configuration management in their data center. She needs to leverage their existing Chef recipes in AWS.

Which of the following services should she use?

- AWS CloudFormation
- **AWS OpsWorks**
- AWS Elastic Beanstalk
- Amazon Simple Workflow Service

Correct

AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet. Chef and Puppet are automation platforms that allow you to use code to automate the configurations of your servers. OpsWorks lets you use Chef and Puppet to automate how servers are configured, deployed, and managed across your Amazon EC2 instances or on-premises compute environments. OpsWorks has three offerings – AWS Opsworks for Chef Automate, AWS OpsWorks for Puppet Enterprise, and AWS OpsWorks Stacks.

Welcome to OpsWorks for Chef Automate

OpsWorks for Chef Automate helps you automate, provision, and configure your environment. The Chef Automate platform delivers DevOps workflow, automated compliance, and end-to-end pipeline visibility.

A Chef Automate server manages nodes in your environment, stores information about those nodes, and serves as a central repository for your Chef cookbooks. [Learn more](#).

[Create Chef Automate server](#)

OpsWorks for Chef Automate benefits



Easy to launch
Create Chef Automate servers in a few simple steps. You can customize instance size, security, maintenance, and more.



Automated infrastructure
Automate your infrastructure, compliance, and applications to create and deploy AWS resources easily.



Zero maintenance
Manage your infrastructure and apps along with automated backups, upgrades, and restorations.

Amazon Simple Workflow Service is incorrect because AWS SWF is a fully-managed state tracker and task coordinator in the Cloud. It does not let you leverage Chef recipes.

AWS Elastic Beanstalk is incorrect because this handles an application's deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. It does not let you leverage Chef recipes just like Amazon SWF.

AWS CloudFormation is incorrect because this is a service that lets you create a collection of related AWS resources and provision them in a predictable fashion using infrastructure as code. It does not let you leverage Chef recipes just like Amazon SWF and AWS Elastic Beanstalk.

Reference:

<https://aws.amazon.com/opsworks/>

Check out this AWS OpsWorks Cheat Sheet:

<https://tutorialsdojo.com/aws-opsworks/>

Elastic Beanstalk vs CloudFormation vs OpsWorks vs CodeDeploy:

<https://tutorialsdojo.com/elastic-beanstalk-vs-cloudformation-vs-opsworks-vs-codedeploy/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

13. QUESTION

Category: CSAA – Design Resilient Architectures

A company has multiple VPCs with IPv6 enabled for its suite of web applications. The Solutions Architect tried to deploy a new Amazon EC2 instance but she received an error saying that there is no IP address available on the subnet.

How should the Solutions Architect resolve this problem?

- Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance.
- Disable the IPv4 support in the VPC and use the available IPv6 addresses.
- Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC.
- Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance.

Incorrect

Amazon Virtual Private Cloud (VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. You can also optionally assign an IPv6 CIDR block to your VPC, and assign IPv6 CIDR blocks to your subnets.

The screenshot shows the AWS VPC console for a VPC named 'vpc-f2bf5897 / Default VPC'. In the 'Details' tab, there is a yellow callout bubble labeled 'IPv6 enabled' pointing to the 'IPv6 pool' section, which contains 'Amazon' and 'Associated'. A purple callout bubble labeled 'IPv4 CIDRs. (Required)' points to the 'IPv4 CIDRs' section, which contains '172.31.0.0/16' and 'Associated'. An orange callout bubble labeled 'IPv6 CIDRs (Optional)' points to the 'IPv6 CIDRs' section, which contains '2600:1f18:15b3:b00::/56 (us-east-1)', 'Amazon', and 'Associated'.

If you have an existing VPC that supports IPv4 only and resources in your subnet that are configured to use IPv4 only, you can enable IPv6 support for your VPC and resources. Your VPC can operate in dual-stack mode — your resources can communicate over IPv4, or IPv6, or both. IPv4 and IPv6 communication are independent of each other. You cannot disable IPv4 support for your VPC and subnets since this is the default IP addressing system for Amazon VPC and Amazon EC2.

By default, a new EC2 instance uses an IPv4 addressing protocol. To fix the problem in the scenario, you need to create a new IPv4 subnet and deploy the EC2 instance in the new subnet.

Hence, the correct answer is: ***Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance.***

The option that says: ***Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance*** is incorrect because you need to add IPv4 subnet first before you can create an IPv6 subnet.

The option that says: ***Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC*** is incorrect because you can't have a VPC with IPv6 CIDRs only. The default IP addressing system in VPC is IPv4. You can only change your VPC to dual-stack mode where your resources can communicate over IPv4, or IPv6, or both, but not exclusively with IPv6 only.

The option that says: ***Disable the IPv4 support in the VPC and use the available IPv6 addresses*** is incorrect because you cannot disable the IPv4 support for your VPC and subnets since this is the default IP addressing system.

References:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-migrate-ipv6.html>

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html>

<https://aws.amazon.com/vpc/faqs/>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

14. QUESTION

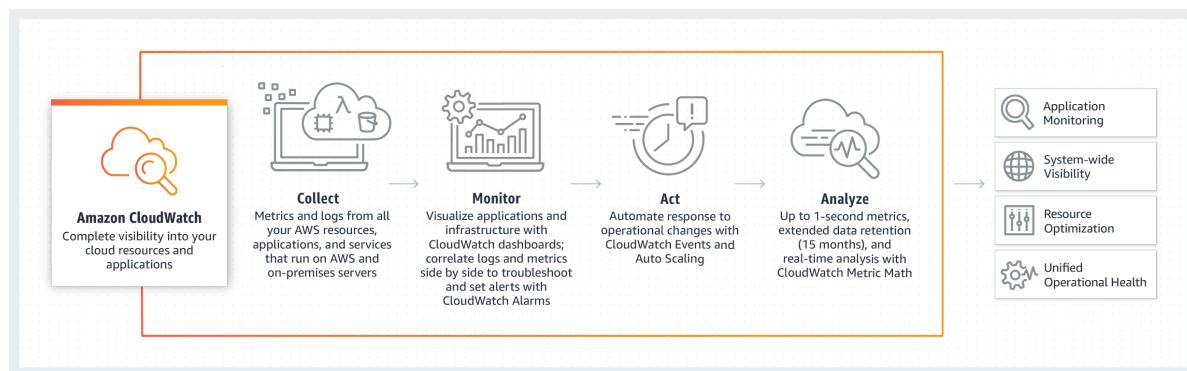
Category: CSAA – Design Resilient Architectures

A company has a top priority requirement to monitor a few database metrics and then afterward, send email notifications to the Operations team in case there is an issue. Which AWS services can accomplish this requirement? (Select TWO.)

- Amazon Simple Queue Service (SQS)
- **Amazon CloudWatch**
- **Amazon Simple Notification Service (SNS)**
- Amazon EC2 Instance with a running Berkeley Internet Name Domain (BIND) Server.
- Amazon Simple Email Service

Incorrect

Amazon CloudWatch and ***Amazon Simple Notification Service (SNS)* are correct. In this requirement, you can use Amazon CloudWatch to monitor the database and then Amazon SNS to send the emails to the Operations team. Take note that **you should use SNS instead of SES (Simple Email Service) when you want to monitor your EC2 instances.****



CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS, and on-premises servers.

SNS is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications.

Amazon Simple Email Service is incorrect. SES is a cloud-based email sending service designed to send notification and transactional emails.

Amazon Simple Queue Service (SQS) is incorrect. SQS is a fully-managed message queuing service. It does not monitor applications nor send email notifications unlike SES.

Amazon EC2 Instance with a running Berkeley Internet Name Domain (BIND) Server is incorrect because BIND is primarily used as a Domain Name System (DNS) web service. This is only applicable if you have a private hosted zone in your AWS account. It does not monitor applications nor send email notifications.

References:

<https://aws.amazon.com/cloudwatch/>

<https://aws.amazon.com/sns/>

Check out this Amazon CloudWatch Cheat Sheet:

15. QUESTION

Category: CSAA – Design High-Performing Architectures

A company is running a multi-tier web application farm in a virtual private cloud (VPC) that is not connected to their corporate network. They are connecting to the VPC over the Internet to manage the fleet of Amazon EC2 instances running in both the public and private subnets. The Solutions Architect has added a bastion host with Microsoft Remote Desktop Protocol (RDP) access to the application instance security groups, but the company wants to further limit administrative access to all of the instances in the VPC.

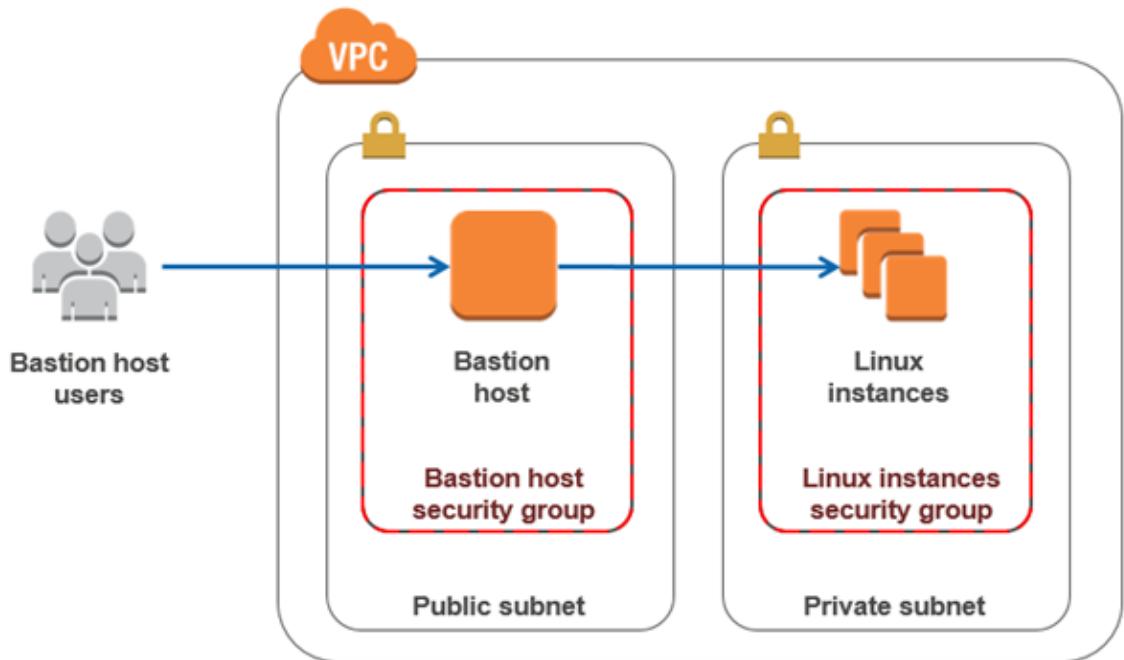
Which of the following bastion host deployment options will meet this requirement?

- Deploy a Windows Bastion host on the corporate network that has RDP access to all EC2 instances in the VPC.
- Deploy a Windows Bastion host with an Elastic IP address in the private subnet, and restrict RDP access to the bastion from only the corporate public IP addresses.
- Deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow SSH access to the bastion from anywhere.
- Deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow RDP access to bastion only from the corporate IP addresses.

Correct

The correct answer is to deploy a Windows Bastion host with an Elastic IP address in the public subnet and allow RDP access to bastion only from the corporate IP addresses.

A bastion host is a special purpose computer on a network specifically designed and configured to withstand attacks. If you have a bastion host in AWS, it is basically just an EC2 instance. It should be in a public subnet with either a public or Elastic IP address with sufficient RDP or SSH access defined in the security group. Users log on to the bastion host via SSH or RDP and then use that session to manage other hosts in the private subnets.



Deploying a Windows Bastion host on the corporate network that has RDP access to all EC2 instances in the VPC is incorrect since you do not deploy the Bastion host to your corporate network. It should be in the public subnet of a VPC.

Deploying a Windows Bastion host with an Elastic IP address in the private subnet, and restricting RDP access to the bastion from only the corporate public IP addresses is incorrect since it should be deployed in a public subnet, not a private subnet.

Deploying a Windows Bastion host with an Elastic IP address in the public subnet and allowing SSH access to the bastion from anywhere is incorrect. Since it is a Windows bastion, you should allow RDP access and not SSH as this is mainly used for Linux-based systems.

Reference:

<https://docs.aws.amazon.com/quickstart/latest/linux-bastion/architecture.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

16. QUESTION

Category: CSAA – Design Resilient Architectures

A company developed a web application and deployed it on a fleet of EC2 instances that uses Amazon SQS. The requests are saved as messages in the SQS queue, which is configured with the maximum message retention period. However, after thirteen days of operation, the web application suddenly crashed and there are 10,000 unprocessed messages that are still waiting in the queue. Since they developed the application, they can easily resolve the issue but they need to send a communication to the users on the issue.

What information should they provide and what will happen to the unprocessed messages?

- Tell the users that the application will be operational shortly, however, requests sent over three days ago will need to be resubmitted.
- **Tell the users that the application will be operational shortly and all received requests will be processed after the web application is restarted.**
- Tell the users that unfortunately, they have to resubmit all of the requests since the queue would not be able to process the 10,000 messages together.
- Tell the users that unfortunately, they have to resubmit all the requests again.

Correct

In **Amazon SQS**, you can configure the message retention period to a value from 1 minute to 14 days. The default is 4 days. Once the message retention limit is reached, your messages are automatically deleted.

A single Amazon SQS message queue can contain an unlimited number of messages. However, there is a 120,000 limit for the number of inflight messages for a standard queue and 20,000 for a FIFO queue.

Messages are inflight after they have been received from the queue by a consuming component, but have not yet been deleted from the queue.

In this scenario, it is stated that the SQS queue is configured with the maximum message retention period. The maximum message retention in SQS is 14 days that is why the option that says: ***Tell the users that the application will be operational shortly and all received requests will be processed after the web application is restarted*** is the correct answer i.e. there will be no missing messages.

The options that say: ***Tell the users that unfortunately, they have to resubmit all the requests again*** and ***Tell the users that the application will be operational shortly, however, requests sent over three days ago will need to be resubmitted*** are incorrect as there are no missing messages in the queue thus, there is no need to resubmit any previous requests.

The option that says: ***Tell the users that unfortunately, they have to resubmit all of the requests since the queue would not be able to process the 10,000 messages together*** is incorrect as the queue can contain an unlimited number of messages, not just 10,000 messages.

Reference:

<https://aws.amazon.com/sqs/>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

17. QUESTION

Category: CSAA – Design High-Performing Architectures

A company is receiving semi-structured and structured data from different sources every day. The Solutions Architect plans to use big data processing frameworks to analyze vast amounts of data and access it using various business intelligence tools and standard SQL queries.

Which of the following provides the MOST high-performing solution that fulfills this requirement?

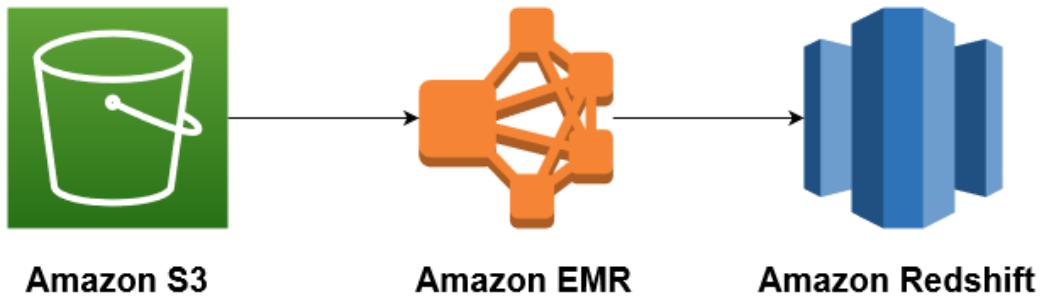
- **Create an Amazon EMR cluster and store the processed data in Amazon Redshift.**
- Create an Amazon EC2 instance and store the processed data in Amazon EBS.
- Use AWS Glue and store the processed data in Amazon S3.

- Use Amazon Kinesis Data Analytics and store the processed data in Amazon DynamoDB.

Correct

Amazon EMR is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. By using these frameworks and related open-source projects, such as Apache Hive and Apache Pig, you can process data for analytics purposes and business intelligence workloads. Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases.

Amazon Redshift is the most widely used cloud data warehouse. It makes it fast, simple and cost-effective to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It allows you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data, using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution.



The key phrases in the scenario are “big data processing frameworks” and “various business intelligence tools and standard SQL queries” to analyze the data. To leverage big data processing frameworks, you need to use Amazon EMR. The cluster will perform data transformations (ETL) and load the processed data into Amazon Redshift for analytic and business intelligence applications.

Hence, the correct answer is: ***Create an Amazon EMR cluster and store the processed data in Amazon Redshift.***

The option that says: ***Use AWS Glue and store the processed data in Amazon S3*** is incorrect because AWS Glue is just a serverless ETL service that crawls your data, builds a data catalog, performs data preparation, data transformation, and data ingestion. It won't allow you to utilize different big data frameworks effectively, unlike Amazon EMR. In addition, the S3 Select feature in Amazon S3 can only run simple SQL queries against a subset of data from a specific S3 object. To perform queries in the S3 bucket, you need to use Amazon Athena.

The option that says: ***Use Amazon Kinesis Data Analytics and store the processed data in Amazon DynamoDB*** is incorrect because Amazon DynamoDB doesn't fully support the use of standard SQL and Business Intelligence (BI) tools, unlike Amazon Redshift. It also doesn't allow you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data.

The option that says: ***Create an Amazon EC2 instance and store the processed data in Amazon EBS*** is incorrect because a single EBS-backed EC2 instance is quite limited in its computing capability. Moreover, it also entails an administrative overhead since you have to manually install and maintain the big data frameworks for the EC2 instance yourself. The most suitable solution to leverage big data frameworks is to use EMR clusters.

References:

<https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html>

<https://docs.aws.amazon.com/redshift/latest/dg/loading-data-from-emr.html>

Check out this Amazon EMR Cheat Sheet:

<https://tutorialsdojo.com/amazon-emr/>

18. QUESTION

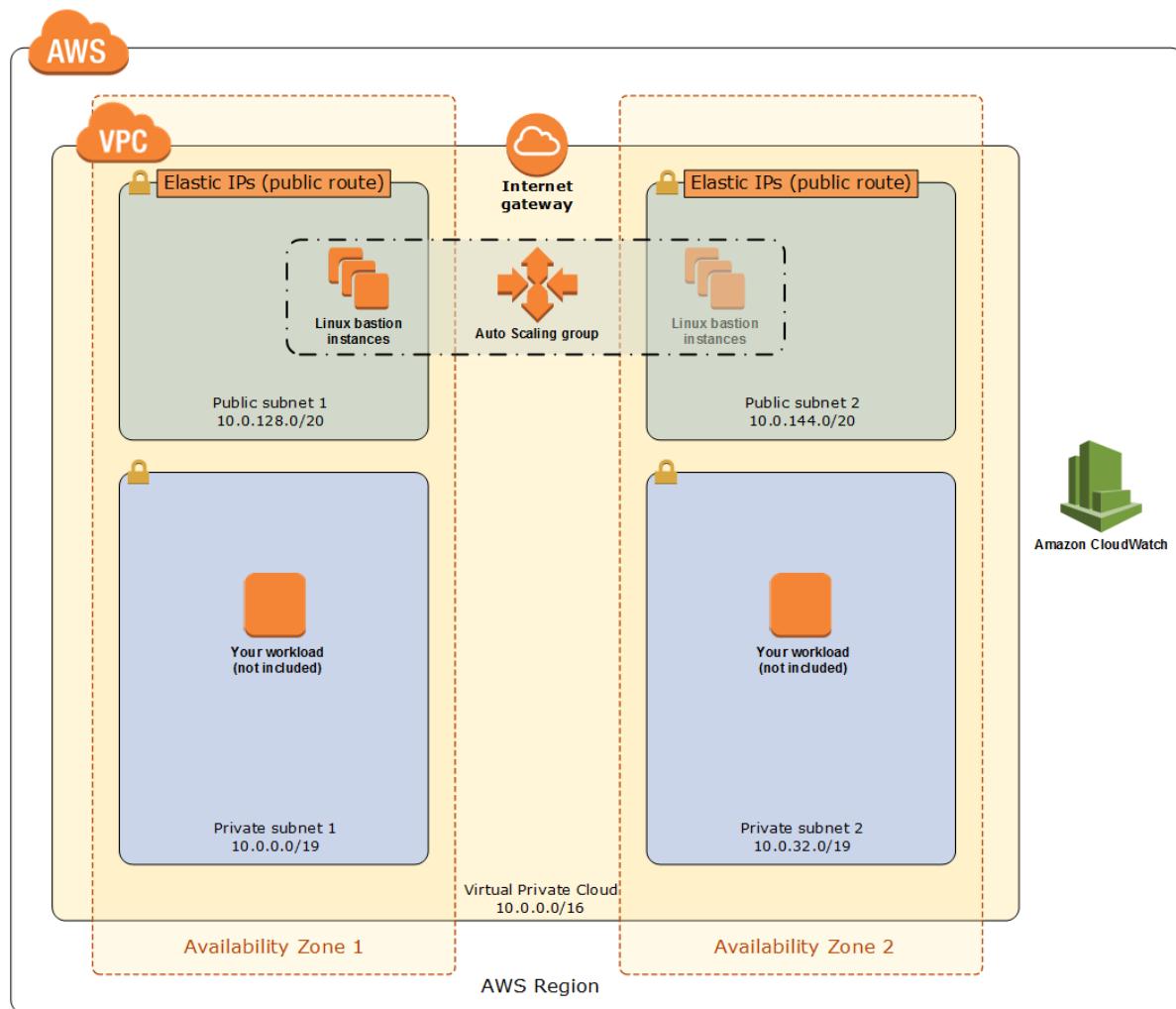
Category: CSAA – Design Resilient Architectures

A Solutions Architect needs to set up a bastion host in Amazon VPC. It should only be accessed from the corporate data center via SSH. What is the best way to achieve this?

- Create a large EC2 instance with a security group which only allows access on port 22 via the IP address of the corporate data center. Use a private key (.pem) file to connect to the bastion host.
- Create a large EC2 instance with a security group which only allows access on port 22 using your own pre-configured password.
- **Create a small EC2 instance with a security group which only allows access on port 22 via the IP address of the corporate data center. Use a private key (.pem) file to connect to the bastion host.**
- Create a small EC2 instance with a security group which only allows access on port 22 using your own pre-configured password.

Correct

The best way to implement a bastion host is to create a small EC2 instance which should only have a security group from a particular IP address for maximum security. This will block any SSH Brute Force attacks on your bastion host. It is also recommended to use a small instance rather than a large one because this host will only act as a jump server to connect to other instances in your VPC and nothing else.



Therefore, there is no point of allocating a large instance simply because it doesn't need that much computing power to process SSH (port 22) or RDP (port 3389) connections. It is possible to use SSH with an ordinary user ID and a pre-configured password as credentials but it is more secure to use public key pairs for SSH authentication for better security.

Hence, the right answer for this scenario is the option that says: ***Create a small EC2 instance with a security group which only allows access on port 22 via the IP address of the corporate data center. Use a private key (.pem) file to connect to the bastion host***.

Creating a large EC2 instance with a security group which only allows access on port 22 using your own pre-configured password and ***creating a small EC2 instance with a security group which only allows access on port 22 using your own pre-configured password*** are incorrect. Even though you have your own pre-configured password, the SSH connection can still be accessed by anyone over the Internet, which poses as a security vulnerability.

The option that says: ***Create a large EC2 instance with a security group which only allows access on port 22 via the IP address of the corporate data center. Use a private key (.pem) file to connect to the bastion host*** is incorrect because you don't need a large instance for a bastion host as it does not require much CPU resources.

References:

<https://docs.aws.amazon.com/quickstart/latest/linux-bastion/architecture.html>

<https://aws.amazon.com/blogs/security/how-to-record-ssh-sessions-established-through-a-bastion-host/>

Check out this Amazon VPC Cheat Sheet:

19. QUESTION

Category: CSAA – Design Resilient Architectures

A company runs a messaging application in the `ap-northeast-1` and `ap-southeast-2` region. A Solutions Architect needs to create a routing policy wherein a larger portion of traffic from the Philippines and North India will be routed to the resource in the `ap-northeast-1` region.

Which Route 53 routing policy should the Solutions Architect use?

- Weighted Routing
- Latency Routing
- Geolocation Routing
- **Geoproximity Routing**

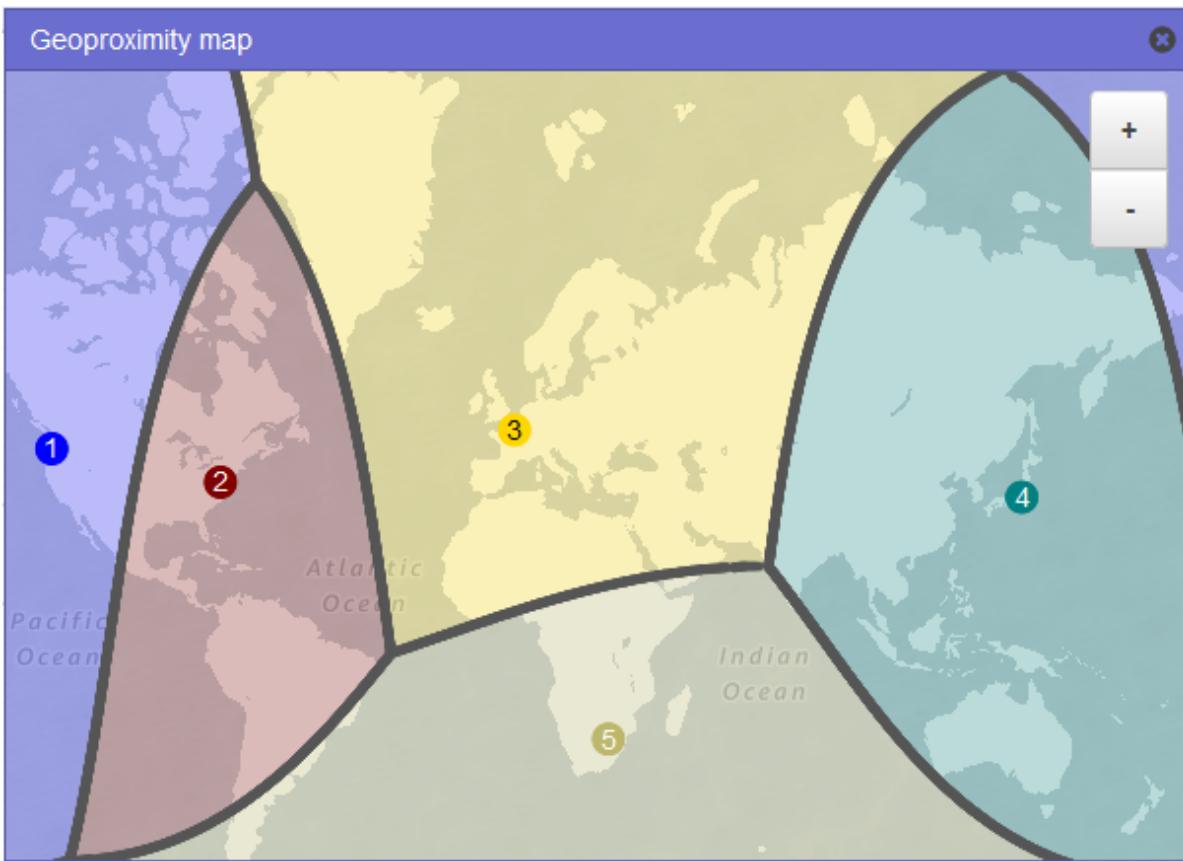
Incorrect

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. After you create a hosted zone for your domain, such as example.com, you create records to tell the Domain Name System (DNS) how you want traffic to be routed for that domain.

For example, you might create records that cause DNS to do the following:

- Route Internet traffic for example.com to the IP address of a host in your data center.
- Route email for that domain (jose.rizal@tutorialsdojo.com) to a mail server (mail.tutorialsdojo.com).
- Route traffic for a subdomain called operations.manila.tutorialsdojo.com to the IP address of a different host.

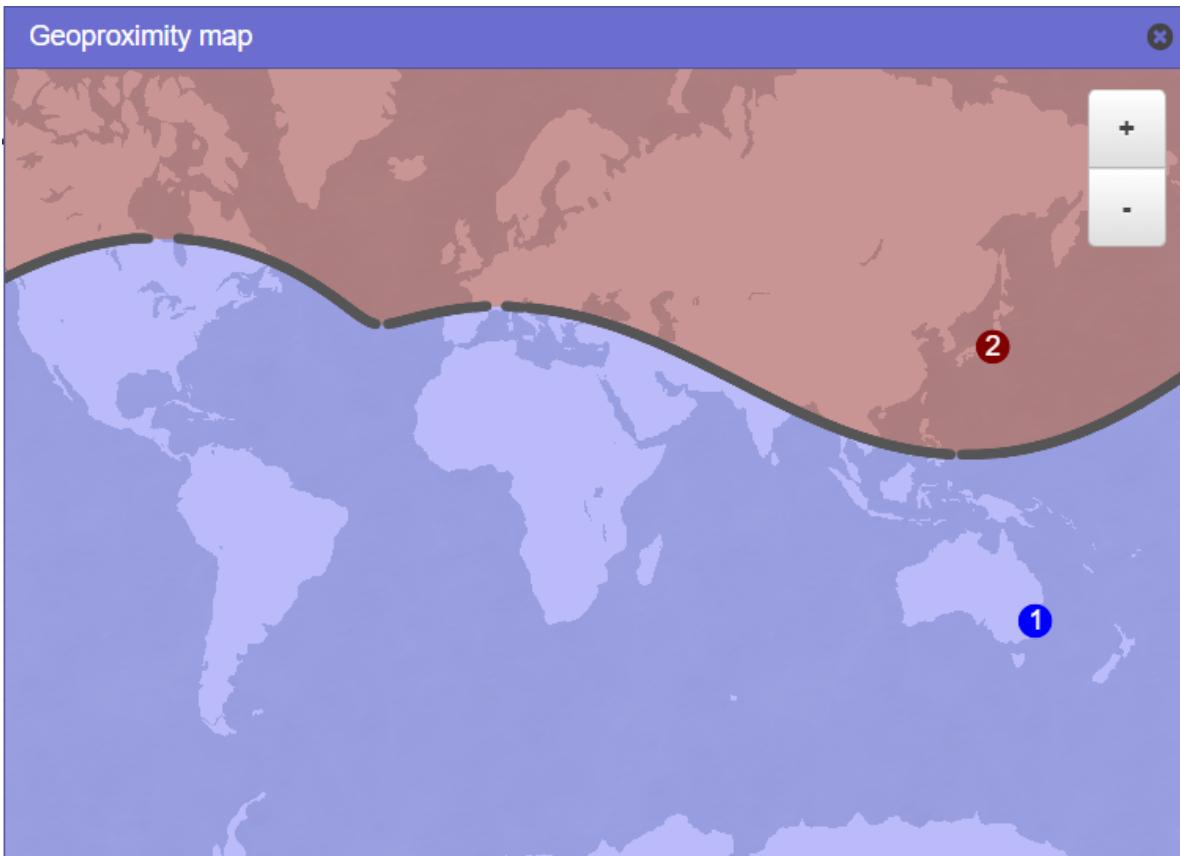
Each record includes the name of a domain or a subdomain, a record type (for example, a record with a type of MX routes email), and other information applicable to the record type (for MX records, the hostname of one or more mail servers and a priority for each server).



Route 53 has different routing policies that you can choose from. Below are some of the policies:

- **Latency Routing** lets Amazon Route 53 serve user requests from the AWS Region that provides the lowest latency. It does not, however, guarantee that users in the same geographic region will be served from the same location.
- **Geoproximity Routing** lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a **bias**. A **bias** expands or shrinks the size of the geographic region from which traffic is routed to a resource.
- **Geolocation Routing** lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from.
- **Weighted Routing** lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (subdomain.tutorialsdojo.com) and choose how much traffic is routed to each resource.

In this scenario, the problem requires a routing policy that will let Route 53 route traffic to the resource in the Tokyo region from a larger portion of the Philippines and North India.



You need to use Geoproximity Routing and specify a bias to control the size of the geographic region from which traffic is routed to your resource. The sample image above uses a bias of -40 in the Tokyo region and a bias of 1 in the Sydney Region. Setting up the bias configuration in this manner would cause Route 53 to route traffic coming from the middle and northern part of the Philippines, as well as the northern part of India to the resource in the Tokyo Region.

Hence, the correct answer is: ***Geoproximity Routing.***

Geolocation Routing is incorrect because you cannot control the coverage size from which traffic is routed to your instance in Geolocation Routing. It just lets you choose the instances that will serve traffic based on the location of your users.

Latency Routing is incorrect because it is mainly used for improving performance by letting Route 53 serve user requests from the AWS Region that provides the lowest latency.

Weighted Routing is incorrect because it is used for routing traffic to multiple resources in proportions that you specify. This can be useful for load balancing and testing new versions of a software.

References:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geoproximity>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/rrsets-working-with.html>

Latency Routing vs Geoproximity Routing vs Geolocation Routing:

<https://tutorialsdojo.com/latency-routing-vs-geoproximity-routing-vs-geolocation-routing/>

20. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A start-up company that offers an intuitive financial data analytics service has consulted you about their AWS architecture. They have a fleet of Amazon EC2 worker instances that process financial data and then outputs reports which are used by their clients. You must store the generated report files in a durable storage. The number of files to be stored can grow over time as the start-up company is expanding rapidly overseas and hence, they also need a way to distribute the reports faster to clients located across the globe.

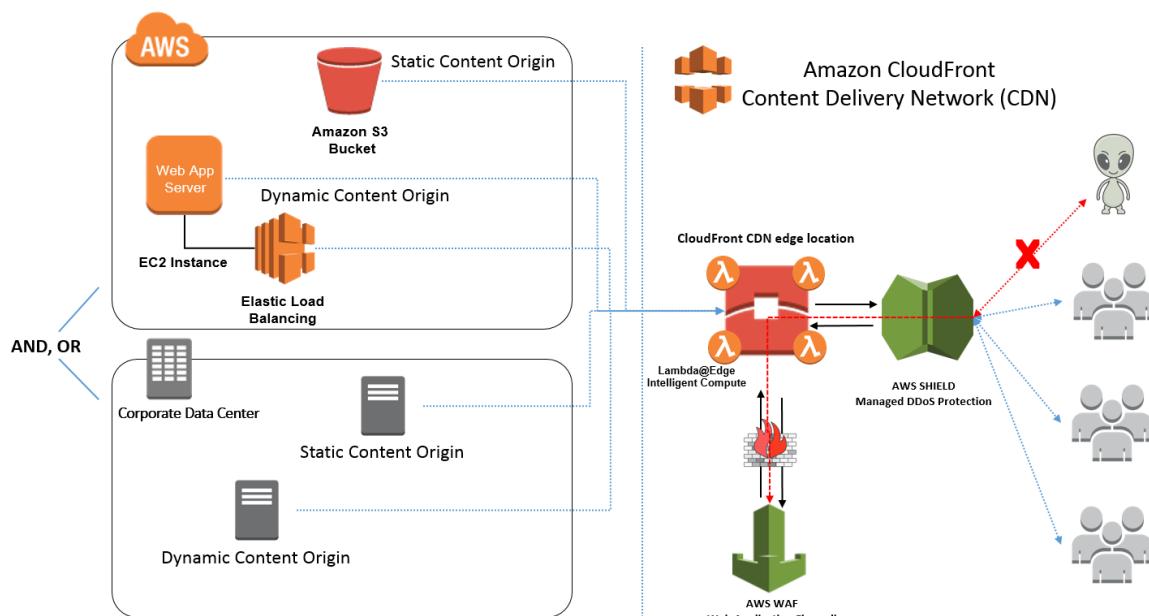
Which of the following is a cost-efficient and scalable storage option that you should use for this scenario?

- Use Amazon S3 as the data storage and CloudFront as the CDN.
- Use Amazon S3 Glacier as the data storage and ElastiCache as the CDN.
- Use multiple EC2 instance stores for data storage and ElastiCache as the CDN.
- Use Amazon Redshift as the data storage and CloudFront as the CDN.

Correct

A Content Delivery Network (CDN) is a critical component of nearly any modern web application. It used to be that CDN merely improved the delivery of content by replicating commonly requested files (static content) across a globally distributed set of caching servers. However, CDNs have become much more useful over time.

For caching, a CDN will reduce the load on an application origin and improve the experience of the requestor by delivering a local copy of the content from a nearby cache edge, or Point of Presence (PoP). The application origin is off the hook for opening the connection and delivering the content directly as the CDN takes care of the heavy lifting. The end result is that the application origins don't need to scale to meet demands for static content.



Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.

Amazon S3 offers a highly durable, scalable, and secure destination for backing up and archiving your critical data. This is the correct option as the start-up company is looking for a durable storage to store the audio and text files. In addition, ElastiCache is only used for caching and not specifically as a Global Content Delivery Network (CDN).

Using Amazon Redshift as the data storage and CloudFront as the CDN is incorrect as Amazon Redshift is usually used as a Data Warehouse.

Using Amazon S3 Glacier as the data storage and ElastiCache as the CDN is incorrect as Amazon S3 Glacier is usually used for data archives.

Using multiple EC2 instance stores for data storage and ElastiCache as the CDN is incorrect as data stored in an instance store is not durable.

References:

<https://aws.amazon.com/s3/>

<https://aws.amazon.com/caching/cdn/>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

21. QUESTION

Category: CSAA – Design Resilient Architectures

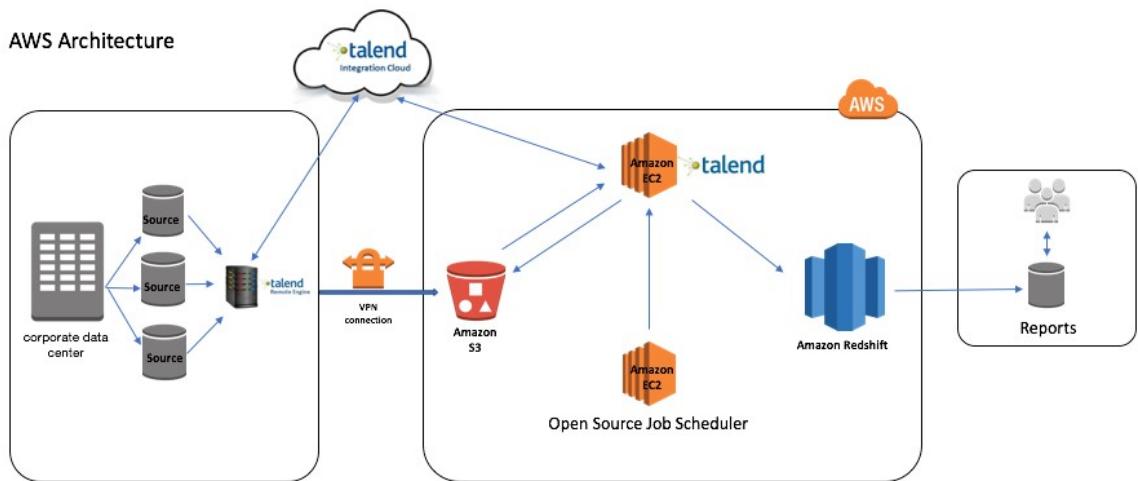
A Solutions Architect is working for an online hotel booking firm with terabytes of customer data coming from the websites and applications. There is an annual corporate meeting where the Architect needs to present the booking behavior and acquire new insights from the customers' data. The Architect is looking for a service to perform super-fast analytics on massive data sets in near real-time.

Which of the following services gives the Architect the ability to store huge amounts of data and perform quick and flexible queries on it?

- Amazon RDS
- Amazon ElastiCache
- Amazon DynamoDB
- **Amazon Redshift**

Correct

Amazon Redshift is a fast, scalable data warehouse that makes it simple and cost-effective to analyze all your data across your data warehouse and data lake. Redshift delivers ten times faster performance than other data warehouses by using machine learning, massively parallel query execution, and columnar storage on high-performance disk.



You can use Redshift to analyze all your data using standard SQL and your existing Business Intelligence (BI) tools. It also allows you to run complex analytic queries against terabytes to petabytes of structured and semi-structured data, using sophisticated query optimization, columnar storage on high-performance storage, and massively parallel query execution.

Hence, the correct answer is: ***Amazon Redshift.***

Amazon DynamoDB is incorrect. DynamoDB is a NoSQL database which is based on key-value pairs used for fast processing of small data that dynamically grows and changes. But if you need to scan large amounts of data (ie a lot of keys all in one query), the performance will not be optimal.

Amazon ElastiCache is incorrect because this is used to increase the performance, speed, and redundancy with which applications can retrieve data by providing an in-memory database caching system, and not for database analytical processes.

Amazon RDS is incorrect because this is mainly used for On-Line Transaction Processing (OLTP) applications and not for Online Analytics Processing (OLAP).

References:

<https://docs.aws.amazon.com/redshift/latest/mgmt/welcome.html>

<https://docs.aws.amazon.com/redshift/latest/gsg/getting-started.html>

Check out this Amazon Redshift Cheat Sheet:

<https://tutorialsdojo.com/amazon-redshift/>

22. QUESTION

Category: CSAA – Design High-Performing Architectures

A company has a cryptocurrency exchange portal that is hosted in an Auto Scaling group of EC2 instances behind an Application Load Balancer and is deployed across multiple AWS regions. The users can be found all around the globe, but the majority are from Japan and Sweden. Because of the compliance requirements in these two locations, you want the Japanese users to connect to the servers in the `ap-northeast-1` Asia Pacific (Tokyo) region, while the Swedish users should be connected to the servers in the `eu-west-1` EU (Ireland) region.

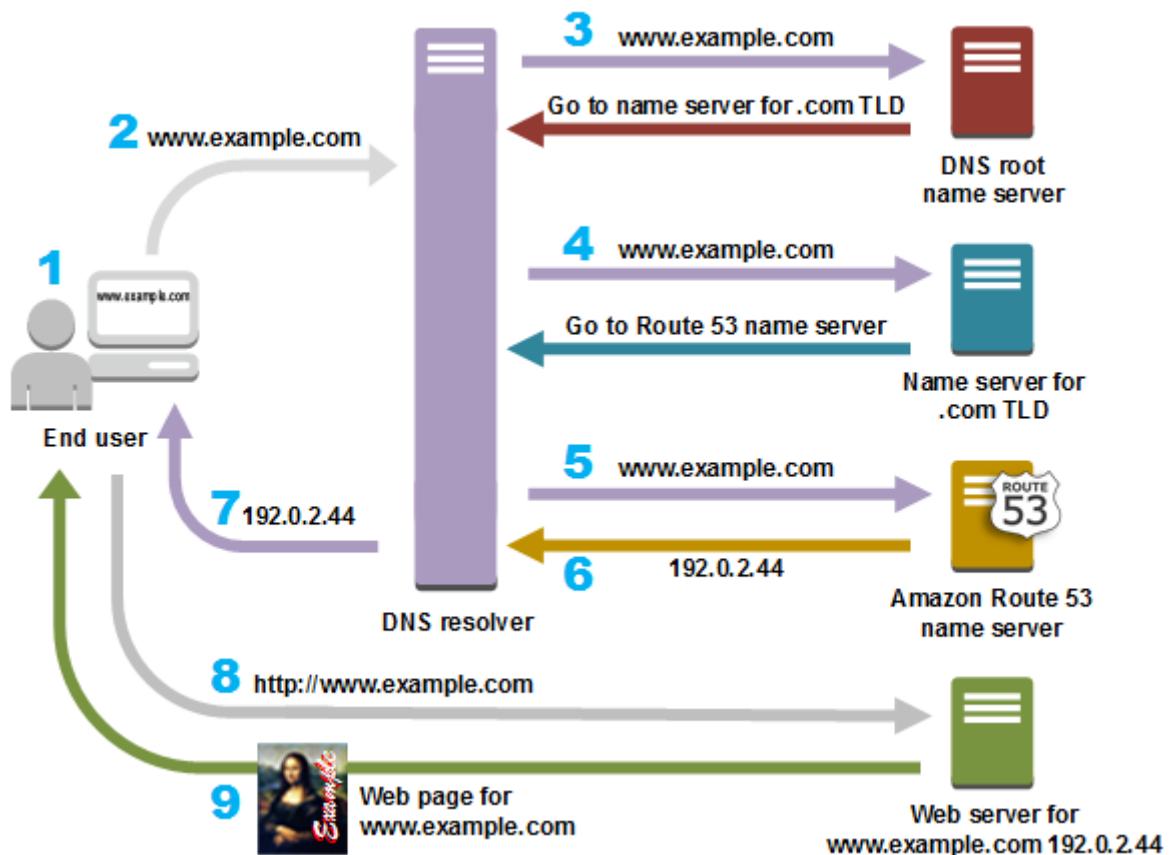
Which of the following services would allow you to easily fulfill this requirement?

- Use Route 53 Geolocation Routing policy.
- Set up an Application Load Balancers that will automatically route the traffic to the proper AWS region.
- Set up a new CloudFront web distribution with the geo-restriction feature enabled.
- Use Route 53 Weighted Routing policy.

Correct

Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from. For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region.

When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.



Setting up an Application Load Balancers that will automatically route the traffic to the proper AWS region is incorrect because Elastic Load Balancers distribute traffic among EC2 instances across multiple Availability Zones but not across AWS regions.

Setting up a new CloudFront web distribution with the geo-restriction feature enabled is incorrect because the CloudFront geo-restriction feature is primarily used to prevent users in specific geographic locations from accessing content that you're distributing through a CloudFront web distribution. It does not let you choose the resources that serve your traffic based on the geographic location of your users, unlike the Geolocation routing policy in Route 53.

Using Route 53 Weighted Routing policy is incorrect because this is not a suitable solution to meet the requirements of this scenario. It just lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (forums.tutorialsdojo.com) and choose how much traffic is routed to each resource. You have to use a Geolocation routing policy instead.

References:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/geolocation-routing-policy/>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

Latency Routing vs Geoproximity Routing vs Geolocation Routing:

<https://tutorialsdojo.com/latency-routing-vs-geoproximity-routing-vs-geolocation-routing/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

23. QUESTION

Category: CSAA – Design Secure Applications and Architectures

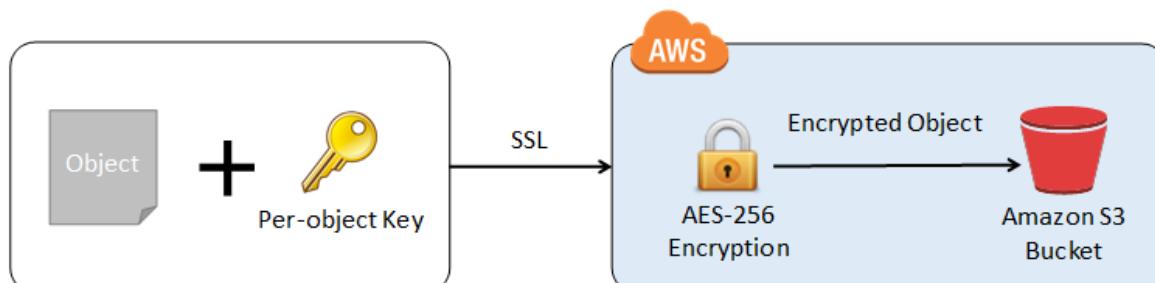
All objects uploaded to an Amazon S3 bucket must be encrypted for security compliance. The bucket will use server-side encryption with Amazon S3-Managed encryption keys (SSE-S3) to encrypt data using 256-bit Advanced Encryption Standard (AES-256) block cipher.

Which of the following request headers must be used?

- `x-amz-server-side-encryption`
- `x-amz-server-side-encryption-customer-key-MD5`
- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`

Correct

Server-side encryption protects data at rest. If you use Server-Side Encryption with Amazon S3-Managed Encryption Keys (SSE-S3), Amazon S3 will encrypt each object with a unique key and as an additional safeguard, it encrypts the key itself with a master key that it rotates regularly. Amazon S3 server-side encryption uses one of the strongest block ciphers available, 256-bit Advanced Encryption Standard (AES-256), to encrypt your data.



If you need server-side encryption for all of the objects that are stored in a bucket, use a bucket policy. For example, the following bucket policy denies permissions to upload an object unless the request includes the **x-amz-server-side-encryption** header to request server-side encryption:

However, if you chose to use server-side encryption with customer-provided encryption keys (SSE-C), you must provide encryption key information using the following request headers:

- `x-amz-server-side-encryption-customer-algorithm`
- `x-amz-server-side-encryption-customer-key`
- `x-amz-server-side-encryption-customer-key-MD5`

Hence, using the ***x-amz-server-side-encryption*** header is correct as this is the one being used for Amazon S3-Managed Encryption Keys (SSE-S3).

All other options are incorrect since they are used for SSE-C.

References:

<https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerSideEncryptionCustomerKeys.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

24. QUESTION

Category: CSAA – Design High-Performing Architectures

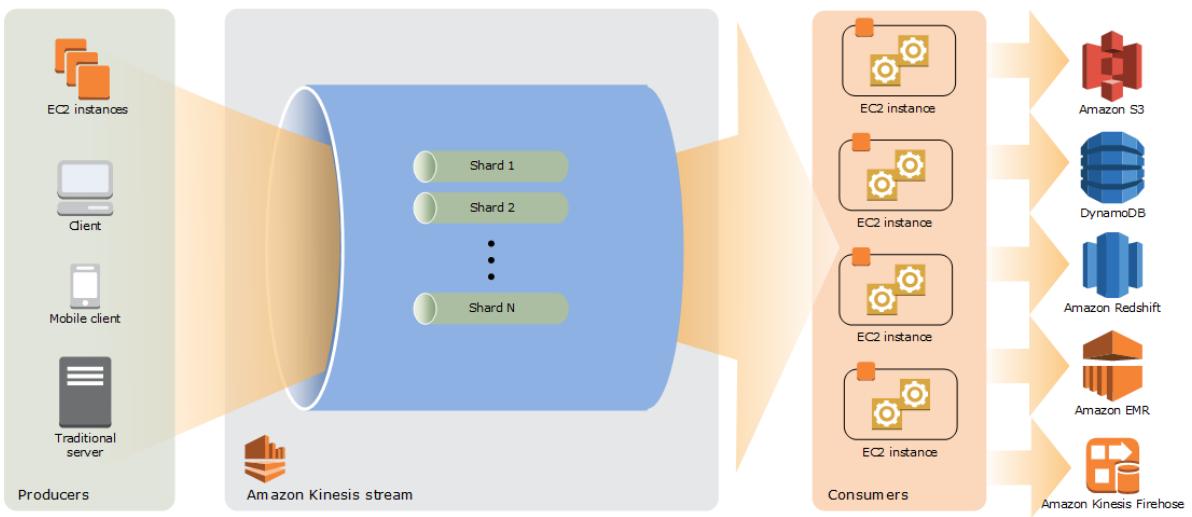
A company has a data analytics application that updates a real-time, foreign exchange dashboard and another separate application that archives data to Amazon Redshift. Both applications are configured to consume data from the same stream concurrently and independently by using Amazon Kinesis Data Streams. However, they noticed that there are a lot of occurrences where a shard iterator expires unexpectedly. Upon checking, they found out that the DynamoDB table used by Kinesis does not have enough capacity to store the lease data.

Which of the following is the most suitable solution to rectify this issue?

- Upgrade the storage capacity of the DynamoDB table.
- Use Amazon Kinesis Data Analytics to properly support the data analytics application instead of Kinesis Data Stream.
- Enable In-Memory Acceleration with DynamoDB Accelerator (DAX).
- Increase the write capacity assigned to the shard table.

Correct

A new shard iterator is returned by every **GetRecords** request (as `NextShardIterator`), which you then use in the next **GetRecords** request (as `ShardIterator`). Typically, this shard iterator does not expire before you use it. However, you may find that shard iterators expire because you have not called **GetRecords** for more than 5 minutes, or because you've performed a restart of your consumer application.



If the shard iterator expires immediately before you can use it, this might indicate that the DynamoDB table used by Kinesis does not have enough capacity to store the lease data. This situation is more likely to happen if you have a large number of shards. To solve this problem, increase the write capacity assigned to the shard table.

Hence, ***increasing the write capacity assigned to the shard table*** is the correct answer.

Upgrading the storage capacity of the DynamoDB table is incorrect because DynamoDB is a fully managed service which automatically scales its storage, without setting it up manually. The scenario refers to the ****write capacity**** of the shard table as it says that the DynamoDB table used by Kinesis does not have enough *capacity* to store the lease data.

Enabling In-Memory Acceleration with DynamoDB Accelerator (DAX) is incorrect because the DAX feature is primarily used for read performance improvement of your DynamoDB table from *milliseconds* response time to *microseconds*. It does not have any relationship with Amazon Kinesis Data Stream in this scenario.

Using Amazon Kinesis Data Analytics to properly support the data analytics application instead of Kinesis Data Stream is incorrect. Although Amazon Kinesis Data Analytics can support a data analytics application, it is still not a suitable solution for this issue. You simply need to increase the write capacity assigned to the shard table in order to rectify the problem which is why switching to Amazon Kinesis Data Analytics is not necessary.

Reference:

<https://docs.aws.amazon.com/streams/latest/dev/kinesis-record-processor-ddb.html>

<https://docs.aws.amazon.com/streams/latest/dev/troubleshooting-consumers.html>

Check out this Amazon Kinesis Cheat Sheet:

<https://tutorialsdojo.com/amazon-kinesis/>

25. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A digital media company shares static content to its premium users around the world and also to their partners who syndicate their media files. The company is looking for ways to reduce its server costs and securely deliver their data to their customers globally with low latency.

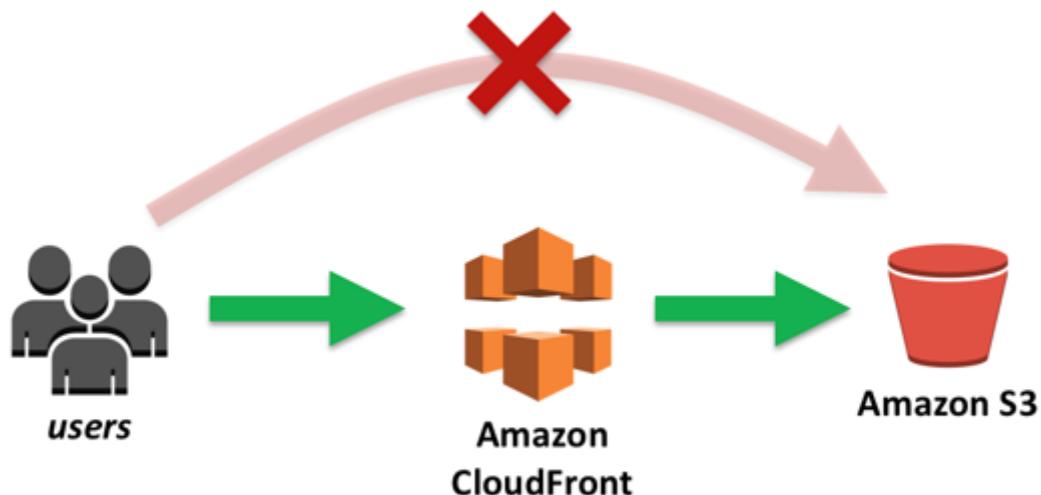
Which combination of services should be used to provide the MOST suitable and cost-effective architecture? (Select TWO.)

- Amazon CloudFront
- AWS Fargate
- AWS Lambda
- Amazon S3
- AWS Global Accelerator

Correct

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers' users and to customize the user experience. Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don't pay for any data transferred between these services and CloudFront.



Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.

AWS Global Accelerator and Amazon CloudFront are separate services that use the AWS global network and its edge locations around the world. CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery). Global Accelerator improves performance for a wide range of applications over TCP or UDP by proxying packets at the edge to applications running in one or more AWS Regions. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Both services integrate with AWS Shield for DDoS protection.

Hence, the correct options are ***Amazon CloudFront*** and ***Amazon S3.***

AWS Fargate is incorrect because this service is just a serverless compute engine for containers that work with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Although this service is more cost-effective than its server-based counterpart, Amazon S3 still costs way less than Fargate, especially for storing static content.

AWS Lambda is incorrect because this simply lets you run your code serverless, without provisioning or managing servers. Although this is also a cost-effective service since you have to pay only for the compute time you consume, you can't use this to store static content or as a Content Delivery Network (CDN). A better combination is Amazon CloudFront and Amazon S3.

AWS Global Accelerator is incorrect because this service is more suitable for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Moreover, there is no direct way that you can integrate AWS Global Accelerator with Amazon S3. It's more suitable to use Amazon CloudFront instead in this scenario.

References:

<https://aws.amazon.com/premiumsupport/knowledge-center/cloudfront-serve-static-website/>

<https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/>

<https://aws.amazon.com/global-accelerator/faqs/>

26. QUESTION

Category: CSAA – Design High-Performing Architectures

A media company is setting up an ECS batch architecture for its image processing application. It will be hosted in an Amazon ECS Cluster with two ECS tasks that will handle image uploads from the users and image processing. The first ECS task will process the user requests, store the image in an S3 input bucket, and push a message to a queue. The second task reads from the queue, parses the message containing the object name, and then downloads the object. Once the image is processed and transformed, it will upload the objects to the S3 output bucket. To complete the architecture, the Solutions Architect must create a queue and the necessary IAM permissions for the ECS tasks.

Which of the following should the Architect do next?

- Launch a new Amazon MQ queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Amazon MQ queue. Set the (`EnableTaskIAMRole`) option to true in the task definition.
- Launch a new Amazon Kinesis Data Firehose and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Kinesis Data Firehose. Specify the ARN of the IAM Role in the (`taskDefinitionArn`) field of the task definition.
- Launch a new Amazon AppStream 2.0 queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and AppStream 2.0 queue. Declare the IAM Role (`taskRoleArn`) in the task definition.
- Launch a new Amazon SQS queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and SQS queue. Declare the IAM Role (`taskRoleArn`) in the task definition.

Correct

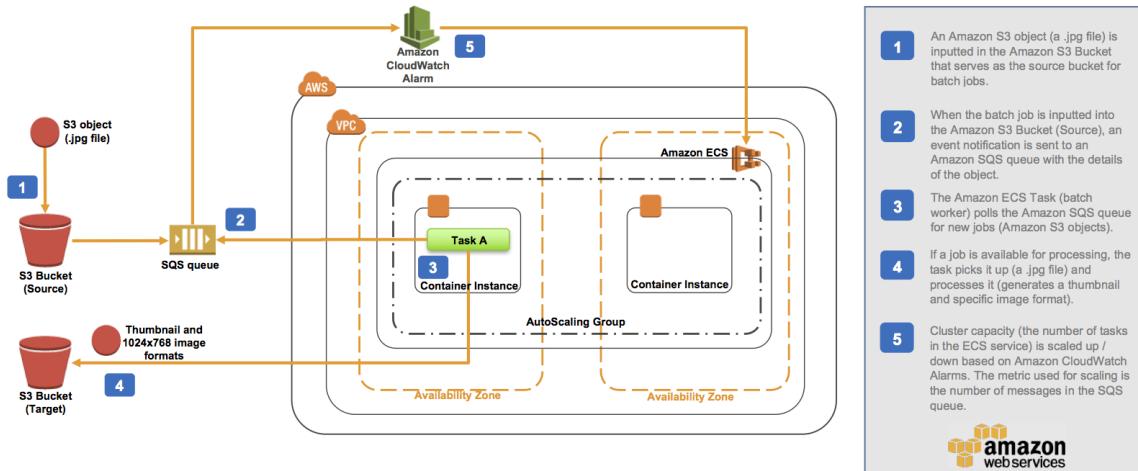
Docker containers are particularly suited for batch job workloads. Batch jobs are often short-lived and embarrassingly parallel. You can package your batch processing application into a Docker image so that you can deploy it anywhere, such as in an Amazon ECS task.

Amazon ECS supports batch jobs. You can use Amazon ECS *Run Task* action to run one or more tasks once. The Run Task action starts the ECS task on an instance that meets the task's requirements including CPU, memory, and ports.

Amazon ECS Batch Processing

Build a batch processing framework to automate your batch jobs

This diagram shows how to use Amazon S3, Amazon SQS, and Amazon ECS to build an automated batch processing framework.



AWS Reference Architectures

For example, you can set up an ECS Batch architecture for an image processing application. You can set up an AWS CloudFormation template that creates an Amazon S3 bucket, an Amazon SQS queue, an Amazon CloudWatch alarm, an ECS cluster, and an ECS task definition. Objects uploaded to the input S3 bucket trigger an event that sends object details to the SQS queue. The ECS task deploys a Docker container that reads from that queue, parses the message containing the object name and then downloads the object. Once transformed it will upload the objects to the S3 output bucket.

By using the SQS queue as the location for all object details, you can take advantage of its scalability and reliability as the queue will automatically scale based on the incoming messages and message retention can be configured. The ECS Cluster will then be able to scale services up or down based on the number of messages in the queue.

You have to create an IAM Role that the ECS task assumes in order to get access to the S3 buckets and SQS queue. Note that the permissions of the IAM role don't specify the S3 bucket ARN for the incoming bucket. This is to avoid a circular dependency issue in the CloudFormation template. You should always make sure to assign the least amount of privileges needed to an IAM role.

Hence, the correct answer is: ***Launch a new Amazon SQS queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and SQS queue. Declare the IAM Role (`taskRoleArn`) in the task definition.***

The option that says: ***Launch a new Amazon AppStream 2.0 queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and AppStream 2.0 queue. Declare the IAM Role (taskRoleArn) in the task definition*** is incorrect because Amazon AppStream 2.0 is a fully managed application streaming service and can't be used as a queue. You have to use Amazon SQS instead.

The option that says: ***Launch a new Amazon Kinesis Data Firehose and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Kinesis Data Firehose. Specify the ARN of the IAM Role in the (taskDefinitionArn) field of the task definition*** is incorrect because Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data. Although it can stream data to an S3 bucket, it is not suitable to be used as a queue for a batch application in this scenario. In addition, the ARN of the IAM Role should be declared in the `taskRoleArn` and not in the `taskDefinitionArn` field.

The option that says: ***Launch a new Amazon MQ queue and configure the second ECS task to read from it. Create an IAM role that the ECS tasks can assume in order to get access to the S3 buckets and Amazon MQ queue. Set the (EnableTaskIAMRole) option to true in the task definition*** is incorrect because Amazon MQ is primarily used as a managed message broker service and not a queue. The `EnableTaskIAMRole` option is only applicable for Windows-based ECS Tasks that require extra configuration.

References:

<https://github.com/aws-samples/ecs-refarch-batch-processing>

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/common_use_cases.html

<https://aws.amazon.com/ecs/faqs/>

27. QUESTION

Category: CSAA – Design Resilient Architectures

A company has a static corporate website hosted in a standard S3 bucket and a new web domain name that was registered using Route 53. You are instructed by your manager to integrate these two services in order to successfully launch their corporate website.

What are the prerequisites when routing traffic using Amazon Route 53 to a website that is hosted in an Amazon S3 Bucket? (Select TWO.)

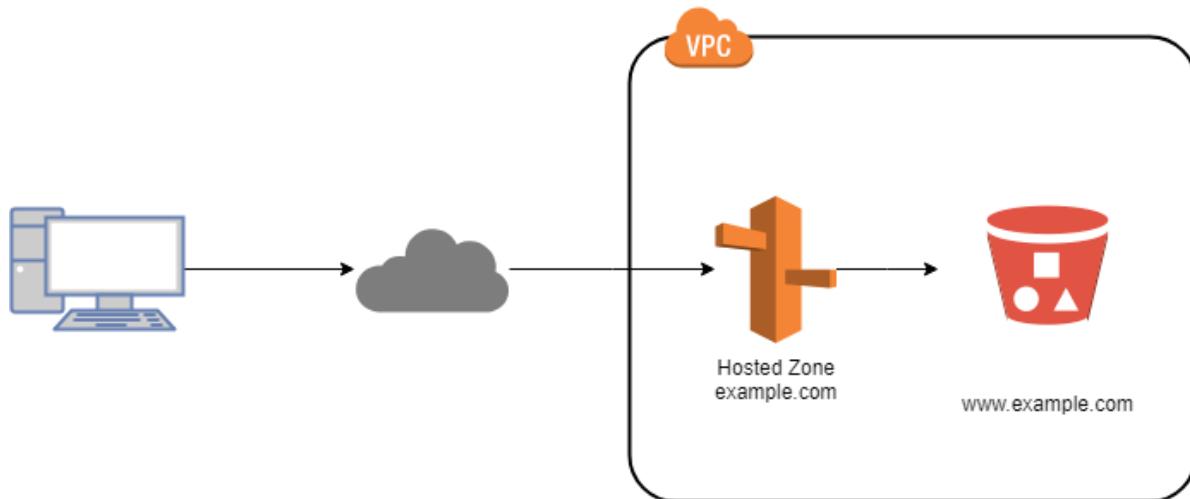
- The S3 bucket name must be the same as the domain name
- A registered domain name
- The record set must be of type "MX"
- The Cross-Origin Resource Sharing (CORS) option should be enabled in the S3 bucket
- The S3 bucket must be in the same region as the hosted zone

Incorrect

Here are the prerequisites for routing traffic to a website that is hosted in an Amazon S3 Bucket:

- An S3 bucket that is configured to host a static website. The bucket must have the same name as your domain or subdomain. For example, if you want to use the subdomain portal.tutorialsdojo.com, the name of the bucket must be portal.tutorialsdojo.com.

- A registered domain name. You can use Route 53 as your domain registrar, or you can use a different registrar.
- Route 53 as the DNS service for the domain. If you register your domain name by using Route 53, we automatically configure Route 53 as the DNS service for the domain.



The option that says: ***The record set must be of type "MX"*** is incorrect since an MX record specifies the mail server responsible for accepting email messages on behalf of a domain name. This is not what is being asked by the question.

The option that says: ***The S3 bucket must be in the same region as the hosted zone*** is incorrect. There is no constraint that the S3 bucket must be in the same region as the hosted zone in order for the Route 53 service to route traffic into it.

The option that says: ***The Cross-Origin Resource Sharing (CORS) option should be enabled in the S3 bucket*** is incorrect because you only need to enable Cross-Origin Resource Sharing (CORS) when your client web application on one domain interacts with the resources in a different domain.

Reference:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/RoutingToS3Bucket.html>

Amazon Route 53 Overview:

<https://www.youtube.com/embed/Su308t19ubY>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

28. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An online events registration system is hosted in AWS and uses ECS to host its front-end tier and an RDS configured with Multi-AZ for its database tier. What are the events that will make Amazon RDS automatically perform a failover to the standby replica? (Select TWO.)

- Loss of availability in primary Availability Zone
- Storage failure on primary
- Compute unit failure on secondary DB instance
- Storage failure on secondary DB instance

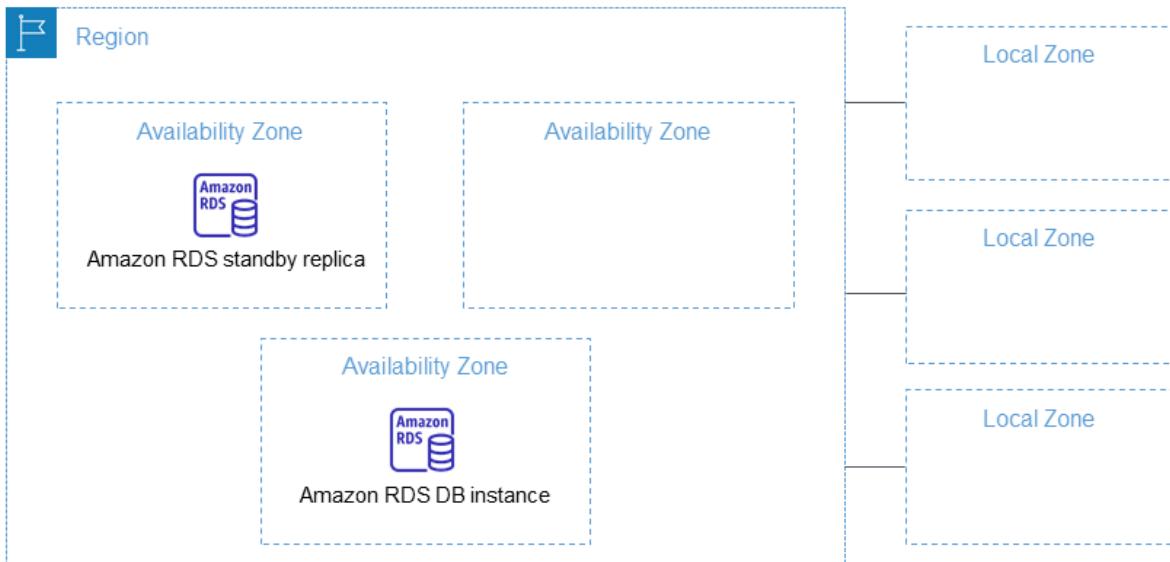
- In the event of Read Replica failure

Correct

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support. Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM).

In a Multi-AZ deployment, Amazon RDS automatically provisions and maintains a synchronous standby replica in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to a standby replica to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups. Running a DB instance with high availability can enhance availability during planned system maintenance, and help protect your databases against DB instance failure and Availability Zone disruption.

Amazon RDS detects and automatically recovers from the most common failure scenarios for Multi-AZ deployments so that you can resume database operations as quickly as possible without administrative intervention.



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.

Amazon RDS automatically performs a failover in the event of any of the following:

1. Loss of availability in primary Availability Zone.
2. Loss of network connectivity to primary.
3. Compute unit failure on primary.
4. Storage failure on primary.

Hence, the correct answers are:

- *- **Loss of availability in primary Availability Zone***
- *- **Storage failure on primary***

The following options are incorrect because all these scenarios do not affect the primary database. Automatic failover only occurs if the primary database is the one that is affected.

- *- **Storage failure on secondary DB instance***
- *- **In the event of Read Replica failure***

- Compute unit failure on secondary DB instance

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.MultiAZ.html>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

29. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

An organization is currently using a tape backup solution to store its application data on-premises. They plan to use a cloud storage service to preserve the backup data for up to 10 years that may be accessed about once or twice a year.

Which of the following is the most cost-effective option to implement this solution?

- Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier.
- Order an AWS Snowball Edge appliance to import the backup directly to Amazon S3 Glacier.
- **Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier Deep Archive.**
- Use Amazon S3 to store the backup data and add a lifecycle rule to transition the current version to Amazon S3 Glacier.

Incorrect

Tape Gateway enables you to replace using physical tapes on-premises with virtual tapes in AWS without changing existing backup workflows. Tape Gateway supports all leading backup applications and caches virtual tapes on-premises for low-latency data access. Tape Gateway encrypts data between the gateway and AWS for secure data transfer and compresses data and transitions virtual tapes between Amazon S3 and Amazon S3 Glacier, or Amazon S3 Glacier Deep Archive, to minimize storage costs.



The scenario requires you to backup your application data to a cloud storage service for long-term retention of data that will be retained for 10 years. Since it uses a tape backup solution, an option that uses AWS Storage Gateway must be the possible answer. Tape Gateway can move your virtual tapes archived in Amazon S3 Glacier or Amazon S3 Glacier Deep Archive storage class, enabling you to further reduce the monthly cost to store long-term data in the cloud by up to 75%.

Hence, the correct answer is: ***Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier Deep Archive***.

The option that says: ***Use AWS Storage Gateway to backup the data directly to Amazon S3 Glacier*** is incorrect. Although this is a valid solution, moving to S3 Glacier is more expensive than directly backing it up to Glacier Deep Archive.

The option that says: ***Order an AWS Snowball Edge appliance to import the backup directly to Amazon S3 Glacier*** is incorrect because Snowball Edge can't directly integrate backups to S3 Glacier. Moreover, you have to use the Amazon S3 Glacier Deep Archive storage class as it is more cost-effective than the regular Glacier class.

The option that says: ***Use Amazon S3 to store the backup data and add a lifecycle rule to transition the current version to Amazon S3 Glacier*** is incorrect. Although this is a possible solution, it is difficult to directly integrate a tape backup solution to S3 without using Storage Gateway.

References:

<https://aws.amazon.com/storagegateway/faqs/>

<https://aws.amazon.com/s3/storage-classes/>

AWS Storage Gateway Overview:

<https://www.youtube.com/embed/pNb7xOBJjHE>

Check out this AWS Storage Gateway Cheat Sheet:

<https://tutorialsdojo.com/aws-storage-gateway/>

30. QUESTION

Category: CSAA – Design Secure Applications and Architectures

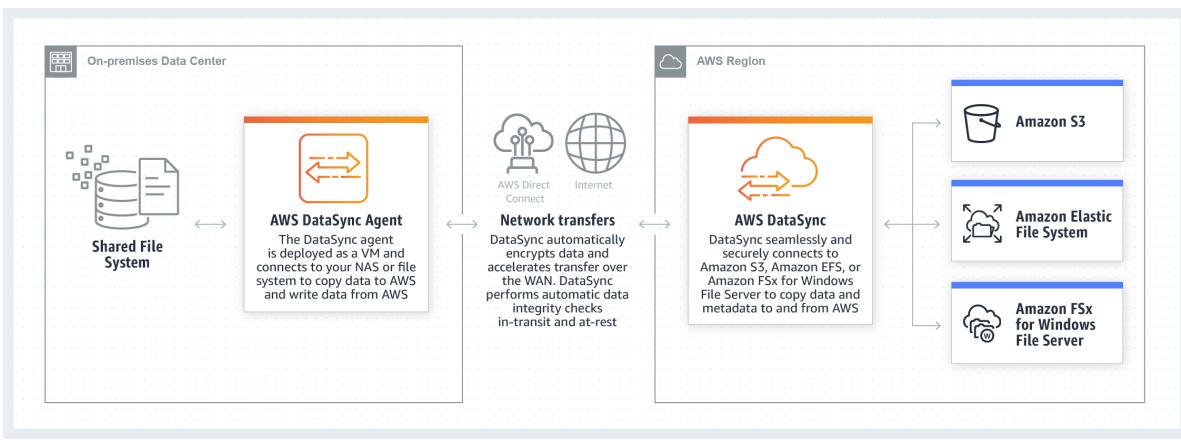
An organization stores and manages financial records of various companies in its on-premises data center, which is almost out of space. The management decided to move all of their existing records to a cloud storage service. All future financial records will also be stored in the cloud. For additional security, all records must be prevented from being deleted or overwritten.

Which of the following should you do to meet the above requirement?

- Use AWS DataSync to move the data. Store all of your data in Amazon EFS and enable object lock.
- **Use AWS DataSync to move the data. Store all of your data in Amazon S3 and enable object lock.**
- Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon S3 and enable object lock.
- Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon EBS and enable object lock.

Correct

AWS DataSync allows you to copy large datasets with millions of files, without having to build custom solutions with open source tools, or license and manage expensive commercial network acceleration software. You can use DataSync to migrate active data to AWS, transfer data to the cloud for analysis and processing, archive data to free up on-premises storage capacity, or replicate data to AWS for business continuity.



AWS DataSync enables you to migrate your on-premises data to Amazon S3, Amazon EFS, and Amazon FSx for Windows File Server. You can configure DataSync to make an initial copy of your entire dataset, and schedule subsequent incremental transfers of changing data towards Amazon S3. Enabling S3 Object Lock prevents your existing and future records from being deleted or overwritten.

AWS DataSync is primarily used to migrate existing data to Amazon S3. On the other hand, AWS Storage Gateway is more suitable if you still want to retain access to the migrated data and for ongoing updates from your on-premises file-based applications.

Hence, the correct answer in this scenario is: ***Use AWS DataSync to move the data. Store all of your data in Amazon S3 and enable object lock***.

The option that says: ***Use AWS DataSync to move the data. Store all of your data in Amazon EFS and enable object lock*** is incorrect because Amazon EFS only supports file locking. Object lock is a feature of Amazon S3 and not Amazon EFS.

The options that says: ***Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon S3 and enable object lock*** is incorrect because the scenario requires that all of the existing records must be migrated to AWS. The future records will also be stored in AWS and not in the on-premises network. This means that setting up a hybrid cloud storage is not necessary since the on-premises storage will no longer be used.

The option that says: ***Use AWS Storage Gateway to establish hybrid cloud storage. Store all of your data in Amazon EBS and enable object lock*** is incorrect because Amazon EBS does not support object lock. Amazon S3 is the only service capable of locking objects to prevent an object from being deleted or overwritten.

References:

<https://aws.amazon.com/datasync/faqs/>

<https://docs.aws.amazon.com/datasync/latest/userguide/what-is-datasync.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lock.html>

Check out this AWS DataSync Cheat Sheet:

<https://tutorialsdojo.com/aws-datasync/>

AWS Storage Gateway vs DataSync:

<https://www.youtube.com/embed/tmfe1rO-AUs>

Amazon S3 vs EBS vs EFS Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/>

31. QUESTION

Category: CSAA – Design Resilient Architectures

A Network Architect developed a food ordering application. The Architect needs to retrieve the instance ID, public keys, and public IP address of the EC2 server made for tagging and grouping the attributes into the internal application running on-premises.

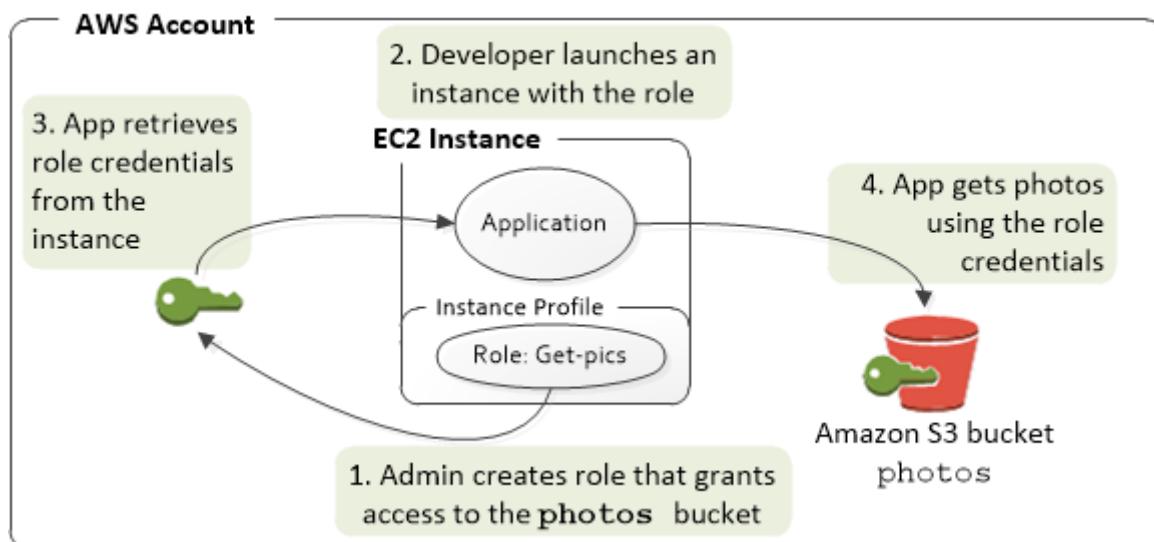
Which of the following options fulfills this requirement?

- Amazon Machine Image
- Instance metadata
- Resource tags
- Instance user data

Correct

Instance metadata is the data about your instance that you can use to configure or manage the running instance. You can get the instance ID, public keys, public IP address and many other information from the instance metadata by firing a URL command in your instance to this URL:

<http://169.254.169.254/latest/meta-data/>



Instance user data is incorrect because this is mainly used to perform common automated configuration tasks and run scripts after the instance starts.

Resource tags is incorrect because these are labels that you assign to an AWS resource. Each tag consists of a key and an optional value, both of which you define.

Amazon Machine Image is incorrect because this mainly provides the information required to launch an instance, which is a virtual server in the cloud.

Reference:

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.htm>

Check out this Amazon EC2 Cheat Sheet:

<https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

32. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An organization needs to control the access for several S3 buckets. They plan to use a gateway endpoint to allow access to trusted buckets.

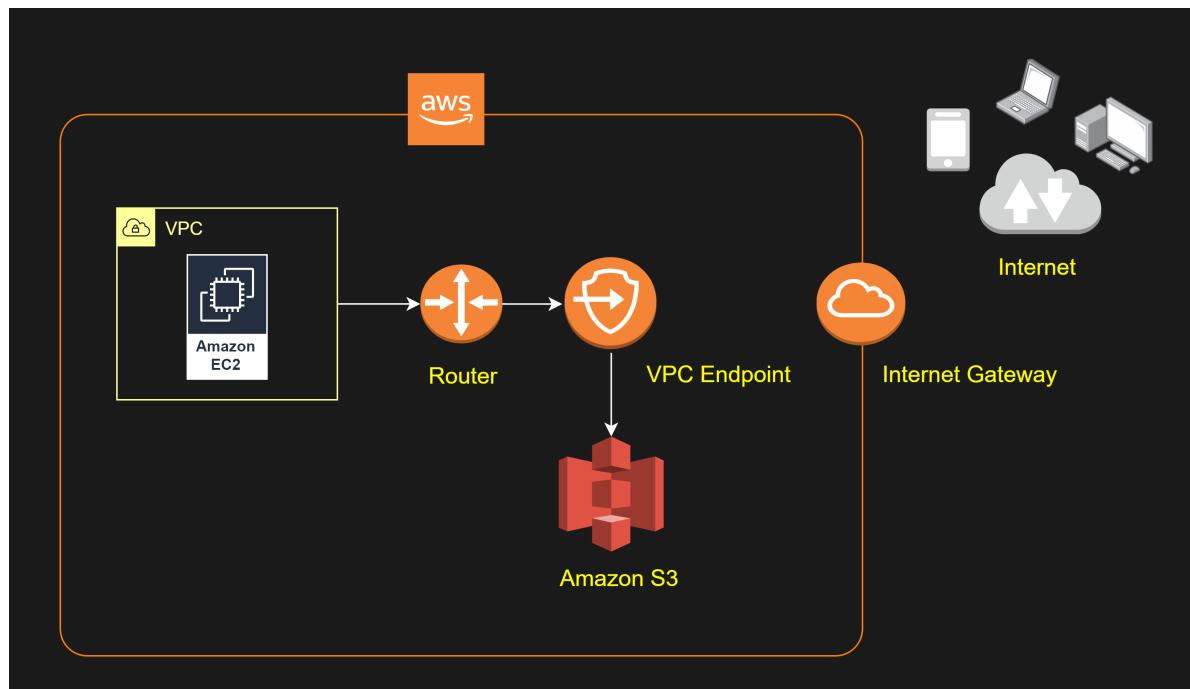
Which of the following could help you achieve this requirement?

- Generate an endpoint policy for trusted VPCs.
- **Generate an endpoint policy for trusted S3 buckets.**
- Generate a bucket policy for trusted S3 buckets.
- Generate a bucket policy for trusted VPCs.

Incorrect

A **VPC endpoint** enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.

When you create a VPC endpoint, you can attach an endpoint policy that controls access to the service to which you are connecting. You can modify the endpoint policy attached to your endpoint and add or remove the route tables used by the endpoint. An endpoint policy does not override or replace IAM user policies or service-specific policies (such as S3 bucket policies). It is a separate policy for controlling access from the endpoint to the specified service.



We can use a bucket policy or an endpoint policy to allow the traffic to trusted S3 buckets. The options that have 'trusted S3 buckets' key phrases will be the possible answer in this scenario. It would take you a lot of time to configure a bucket policy for each S3 bucket instead of using a single endpoint policy. Therefore, you should use an endpoint policy to control the traffic to the trusted Amazon S3 buckets.

Hence, the correct answer is: ***Generate an endpoint policy for trusted S3 buckets***.

The option that says: ***Generate a bucket policy for trusted S3 buckets*** is incorrect. Although this is a valid solution, it takes a lot of time to set up a bucket policy for each and every S3 bucket. This can simply be accomplished by creating an S3 endpoint policy.

The option that says: ***Generate a bucket policy for trusted VPCs*** is incorrect because you are generating a policy for trusted VPCs. Remember that the scenario only requires you to allow the traffic for trusted S3 buckets, and not to the VPCs.

The option that says: ***Generate an endpoint policy for trusted VPCs*** is incorrect because it only allows access to trusted VPCs, and not to trusted Amazon S3 buckets.

References:

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-endpoints-s3.html>

<https://aws.amazon.com/premiumsupport/knowledge-center/connect-s3-vpc-endpoint/>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

33. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An application is hosted in AWS Fargate and uses RDS database in Multi-AZ Deployments configuration with several Read Replicas. A Solutions Architect was instructed to ensure that all of their database credentials, API keys, and other secrets are encrypted and rotated on a regular basis to improve data security. The application should also use the latest version of the encrypted credentials when connecting to the RDS database.

Which of the following is the MOST appropriate solution to secure the credentials?

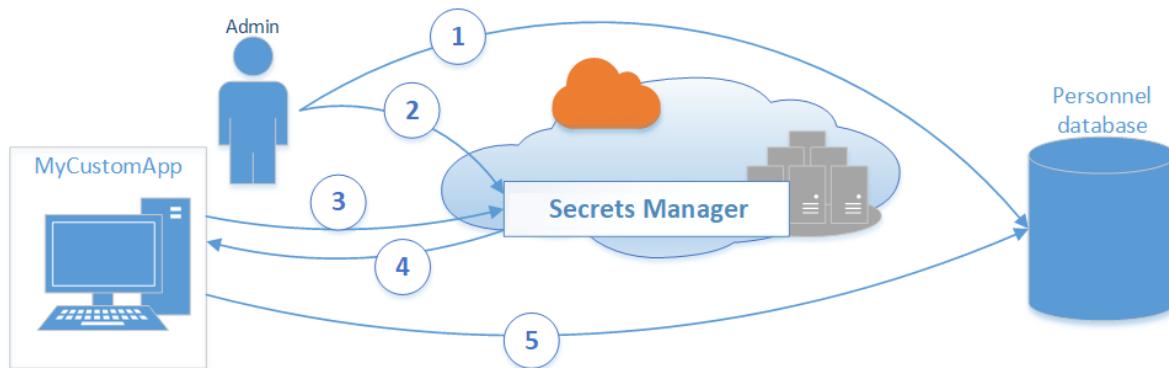
- Store the database credentials, API keys, and other secrets to Systems Manager Parameter Store each with a `SecureString` data type. The credentials are automatically rotated by default.
- **Use AWS Secrets Manager to store and encrypt the database credentials, API keys, and other secrets. Enable automatic rotation for all of the credentials.**
- Store the database credentials, API keys, and other secrets in AWS KMS.
- Store the database credentials, API keys, and other secrets to AWS ACM.

Correct

****AWS Secrets Manager**** is an AWS service that makes it easier for you to manage secrets. *Secrets* can be database credentials, passwords, third-party API keys, and even arbitrary text. You can store and control access to these secrets centrally by using the Secrets Manager console, the Secrets Manager command line interface (CLI), or the Secrets Manager API and SDKs.

In the past, when you created a custom application that retrieves information from a database, you typically had to embed the credentials (the secret) for accessing the database directly in the application. When it came time to rotate the credentials, you had to do much more than just create new credentials. You had to invest time to update the application to use the new credentials. Then you had to distribute the updated application. If you had multiple applications that shared credentials and you missed updating

one of them, the application would break. Because of this risk, many customers have chosen not to regularly rotate their credentials, which effectively substitutes one risk for another.



Secrets Manager enables you to replace hardcoded credentials in your code (including passwords), with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure that the secret can't be compromised by someone examining your code, because the secret simply isn't there. Also, you can configure Secrets Manager to automatically rotate the secret for you according to a schedule that you specify. This enables you to replace long-term secrets with short-term ones, which helps to significantly reduce the risk of compromise.

Hence, the most appropriate solution for this scenario is: ***Use AWS Secrets Manager to store and encrypt the database credentials, API keys, and other secrets. Enable automatic rotation for all of the credentials.***

The option that says: ***Store the database credentials, API keys, and other secrets to Systems Manager Parameter Store each with a SecureString data type. The credentials are automatically rotated by default*** is incorrect because Systems Manager Parameter Store doesn't rotate its parameters by default.

The option that says: ***Store the database credentials, API keys, and other secrets to AWS ACM*** is incorrect because it is just a managed private CA service that helps you easily and securely manage the lifecycle of your private certificates to allow SSL communication to your application. This is not a suitable service to store database or any other confidential credentials.

The option that says: ***Store the database credentials, API keys, and other secrets in AWS KMS*** is incorrect because this only makes it easy for you to create and manage encryption keys and control the use of encryption across a wide range of AWS services. This is primarily used for encryption and not for hosting your credentials.

References:

<https://aws.amazon.com/secrets-manager/>

<https://aws.amazon.com/blogs/security/how-to-securely-provide-database-credentials-to-lambda-functions-by-using-aws-secrets-manager/>

Check out these AWS Secrets Manager and Systems Manager Cheat Sheets:

<https://tutorialsdojo.com/aws-secrets-manager/>

<https://tutorialsdojo.com/aws-systems-manager/>

AWS Security Services Overview - Secrets Manager, ACM, Macie:

<https://youtu.be/ogVamzF2Dzk>

34. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An Intelligence Agency developed a missile tracking application that is hosted on both development and production AWS accounts. The Intelligence agency's junior developer only has access to the development account. She has received security clearance to access the agency's production account but the access is only temporary and only write access to EC2 and S3 is allowed.

Which of the following allows you to issue short-lived access tokens that act as temporary security credentials to allow access to your AWS resources?

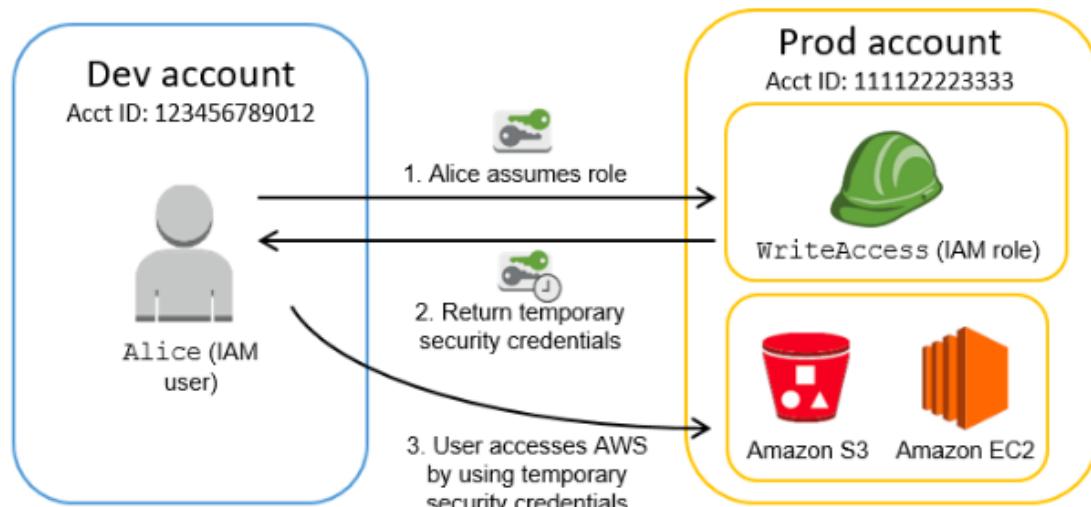
- Use AWS SSO
- All of the given options are correct.
- **Use AWS STS**
- Use AWS Cognito to issue JSON Web Tokens (JWT)

Correct

AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

1. Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
2. STS returns a set of temporary security credentials.
3. Alice uses the temporary security credentials to access services and resources in the Prod account. Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.



Using AWS Cognito to issue JSON Web Tokens (JWT) is incorrect because the Amazon Cognito service is primarily used for user authentication and not for providing access to your AWS resources. A JSON Web Token (JWT) is meant to be used for user authentication and session management.

Using AWS SSO is incorrect. Although the AWS SSO service uses STS, it does not issue short-lived credentials by itself. AWS Single Sign-On (SSO) is a cloud SSO service that makes it easy to centrally manage SSO access to multiple AWS accounts and business applications.

The option that says ***All of the above*** is incorrect as only STS has the ability to provide temporary security credentials.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

35. QUESTION

Category: CSAA – Design Resilient Architectures

A commercial bank has a forex trading application. They created an Auto Scaling group of EC2 instances that allow the bank to cope with the current traffic and achieve cost-efficiency. They want the Auto Scaling group to behave in such a way that it will follow a predefined set of parameters before it scales down the number of EC2 instances, which protects the system from unintended slowdown or unavailability.

Which of the following statements are true regarding the cooldown period? (Select TWO.)

- It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.
- It ensures that before the Auto Scaling group scales out, the EC2 instances have an ample time to cooldown.
- Its default value is 600 seconds.
- It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.
- Its default value is 300 seconds.

Correct

In Auto Scaling, the following statements are correct regarding the cooldown period:

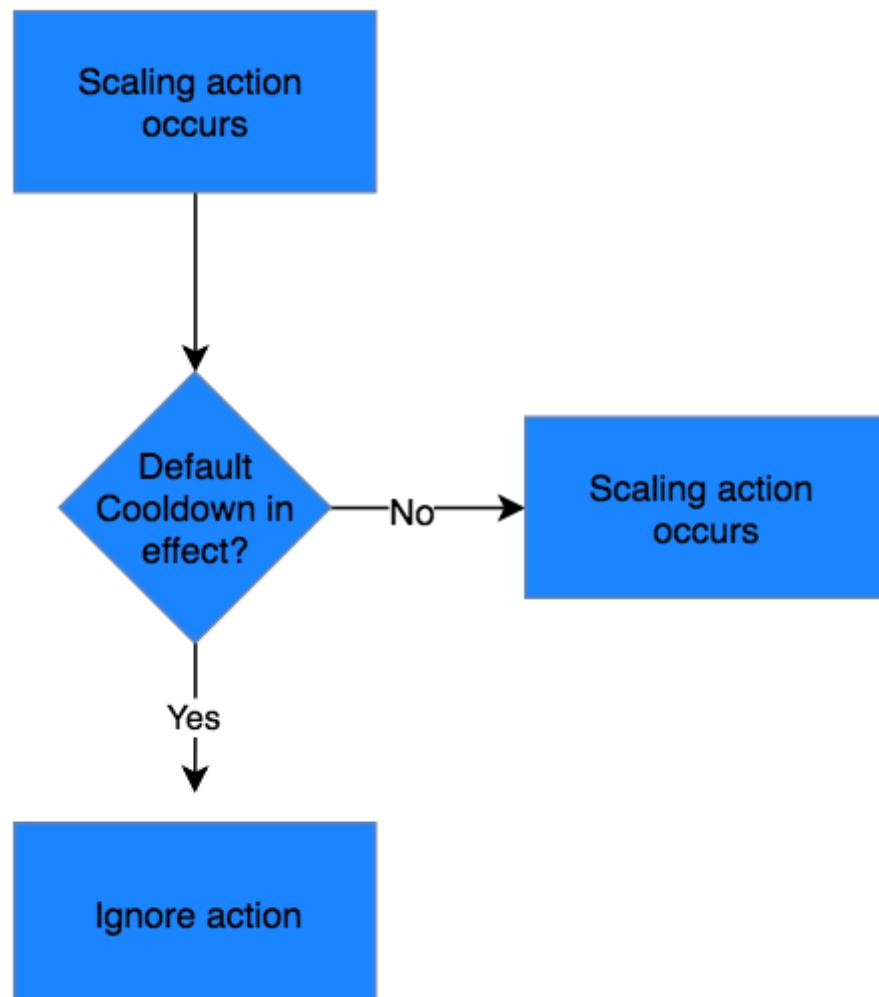
1. It ensures that the Auto Scaling group does not launch or terminate additional EC2 instances before the previous scaling activity takes effect.
2. Its default value is 300 seconds.
3. It is a configurable setting for your Auto Scaling group.

The following options are incorrect:

- *- ***It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.****
- *- ***It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.****
- *- ***Its default value is 600 seconds.****

These statements are inaccurate and don't depict what the word "cooldown" actually means for Auto Scaling. The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect. After the Auto Scaling group dynamically scales using a simple scaling policy, it waits for the cooldown period to complete before resuming scaling activities.

The figure below demonstrates the scaling cooldown:



Reference:

<http://docs.aws.amazon.com/autoscaling/latest/userguide/as-instance-termination.html>

Check out this AWS Auto Scaling Cheat Sheet:

<https://tutorialsdojo.com/aws-auto-scaling/>

36. QUESTION

Category: CSAA – Design High-Performing Architectures

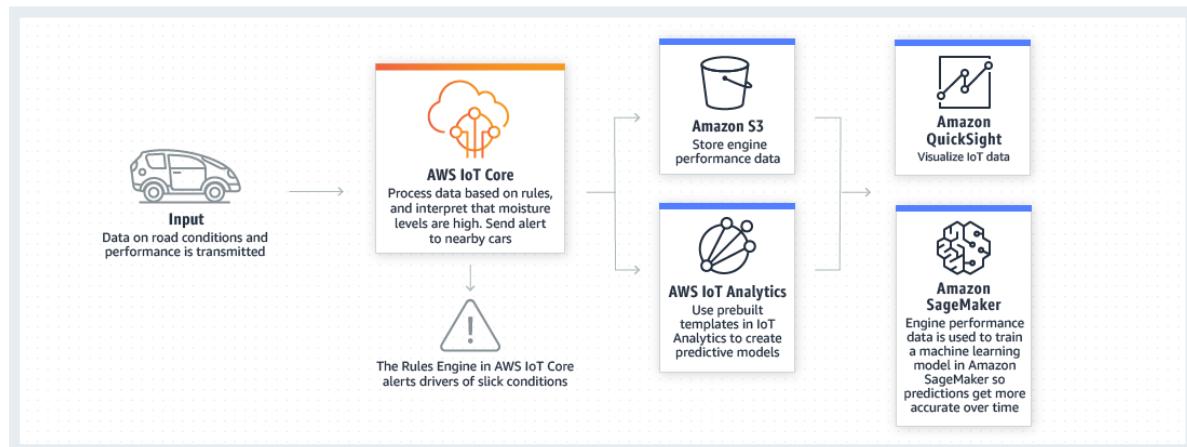
A company is working with a government agency to improve traffic planning and maintenance of roadways to prevent accidents. The proposed solution is to manage the traffic infrastructure in real-time, alert traffic engineers and emergency response teams when problems are detected, and automatically change traffic signals to get emergency personnel to accident scenes faster by using sensors and smart devices.

Which AWS service will allow the developers of the agency to connect the smart devices to the cloud-based applications?

- Amazon Elastic Container Service
- AWS Elastic Beanstalk
- **AWS IoT Core**
- AWS CloudFormation

Correct

AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices. AWS IoT Core provides secure communication and data processing across different kinds of connected devices and locations so you can easily build IoT applications.



AWS IoT Core allows you to connect multiple devices to the cloud and to other devices without requiring you to deploy or manage any servers. You can also filter, transform, and act upon device data on the fly based on the rules you define. With AWS IoT Core, your applications can keep track of and communicate with all of your devices, all the time, even when they aren't connected.

Hence, the correct answer is: ***AWS IoT Core.***

AWS CloudFormation is incorrect because this is mainly used for creating and managing the architecture and not for handling connected devices. You have to use AWS IoT Core instead.

AWS Elastic Beanstalk is incorrect because this is just an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, and other programming languages. Elastic Beanstalk can't be used to connect smart devices to cloud-based applications.

Amazon Elastic Container Service is incorrect because this is mainly used for creating and managing docker instances and not for handling devices.

References:

37. QUESTION

Category: CSAA – Design Resilient Architectures

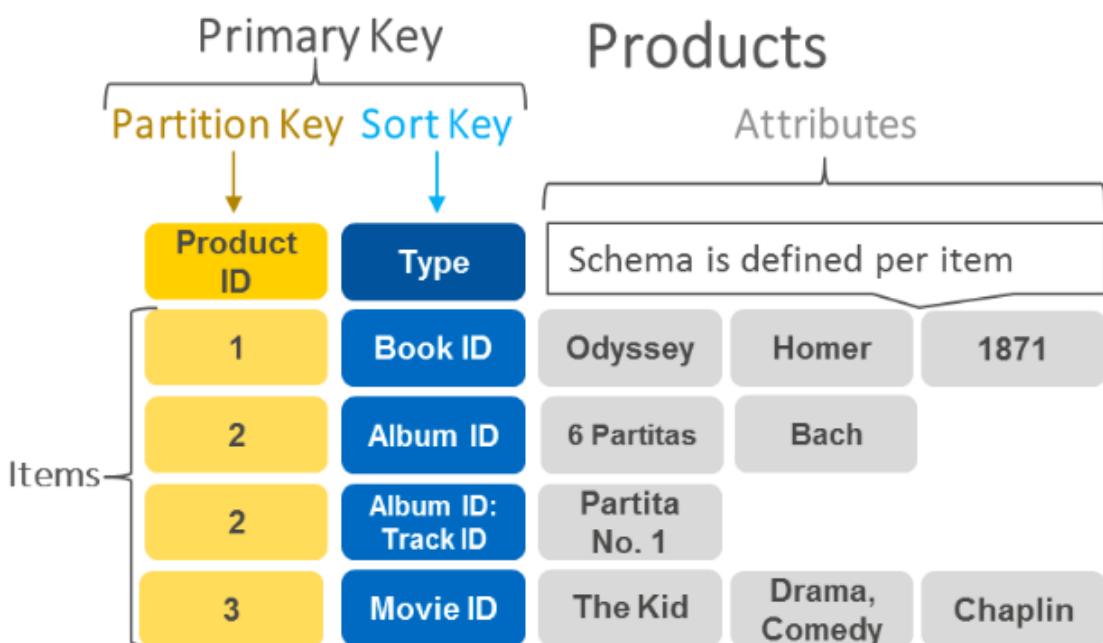
A music publishing company is building a multitier web application that requires a key-value store which will save the document models. Each model is composed of band ID, album ID, song ID, composer ID, lyrics, and other data. The web tier will be hosted in an Amazon ECS cluster with AWS Fargate launch type.

Which of the following is the MOST suitable setup for the database-tier?

- Launch an Amazon Aurora Serverless database.
- Use Amazon WorkDocs to store the document models.
- Launch a DynamoDB table.
- Launch an Amazon RDS database with Read Replicas.

Correct

Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. It is a fully managed cloud database and supports both document and key-value store models. Its flexible data model, reliable performance, and automatic scaling of throughput capacity makes it a great fit for mobile, web, gaming, ad tech, IoT, and many other applications.



Hence, the correct answer is: ***Launch a DynamoDB table.***

The option that says: ***Launch an Amazon RDS database with Read Replicas*** is incorrect because this is a relational database. This is not suitable to be used as a key-value store. A better option is to use DynamoDB as it supports both document and key-value store models.

The option that says: ***Use Amazon WorkDocs to store the document models*** is incorrect because Amazon WorkDocs simply enables you to share content, provide rich feedback, and collaboratively edit documents. It is not a key-value store like DynamoDB.

The option that says: ***Launch an Amazon Aurora Serverless database*** is incorrect because this type of database is not suitable to be used as a key-value store. Amazon Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora where the database will automatically start-up, shut down, and scale capacity up or down based on your application's needs. It enables you to run your database in the cloud without managing any database instances. It's a simple, cost-effective option for infrequent, intermittent, or unpredictable workloads and not as a key-value store.

References:

<https://aws.amazon.com/dynamodb/>

<https://aws.amazon.com/nosql/key-value/>

Check out this Amazon DynamoDB Cheat Sheet:

<https://tutorialsdojo.com/amazon-dynamodb/>

Amazon DynamoDB Overview:

<https://youtu.be/3ZOyUNleorU>

38. QUESTION

Category: CSAA – Design Secure Applications and Architectures

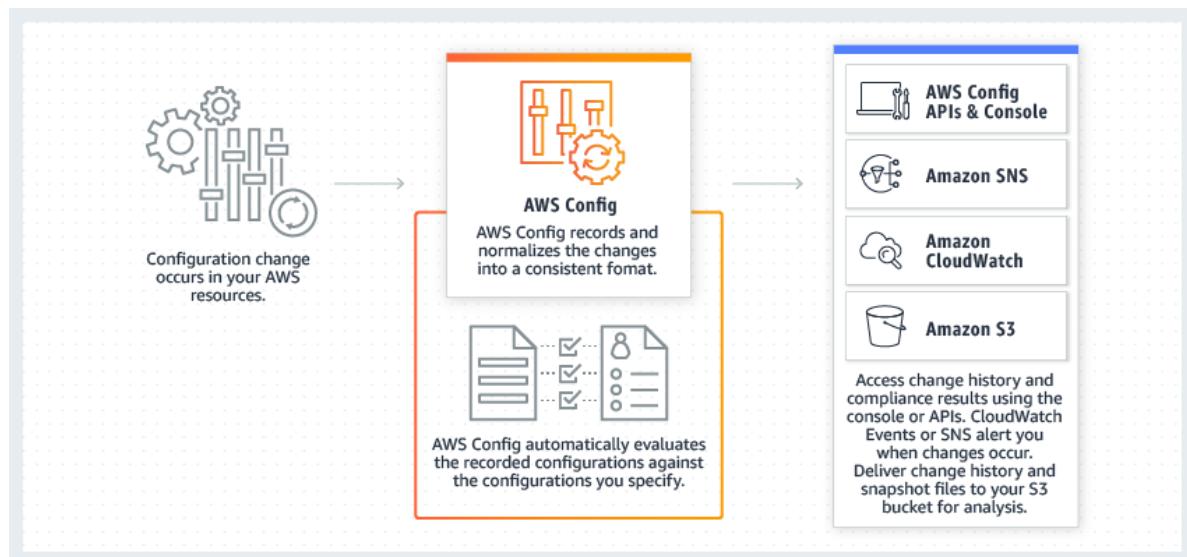
A company needs to assess and audit all the configurations in their AWS account. It must enforce strict compliance by tracking all configuration changes made to any of its Amazon S3 buckets. Publicly accessible S3 buckets should also be identified automatically to avoid data breaches.

Which of the following options will meet this requirement?

- Use AWS CloudTrail and review the event history of your AWS account.
- **Use AWS Config to set up a rule in your AWS account.**
- Use AWS Trusted Advisor to analyze your AWS environment.
- Use AWS IAM to generate a credential report.

Incorrect

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.



You can use AWS Config to evaluate the configuration settings of your AWS resources. By creating an AWS Config rule, you can enforce your ideal configuration in your AWS account. It also checks if the applied configuration in your resources violates any of the conditions in your rules. The AWS Config dashboard shows the compliance status of your rules and resources. You can verify if your resources comply with your desired configurations and learn which specific resources are noncompliant.

Hence, the correct answer is: ***Use AWS Config to set up a rule in your AWS account***.

The option that says: ***Use AWS Trusted Advisor to analyze your AWS environment*** is incorrect because AWS Trusted Advisor only provides best practice recommendations. It cannot define rules for your AWS resources.

The option that says: ***Use AWS IAM to generate a credential report*** is incorrect because this report will not help you evaluate resources. The IAM credential report is just a list of all IAM users in your AWS account.

The option that says: ***Use AWS CloudTrail and review the event history of your AWS account*** is incorrect. Although it can track changes and store a history of what happened to your resources, this service still cannot enforce rules to comply with your organization's policies.

References:

<https://aws.amazon.com/config/>

<https://docs.aws.amazon.com/config/latest/developerguide/evaluate-config.html>

Check out this AWS Config Cheat Sheet:

<https://tutorialsdojo.com/aws-config/>

39. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

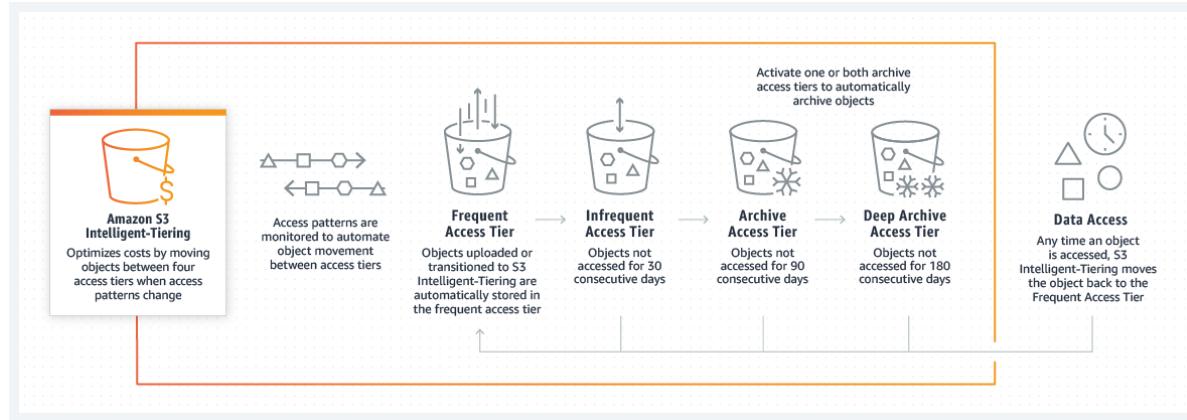
A company is building a transcription service in which a fleet of EC2 worker instances processes an uploaded audio file and generates a text file as an output. They must store both of these frequently accessed files in the same durable storage until the text file is retrieved by the uploader. Due to an expected surge in demand, they have to ensure that the storage is scalable and can be retrieved within minutes.

Which storage option in AWS can they use in this situation, which is both cost-efficient and scalable?

- Multiple instance stores
- **A single Amazon S3 bucket**
- Amazon S3 Glacier Deep Archive
- Multiple Amazon EBS volume with snapshots

Correct

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. It provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.



In this scenario, the requirement is to have cost-efficient and scalable storage. Among the given options, the best option is to use Amazon S3. It's a simple storage service that offers a highly-scalable, reliable, and low-latency data storage infrastructure at very low costs.

Hence, the correct answer is: ***A single Amazon S3 bucket.***

The option that says: ***Multiple Amazon EBS volume with snapshots*** is incorrect because Amazon S3 is more cost-efficient than EBS volumes.

The option that says: ***Multiple instance stores*** is incorrect. Just like the option above, you must use Amazon S3 since it is scalable and cost-efficient than instance store volumes.

The option that says: ***Amazon S3 Glacier Deep Archive*** is incorrect because this is mainly used for data archives with data retrieval times that can take more than 12 hours. Hence, it is not suitable for the transcription service where the data are stored and frequently accessed.

References:

<https://aws.amazon.com/s3/pricing/>

<https://docs.aws.amazon.com/AmazonS3/latest/gsg/GetStartedWithS3.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

40. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

Both historical records and frequently accessed data are stored on an on-premises storage system. The amount of current data is growing at an exponential rate. As the storage's capacity is nearing its limit, the company's Solutions Architect has decided to move the historical records to AWS to free up space for the active data.

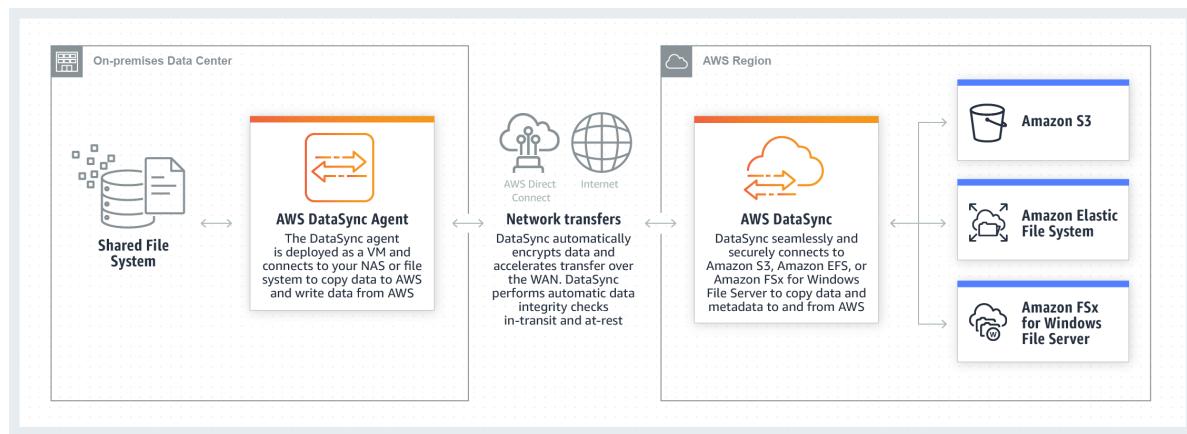
Which of the following architectures deliver the best solution in terms of cost and operational management?

- Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.
- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.
- Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Standard to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.
- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.

Correct

AWS DataSync makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3, Amazon Elastic File System (Amazon EFS), or Amazon FSx for Windows File Server. Manual tasks related to data transfers can slow down migrations and burden IT operations. DataSync eliminates or automatically handles many of these tasks, including scripting copy jobs, scheduling, and monitoring transfers, validating data, and optimizing network utilization. The DataSync software agent connects to your Network File System (NFS), Server Message Block (SMB) storage, and your self-managed object storage, so you don't have to modify your applications.

DataSync can transfer hundreds of terabytes and millions of files at speeds up to 10 times faster than open-source tools, over the Internet or AWS Direct Connect links. You can use DataSync to migrate active data sets or archives to AWS, transfer data to the cloud for timely analysis and processing, or replicate data to AWS for business continuity. Getting started with DataSync is easy: deploy the DataSync agent, connect it to your file system, select your AWS storage resources, and start moving data between them. You pay only for the data you move.



Since the problem is mainly about moving historical records from on-premises to AWS, using AWS DataSync is a more suitable solution. You can use DataSync to move cold data from expensive on-premises storage systems directly to durable and secure long-term storage, such as Amazon S3 Glacier or Amazon S3 Glacier Deep Archive.

Hence, the correct answer is the option that says: ***Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.***

The following options are both incorrect:

- *- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier Deep Archive to be the destination for the data.***
- *- Use AWS Storage Gateway to move the historical records from on-premises to AWS. Choose Amazon S3 Glacier to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days.***

Although you can copy data from on-premises to AWS with Storage Gateway, it is not suitable for transferring large sets of data to AWS. Storage Gateway is mainly used in providing low-latency access to data by caching frequently accessed data on-premises while storing archive data securely and durably in Amazon cloud storage services. Storage Gateway optimizes data transfer to AWS by sending only changed data and compressing data.

The option that says: ***Use AWS DataSync to move the historical records from on-premises to AWS. Choose Amazon S3 Standard to be the destination for the data. Modify the S3 lifecycle configuration to move the data from the Standard tier to Amazon S3 Glacier Deep Archive after 30 days*** is incorrect because, with AWS DataSync, you can transfer data from on-premises directly to Amazon S3 Glacier Deep Archive. You don't have to configure the S3 lifecycle policy and wait for 30 days to move the data to Glacier Deep Archive.

References:

<https://aws.amazon.com/datasync/faqs/>

<https://aws.amazon.com/storagegateway/faqs/>

Check out these AWS DataSync and Storage Gateway Cheat Sheets:

<https://tutorialsdojo.com/aws-datasync/>

<https://tutorialsdojo.com/aws-storage-gateway/>

AWS Storage Gateway vs DataSync:

<https://youtu.be/tmfe1rO-AUs>

41. QUESTION

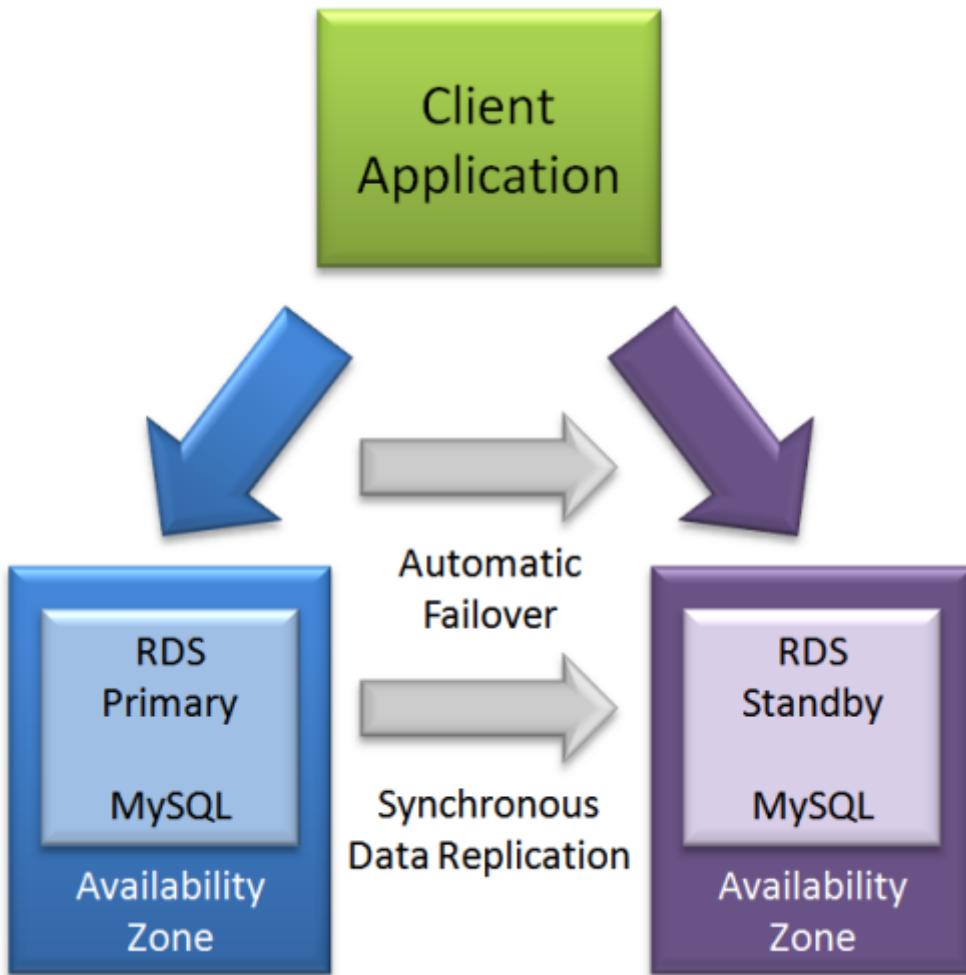
Category: CSAA – Design Resilient Architectures

An accounting application uses an RDS database configured with Multi-AZ deployments to improve availability. What would happen to RDS if the primary database instance fails?

- The canonical name record (CNAME) is switched from the primary to standby instance.
- A new database instance is created in the standby Availability Zone.
- The primary database instance will reboot.
- The IP address of the primary DB instance is switched to the standby DB instance.

Correct

In **Amazon RDS**, failover is automatically handled so that you can resume database operations as quickly as possible without administrative intervention in the event that your primary database instance went down. When failing over, Amazon RDS simply flips the canonical name record (CNAME) for your DB instance to point at the standby, which is in turn promoted to become the new primary.



The option that says: ***The IP address of the primary DB instance is switched to the standby DB instance*** is incorrect since IP addresses are per subnet, and subnets cannot span multiple AZs.

The option that says: ***The primary database instance will reboot*** is incorrect since in the event of a failure, there is no database to reboot with.

The option that says: ***A new database instance is created in the standby Availability Zone*** is incorrect since with multi-AZ enabled, you already have a standby database in another AZ.

References:

<https://aws.amazon.com/rds/details/multi-az/>

<https://aws.amazon.com/rds/faqs/>

Amazon RDS Overview:

<https://www.youtube.com/embed/aZmpLI8K1UU>

Check out this Amazon RDS Cheat Sheet:

<https://tutorialsdojo.com/amazon-relational-database-service-amazon-rds/>

42. QUESTION

Category: CSAA – Design Resilient Architectures

A company plans to migrate all of their applications to AWS. The Solutions Architect suggested to store all the data to EBS volumes. The Chief Technical Officer is worried that EBS volumes are not appropriate for the existing workloads due to compliance requirements, downtime scenarios, and IOPS performance.

Which of the following are valid points in proving that EBS is the best service to use for migration? (Select TWO.)

- When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component.
- EBS volumes can be attached to any EC2 Instance in any Availability Zone.
- **EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.**
- **An EBS volume is off-instance storage that can persist independently from the life of an instance.**
- Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones

Correct

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Here is a list of important information about EBS Volumes:

- When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single hardware component.
- After you create a volume, you can attach it to any EC2 instance in the same Availability Zone
- Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1) volume to multiple Nitro-based instances that are in the same Availability Zone. However, other EBS types are not supported.
- An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.
- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.
- Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)
- EBS Volumes offer 99.999% SLA.

The option that says: ***When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component*** is incorrect because when you create an EBS volume in an Availability Zone, it is automatically replicated within that zone only, and not on a separate AWS region, to prevent data loss due to a failure of any single hardware component.

The option that says: ***EBS volumes can be attached to any EC2 Instance in any Availability Zone*** is incorrect as EBS volumes can only be attached to an EC2 instance in the same Availability Zone.

The option that says: ***Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones*** is almost correct. But instead of storing the volume to Amazon RDS, the EBS Volume snapshots are actually sent to Amazon S3.

References:

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

<https://aws.amazon.com/ebs/features/>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/>

Here is a short video tutorial on EBS:

<https://youtu.be/ljYH5IHQdxo>

43. QUESTION

Category: CSAA – Design Resilient Architectures

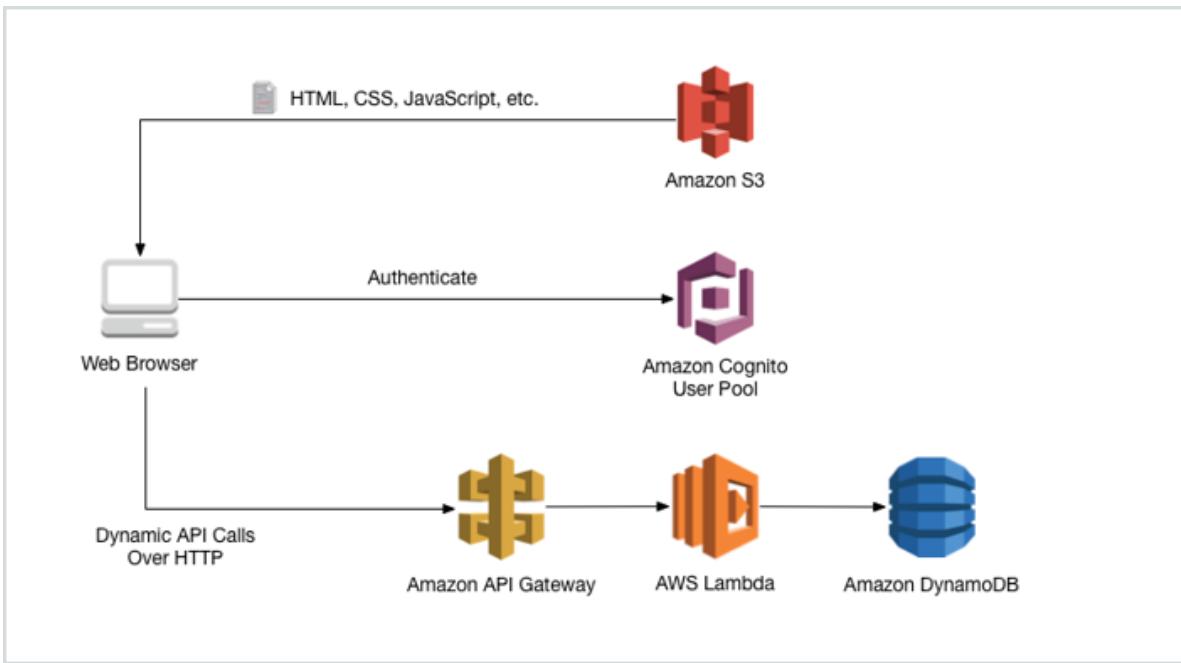
A Solutions Architect of a multinational gaming company develops video games for PS4, Xbox One, and Nintendo Switch consoles, plus a number of mobile games for Android and iOS. Due to the wide range of their products and services, the architect proposed that they use API Gateway.

What are the key features of API Gateway that the architect can tell to the client? (Select TWO.)

- You pay only for the API calls you receive and the amount of data transferred out.
- Enables you to run applications requiring high levels of inter-node communications at scale on AWS through its custom-built operating system (OS) bypass hardware interface.
- Enables you to build RESTful APIs and WebSocket APIs that are optimized for serverless workloads.
- Provides you with static anycast IP addresses that serve as a fixed entry point to your applications hosted in one or more AWS Regions.
- It automatically provides a query language for your APIs similar to GraphQL.

Correct

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services, such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any web application. Since it can use AWS Lambda, you can run your APIs without servers.



Amazon API Gateway handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management. Amazon API Gateway has no minimum fees or startup costs. You pay only for the API calls you receive and the amount of data transferred out.

Hence, the correct answers are:

- Enables you to build RESTful APIs and WebSocket APIs that are optimized for serverless workloads

- You pay only for the API calls you receive and the amount of data transferred out.

The option that says: ***It automatically provides a query language for your APIs similar to GraphQL*** is incorrect because this is not provided by API Gateway.

The option that says: ***Provides you with static anycast IP addresses that serve as a fixed entry point to your applications hosted in one or more AWS Regions*** is incorrect because this is a capability of AWS Global Accelerator and not API Gateway.

The option that says: ***Enables you to run applications requiring high levels of inter-node communications at scale on AWS through its custom-built operating system (OS) bypass hardware interface*** is incorrect because this is a capability of Elastic Fabric Adapter and not API Gateway.

References:

<https://aws.amazon.com/api-gateway/>

<https://aws.amazon.com/api-gateway/features/>

Check out this Amazon API Gateway Cheat Sheet:

<https://tutorialsdojo.com/amazon-api-gateway/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

44. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company recently adopted a hybrid architecture that integrates its on-premises data center to AWS cloud. You are assigned to configure the VPC and implement the required IAM users, IAM roles, IAM groups, and IAM policies.

In this scenario, what is the best practice when creating IAM policies?

- Use the principle of least privilege which means granting only the least number of people with full root access.
- Determine what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations.
- Grant all permissions to any EC2 user.
- Use the principle of least privilege which means granting only the permissions required to perform a task.

Correct

One of the best practices in AWS IAM is to **grant least privilege**.

When you create IAM policies, follow the standard security advice of granting *least privilege*—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform *only* those tasks.

Therefore, ***using the principle of least privilege which means granting only the permissions required to perform a task*** is the correct answer.

Start with a minimum set of permissions and grant additional permissions as necessary. Defining the right set of permissions requires some understanding of the user's objectives. Determine what is required for the specific task, what actions a particular service supports, and what permissions are required in order to perform those actions.

Granting all permissions to any EC2 user is incorrect since you don't want your users to gain access to everything and perform unnecessary actions. Doing so is not a good security practice.

Using the principle of least privilege which means granting only the least number of people with full root access is incorrect because this is not the correct definition of what the principle of least privilege is.

Determining what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations is incorrect since there are some users who you should not give administrative access to. You should follow the principle of least privilege when providing permissions and accesses to your resources.

Reference:

<https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#use-groups-for-permissions>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

Service Control Policies (SCP) vs IAM Policies:

<https://tutorialsdojo.com/service-control-policies-scp-vs-iam-policies/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

45. QUESTION

Category: CSAA – Design Secure Applications and Architectures

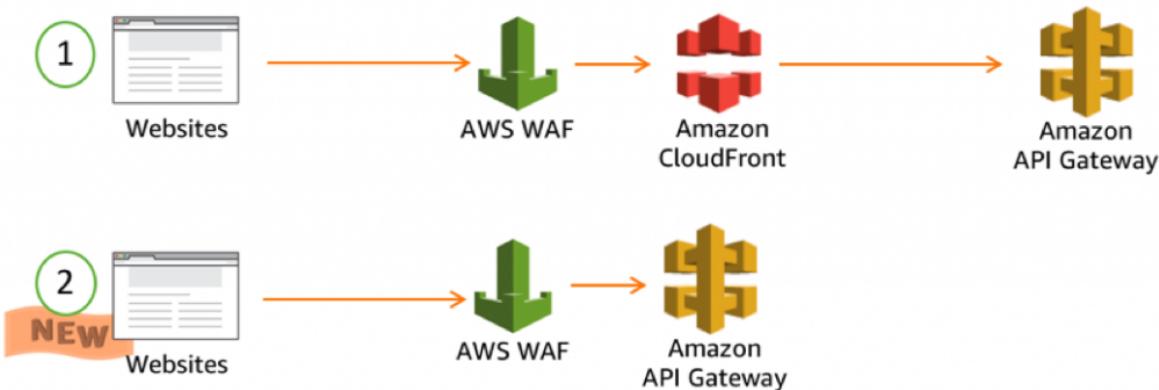
A company is hosting its web application in an Auto Scaling group of EC2 instances behind an Application Load Balancer. Recently, the Solutions Architect identified a series of SQL injection attempts and cross-site scripting attacks to the application, which had adversely affected their production data.

Which of the following should the Architect implement to mitigate this kind of attack?

- Using AWS Firewall Manager, set up security rules that block SQL injection and cross-site scripting attacks. Associate the rules to the Application Load Balancer.
- Use Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application.
- Block all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List.
- Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.**

Correct

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.



At the simplest level, AWS WAF lets you choose one of the following behaviors:

Allow all requests except the ones that you specify – This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.

Block all requests except the ones that you specify – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.

Count the requests that match the properties that you specify – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

Hence, the correct answer in this scenario is: ***Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.***

Using Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application is incorrect because Amazon GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

Using AWS Firewall Manager to set up security rules that block SQL injection and cross-site scripting attacks, then associating the rules to the Application Load Balancer is incorrect because AWS Firewall Manager just simplifies your AWS WAF and AWS Shield Advanced administration and maintenance tasks across multiple accounts and resources.

Blocking all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List is incorrect because this is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACLs are not effective in blocking SQL injection and cross-site scripting attacks

References:

<https://aws.amazon.com/waf/>

<https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html>

Check out this AWS WAF Cheat Sheet:

<https://tutorialsdojo.com/aws-waf/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

46. QUESTION

Category: CSAA – Design Secure Applications and Architectures

For data privacy, a healthcare company has been asked to comply with the Health Insurance Portability and Accountability Act (HIPAA). The company stores all its backups on an Amazon S3 bucket. It is required that data stored on the S3 bucket must be encrypted.

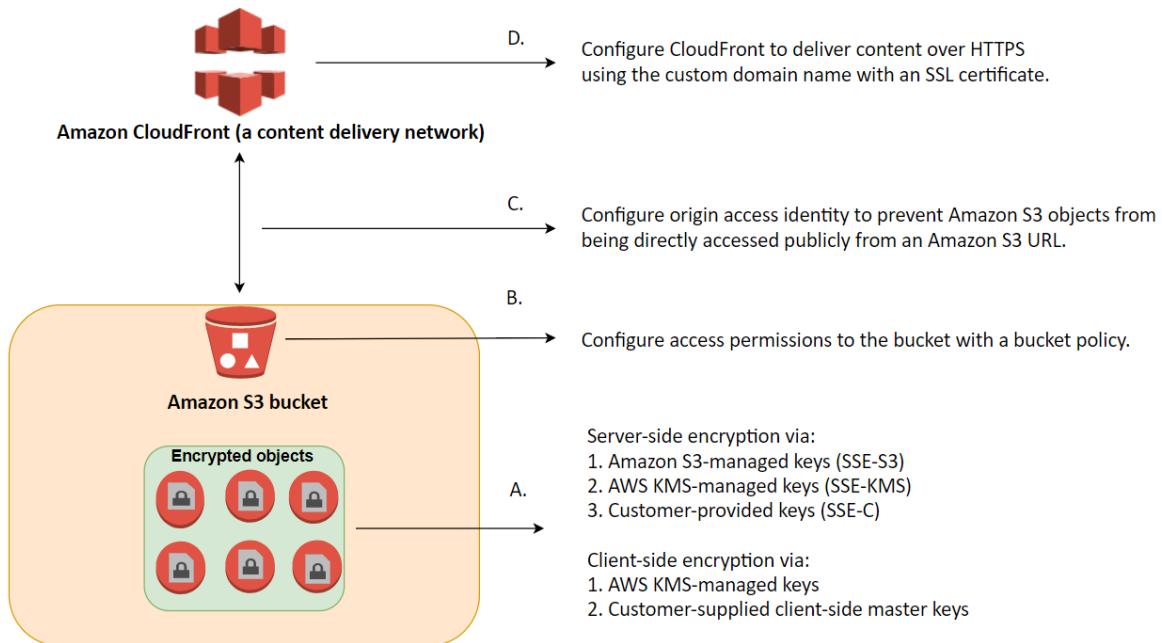
What is the best option to do this? (Select TWO.)

- Store the data on EBS volumes with encryption enabled instead of using Amazon S3.
- Enable Server-Side Encryption on an S3 bucket to make use of AES-128 encryption.

- Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys.
- Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption.
- Store the data in encrypted EBS snapshots.

Correct

Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a pre-signed URL, that URL works the same way for both encrypted and unencrypted objects.



You have three mutually exclusive options depending on how you choose to manage the encryption keys:

1. Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
2. Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
3. Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

The options that say: ***Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys*** and ***Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption*** are correct because these options are using client-side encryption and Amazon S3-Managed Keys (SSE-S3) respectively. **Client-side encryption** is the act of encrypting data before sending it to Amazon S3 while SSE-S3 uses AES-256 encryption.

Storing the data on EBS volumes with encryption enabled instead of using Amazon S3 and ***storing the data in encrypted EBS snapshots*** are incorrect because both options use EBS encryption and not S3.

Enabling Server-Side Encryption on an S3 bucket to make use of AES-128 encryption is incorrect as S3 doesn't provide AES-128 encryption, only AES-256.

References:

<http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html>

<https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

47. QUESTION

Category: CSAA – Design High-Performing Architectures

A software company has resources hosted in AWS and on-premises servers. You have been requested to create a decoupled architecture for applications which make use of both resources.

Which of the following options are valid? (Select TWO.)

- Use SQS to utilize both on-premises servers and EC2 instances for your decoupled application
- Use DynamoDB to utilize both on-premises servers and EC2 instances for your decoupled application
- Use VPC peering to connect both on-premises servers and EC2 instances for your decoupled application
- Use RDS to utilize both on-premises servers and EC2 instances for your decoupled application
- Use SWF to utilize both on-premises servers and EC2 instances for your decoupled application

Incorrect

Amazon Simple Queue Service (SQS) and **Amazon Simple Workflow Service (SWF)** are the services that you can use for creating a decoupled architecture in AWS. Decoupled architecture is a type of computing architecture that enables computing components or layers to execute independently while still interfacing with each other.

Amazon SQS offers reliable, highly-scalable hosted queues for storing messages while they travel between applications or microservices. Amazon SQS lets you move data between distributed application components and helps you decouple these components. Amazon SWF is a web service that makes it easy to coordinate work across distributed application components.

Using RDS to utilize both on-premises servers and EC2 instances for your decoupled application and ***using DynamoDB to utilize both on-premises servers and EC2 instances for your decoupled application*** are incorrect as RDS and DynamoDB are database services.

Using VPC peering to connect both on-premises servers and EC2 instances for your decoupled application is incorrect because you can't create a VPC peering for your on-premises network and AWS VPC.

References:

<https://aws.amazon.com/sqs/>

<http://docs.aws.amazon.com/amazonswf/latest/developerguide/swf-welcome.html>

Check out this Amazon SQS Cheat Sheet:

<https://tutorialsdojo.com/amazon-sqs/>

Amazon Simple Workflow (SWF) vs AWS Step Functions vs Amazon SQS:

<https://tutorialsdojo.com/amazon-simple-workflow-swf-vs-aws-step-functions-vs-amazon-sqs/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

48. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A media company hosts large volumes of archive data that are about 250 TB in size on their internal servers. They have decided to move these data to S3 because of its durability and redundancy. The company currently has a 100 Mbps dedicated line connecting their head office to the Internet.

Which of the following is the FASTEST and the MOST cost-effective way to import all these data to Amazon S3?

- Upload it directly to S3
- Order multiple AWS Snowball devices to upload the files to Amazon S3.
- Establish an AWS Direct Connect connection then transfer the data over to S3.
- Use AWS Snowmobile to transfer the data over to S3.

Incorrect

AWS Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of the AWS cloud. Using Snowball addresses common challenges with large-scale data transfers including high network costs, long transfer times, and security concerns. Transferring data with Snowball is simple, fast, secure, and can be as little as one-fifth the cost of high-speed Internet.



Snowball is a strong choice for data transfer if you need to more securely and quickly transfer terabytes to many petabytes of data to AWS. Snowball can also be the right choice if you don't want to make expensive upgrades to your network infrastructure, if you frequently experience large backlogs of data, if you're located in a physically isolated environment, or if you're in an area where high-speed Internet connections are not available or cost-prohibitive.

As a rule of thumb, if it takes more than one week to upload your data to AWS using the spare capacity of your existing Internet connection, then you should consider using Snowball. For example, if you have a 100 Mb connection that you can solely dedicate to transferring your data and need to transfer 100 TB of data, it takes more than 100 days to complete data transfer over that connection. You can make the same transfer by using multiple Snowballs in about a week.

Available Internet Connection	Theoretical Min. Number of Days to Transfer 100TB at 80% Network Utilization	When to Consider AWS Snowball?
T3 (44.736Mbps)	269 days	2TB or more
100Mbps	120 days	5TB or more
1000Mbps	12 days	60TB or more

Hence, ***ordering multiple AWS Snowball devices to upload the files to Amazon S3*** is the correct answer.

Uploading it directly to S3 is incorrect since this would take too long to finish due to the slow Internet connection of the company.

Establishing an AWS Direct Connect connection then transferring the data over to S3 is incorrect since provisioning a line for Direct Connect would take too much time and might not give you the fastest data transfer solution. In addition, the scenario didn't warrant an establishment of a dedicated connection from your on-premises data center to AWS. The primary goal is to just do a one-time migration of data to AWS which can be accomplished by using AWS Snowball devices.

Using AWS Snowmobile to transfer the data over to S3 is incorrect because Snowmobile is more suitable if you need to move extremely large amounts of data to AWS or need to transfer up to 100PB of data. This will be transported on a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. Take note that you only need to migrate 250 TB of data, hence, this is not the most suitable and cost-effective solution.

References:

<https://aws.amazon.com/snowball/>

<https://aws.amazon.com/snowball/faqs/>

Check out this AWS Snowball Cheat Sheet:

<https://tutorialsdojo.com/aws-snowball/>

S3 Transfer Acceleration vs Direct Connect vs VPN vs Snowball vs Snowmobile:

<https://tutorialsdojo.com/s3-transfer-acceleration-vs-direct-connect-vs-vpn-vs-snowball-vs-snowmobile/>

Comparison of AWS Services Cheat Sheets:

<https://tutorialsdojo.com/comparison-of-aws-services/>

49. QUESTION

Category: CSAA – Design Resilient Architectures

An advertising company is currently working on a proof of concept project that automatically provides SEO analytics for its clients. Your company has a VPC in AWS that operates in a dual-stack mode in which IPv4 and IPv6 communication is allowed. You deployed the application to an Auto Scaling group of EC2 instances with an Application Load Balancer in front that evenly distributes the incoming traffic. You are ready to go live but you need to point your domain name (tutorialsdojo.com) to the Application Load Balancer.

In Route 53, which record types will you use to point the DNS name of the Application Load Balancer? (Select TWO.)

- Alias with a type of "MX" record set
- **Alias with a type "A" record set**
- Alias with a type "CNAME" record set
- **Alias with a type "AAAA" record set**
- Non-Alias with a type "A" record set

Incorrect

The correct answers are: *Alias with a type "AAAA" record set* and *Alias with a type "A" record set.*

To route domain traffic to an ELB load balancer, use Amazon Route 53 to create an alias record that points to your load balancer. An alias record is a Route 53 extension to DNS. It's similar to a CNAME record, but you can create an alias record both for the root domain, such as tutorialsdojo.com, and for subdomains, such as portal.tutorialsdojo.com. (You can create CNAME records only for subdomains.) To enable IPv6 resolution, you would need to create a second resource record, tutorialsdojo.com ALIAS AAAA -> myelb.us-west-2.elb.amazonaws.com, this is assuming your Elastic Load Balancer has IPv6 support.

Create Record Set

Name: tutorialsdojo.com.

Type: AAAA – IPv6 address

Alias: Yes No

Alias Target: dualstack.tutor-Appl-1ICKV12Q66A

Alias Hosted Zone ID: KTTL2X6KTTL2

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-2.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com
- VPC endpoint: example.us-east-2.vpce.amazonaws.com
- API Gateway custom regional API: d-abcde12345.execute-api.us-west-2.amazonaws.com

[Learn More](#)

Routing Policy: Simple

Route 53 responds to queries based only on the values in this record.

[Learn More](#)

Evaluate Target Health: Yes No

Non-Alias with a type "A" record set is incorrect because you only use Non-Alias with a type "A" record set for IP addresses.

Alias with a type “CNAME” record set is incorrect because you can't create a CNAME record at the zone apex. For example, if you register the DNS name tutorialsdojo.com, the zone apex is tutorialsdojo.com.

Alias with a type of “MX” record set is incorrect because an MX record is primarily used for mail servers. It includes a priority number and a domain name, for example: 10
mailserver.tutorialsdojo.com .

Reference:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-to-elb-load-balancer.html>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/resource-record-sets-choosing-alias-non-alias.html>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

50. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An insurance company utilizes SAP HANA for its day-to-day ERP operations. Since they can't migrate this database due to customer preferences, they need to integrate it with the current AWS workload in the VPC in which they are required to establish a site-to-site VPN connection.

What needs to be configured outside of the VPC for them to have a successful site-to-site VPN connection?

- The main route table in your VPC to route traffic through a NAT instance
- A dedicated NAT instance in a public subnet
- An Internet-routable IP address (static) of the customer gateway's external interface for the on-premises network
- An EIP to the Virtual Private Gateway

Correct

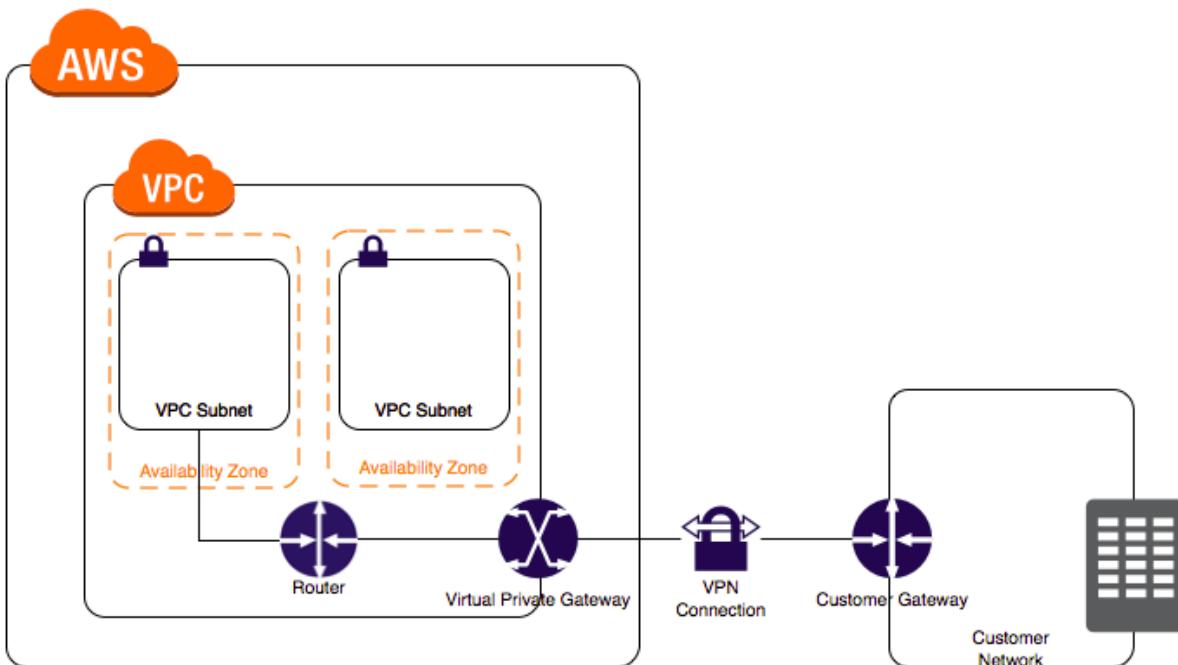
By default, instances that you launch into a virtual private cloud (VPC) can't communicate with your own network. You can enable access to your network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS managed VPN connection.

Although the term **VPN connection** is a general term, in the Amazon VPC documentation, a VPN connection refers to the connection between your VPC and your own network. AWS supports Internet Protocol security (IPsec) VPN connections.

A **customer gateway** is a physical device or software application on your side of the VPN connection.

To create a VPN connection, you must create a customer gateway resource in AWS, which provides information to AWS about your customer gateway device. Next, you have to set up an Internet-routable IP address (static) of the customer gateway's external interface.

The following diagram illustrates single VPN connections. The VPC has an attached virtual private gateway, and your remote network includes a customer gateway, which you must configure to enable the VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.



The options that say: ***A dedicated NAT instance in a public subnet*** and ***the main route table in your VPC to route traffic through a NAT instance*** are incorrect since you don't need a NAT instance for you to be able to create a VPN connection.

An EIP to the Virtual Private Gateway is incorrect since you do not attach an EIP to a VPG.

References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html

<https://docs.aws.amazon.com/vpc/latest/userguide/SetUpVPNConnections.html>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

51. QUESTION

Category: CSAA – Design Resilient Architectures

One member of your DevOps team consulted you about a connectivity problem in one of your Amazon EC2 instances. The application architecture is initially set up with four EC2 instances, each with an EIP address that all belong to a public non-default subnet. You launched another instance to handle the increasing workload of your application. The EC2 instances also belong to the same security group. Everything works well as expected except for one of the EC2 instances which is not able to send nor receive traffic over the Internet.

Which of the following is the MOST likely reason for this issue?

- The EC2 instance is running in an Availability Zone that is not connected to an Internet gateway.
- **The EC2 instance does not have a public IP address associated with it.**

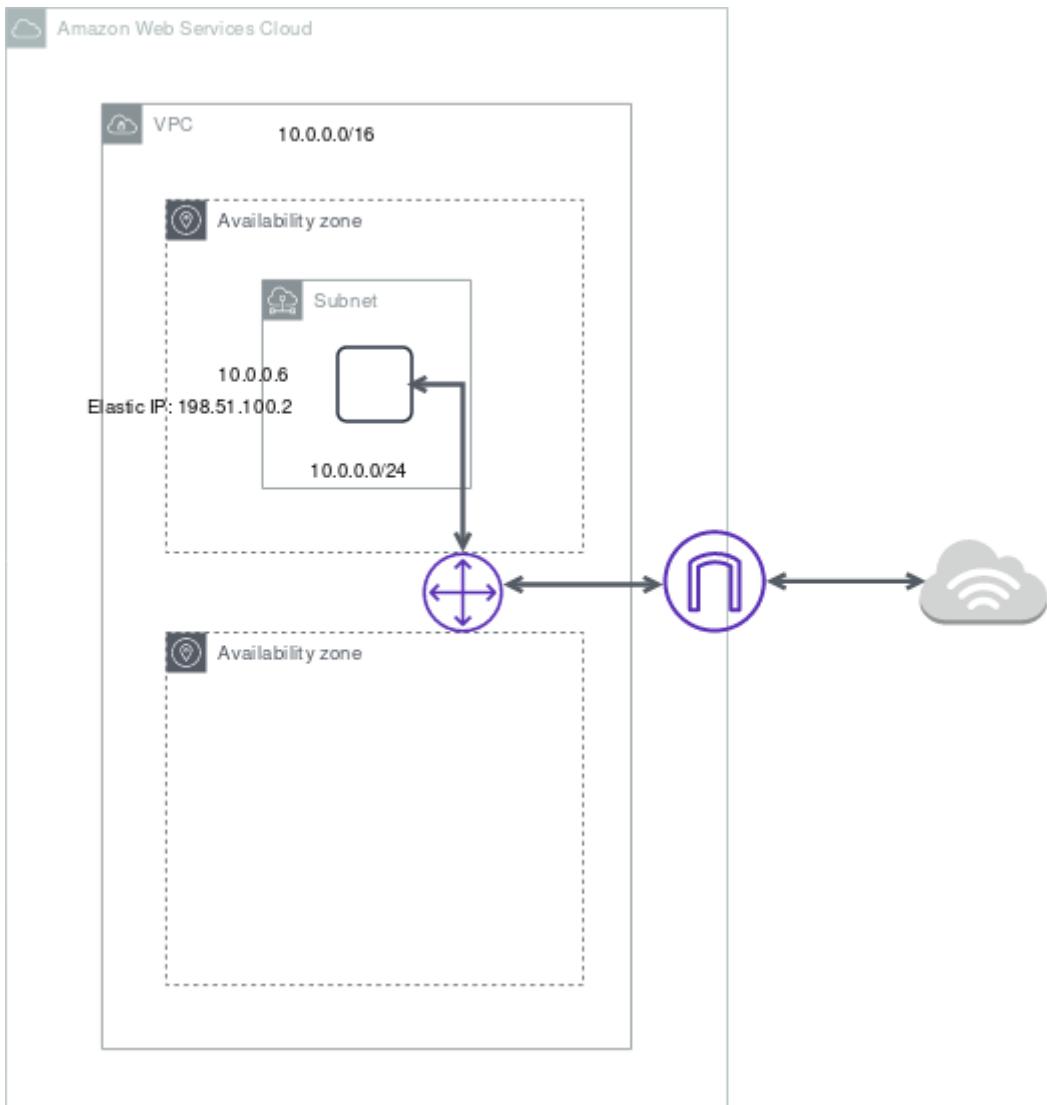
- The route table is not properly configured to allow traffic to and from the Internet through the Internet gateway.
- The EC2 instance does not have a private IP address associated with it.

Incorrect

IP addresses enable resources in your VPC to communicate with each other, and with resources over the Internet. Amazon EC2 and Amazon VPC support the IPv4 and IPv6 addressing protocols.

By default, Amazon EC2 and Amazon VPC use the IPv4 addressing protocol. When you create a VPC, you must assign it an IPv4 CIDR block (a range of private IPv4 addresses). Private IPv4 addresses are not reachable over the Internet. To connect to your instance over the Internet, or to enable communication between your instances and other AWS services that have public endpoints, you can assign a globally-unique public IPv4 address to your instance.

You can optionally associate an IPv6 CIDR block with your VPC and subnets, and assign IPv6 addresses from that block to the resources in your VPC. IPv6 addresses are public and reachable over the Internet.



All subnets have a modifiable attribute that determines whether a network interface created in that subnet is assigned a public IPv4 address and, if applicable, an IPv6 address. This includes the primary network interface (eth0) that's created for an instance when you launch an instance in that subnet. Regardless of the subnet attribute, you can still override this setting for a specific instance during launch.

By default, nondefault subnets have the IPv4 public addressing attribute set to `false`, and default subnets have this attribute set to `true`. An exception is a nondefault subnet created by the Amazon EC2 launch instance wizard — the wizard sets the attribute to `true`. You can modify this attribute using the Amazon VPC console.

In this scenario, there are 5 EC2 instances that belong to the same security group that should be able to connect to the Internet. The main route table is properly configured but there is a problem connecting to one instance. Since the other four instances are working fine, we can assume that the security group and the route table are correctly configured. One possible reason for this issue is that the problematic instance does not have a public or an EIP address.

Take note as well that the four EC2 instances all belong to a public **non-default** subnet. Which means that a new EC2 instance will not have a public IP address by default since the since IPv4 public addressing attribute is initially set to `false`.

Hence, the correct answer is the option that says: ***The EC2 instance does not have a public IP address associated with it.***

The option that says: ***The route table is not properly configured to allow traffic to and from the Internet through the Internet gateway*** is incorrect because the other three instances, which are associated with the same route table and security group, do not have any issues.

The option that says: ***The EC2 instance is running in an Availability Zone that is not connected to an Internet gateway*** is incorrect because there is no relationship between the Availability Zone and the Internet Gateway (IGW) that may have caused the issue.

References:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Scenario1.html

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html#vpc-ip-addressing-subnet>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

52. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company plans to conduct a network security audit. The web application is hosted on an Auto Scaling group of EC2 Instances with an Application Load Balancer in front to evenly distribute the incoming traffic. A Solutions Architect has been tasked to enhance the security posture of the company's cloud infrastructure and minimize the impact of DDoS attacks on its resources.

Which of the following is the most effective solution that should be implemented?

- Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification.
- Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront.**
- Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that

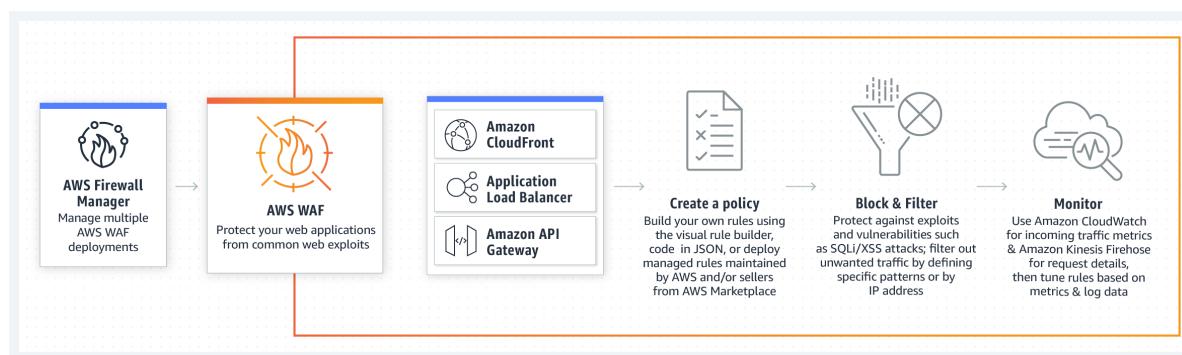
processes the flow logs and invokes Amazon SNS for notification.

- Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification.

Incorrect

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define. You can deploy AWS WAF on Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on EC2, or Amazon API Gateway for your APIs.

To detect and mitigate DDoS attacks, you can use **AWS WAF** in addition to AWS Shield. AWS WAF is a web application firewall that helps detect and mitigate web application layer DDoS attacks by inspecting traffic inline. Application layer DDoS attacks use well-formed but malicious requests to evade mitigation and consume application resources. You can define custom security rules that contain a set of conditions, rules, and actions to block attacking traffic. After you define web ACLs, you can apply them to CloudFront distributions, and web ACLs are evaluated in the priority order you specified when you configured them.



By using AWS WAF, you can configure web access control lists (Web ACLs) on your CloudFront distributions or Application Load Balancers to filter and block requests based on request signatures. Each Web ACL consists of rules that you can configure to string match or regex match one or more request attributes, such as the URI, query-string, HTTP method, or header key. In addition, by using AWS WAF's rate-based rules, you can automatically block the IP addresses of bad actors when requests matching a rule exceed a threshold that you define. Requests from offending client IP addresses will receive 403 Forbidden error responses and will remain blocked until request rates drop below the threshold. This is useful for mitigating HTTP flood attacks that are disguised as regular web traffic.

It is recommended that you add web ACLs with rate-based rules as part of your AWS Shield Advanced protection. These rules can alert you to sudden spikes in traffic that might indicate a potential DDoS event. A rate-based rule counts the requests that arrive from any individual address in any five-minute period. If the number of requests exceeds the limit that you define, the rule can trigger an action such as sending you a notification.

Hence, the correct answer is: ***Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront.***

The option that says: ***Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that processes the flow logs and invokes Amazon SNS for notification*** is incorrect because this option only allows you to monitor the traffic that is reaching your instance. You can't use VPC Flow Logs to

mitigate DDoS attacks.

The option that says: ***Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification*** is incorrect. To deny suspicious addresses, you must manually insert the IP addresses of these hosts. This is a manual task which is not a sustainable solution. Take note that attackers generate large volumes of packets or requests to overwhelm the target system. Using a security group in this scenario won't help you mitigate DDoS attacks.

The option that says: ***Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification*** is incorrect because Amazon GuardDuty is just a threat detection service. You should use AWS WAF and create your own AWS WAF rate-based rules for mitigating HTTP flood attacks that are disguised as regular web traffic.

References:

<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>

<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-get-started-rate-based-rules.html>

https://d0.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

Check out this AWS WAF Cheat Sheet:

<https://tutorialsdojo.com/aws-waf/>

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

<https://youtu.be/-1S-RdeAmMo>

53. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A media company has two VPCs: VPC-1 and VPC-2 with peering connection between each other. VPC-1 only contains private subnets while VPC-2 only contains public subnets. The company uses a single AWS Direct Connect connection and a virtual interface to connect their on-premises network with VPC-1.

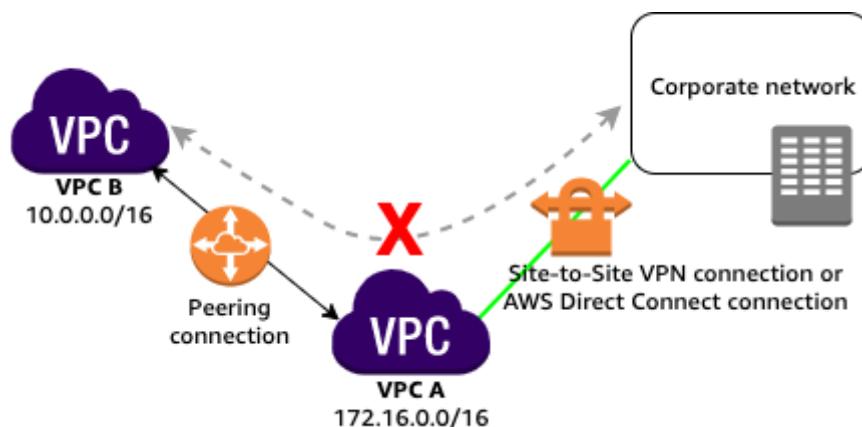
Which of the following options increase the fault tolerance of the connection to VPC-1? (Select TWO.)

- Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.
- Establish a hardware VPN over the Internet between VPC-1 and the on-premises network.
- Establish another AWS Direct Connect connection and private virtual interface in the same AWS region as VPC-1.
- Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.
- Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

Correct

In this scenario, you have two VPCs which have peering connections with each other. Note that a VPC peering connection does not support edge to edge routing. This means that if either VPC in a peering relationship has one of the following connections, you cannot extend the peering relationship to that connection:

- A VPN connection or an AWS Direct Connect connection to a corporate network
- An Internet connection through an Internet gateway
- An Internet connection in a private subnet through a NAT device
- A gateway VPC endpoint to an AWS service; for example, an endpoint to Amazon S3.
- (IPv6) A ClassicLink connection. You can enable IPv4 communication between a linked EC2-Classic instance and instances in a VPC on the other side of a VPC peering connection. However, IPv6 is not supported in EC2-Classic, so you cannot extend this connection for IPv6 communication.



For example, if VPC A and VPC B are peered, and VPC A has any of these connections, then instances in VPC B cannot use the connection to access resources on the other side of the connection. Similarly, resources on the other side of a connection cannot use the connection to access VPC B.

Hence, this means that you cannot use VPC-2 to extend the peering relationship that exists between VPC-1 and the on-premises network. For example, traffic from the corporate network can't directly access VPC-1 by using the VPN connection or the AWS Direct Connect connection to VPC-2, which is why the following options are incorrect:

- *- Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.***
- *- Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.***
- *- Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.***

You can do the following to provide a highly available, fault-tolerant network connection:

- *- Establish a hardware VPN over the Internet between the VPC and the on-premises network.***
- *- Establish another AWS Direct Connect connection and private virtual interface in the same AWS region.***

References:

- <https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#edge-to-edge-vgw>
- <https://aws.amazon.com/premiumsupport/knowledge-center/configure-vpn-backup-dx/>

<https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/>

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

54. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application, which is used by your clients around the world, is hosted in an Auto Scaling group of EC2 instances behind a Classic Load Balancer. You need to secure your application by allowing multiple domains to serve SSL traffic over the same IP address.

Which of the following should you do to meet the above requirement?

- Use an Elastic IP and upload multiple 3rd party certificates in your Classic Load Balancer using the AWS Certificate Manager.
- It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS.
- Generate an SSL certificate with AWS Certificate Manager and create a CloudFront web distribution. Associate the certificate with your web distribution and enable the support for Server Name Indication (SNI).
- Use Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic.

Incorrect

SNI Custom SSL relies on the SNI extension of the Transport Layer Security protocol, which allows multiple domains to serve SSL traffic over the same IP address by including the hostname which the viewers are trying to connect to.

Amazon CloudFront delivers your content from each edge location and offers the same security as the Dedicated IP Custom SSL feature. SNI Custom SSL works with most modern browsers, including Chrome version 6 and later (running on Windows XP and later or OS X 10.5.7 and later), Safari version 3 and later (running on Windows Vista and later or Mac OS X 10.5.6. and later), Firefox 2.0 and later, and Internet Explorer 7 and later (running on Windows Vista and later).

The screenshot shows the AWS CloudFront SSL Configuration page. At the top, there are tabs for AWS, Tags, and a star icon. On the right, there are links for Global and Support.

SSL Certificate:

- Default CloudFront Certificate (*.cloudfront.net)

Choose this option if you want your users to use HTTPS or HTTP to access your content with the CloudFront domain name (such as https://d111111abccdef8.cloudfront.net/logo.jpg).
Important: If you choose this option, CloudFront requires that browsers or devices support TLSv1 or later to access your content.
- Custom SSL Certificate (example.com):

Choose this option if you want your users to access your content by using an alternate domain name, such as https://www.example.com/logo.jpg.
You can use certificates that you created in AWS Certificate Manager (ACM) in the US-East (N. Virginia) Region, or you can use certificates stored in the IAM certificate store.

Below these options are two buttons: a dropdown menu containing ".leeatk.com" and a "Request an ACM certificate" button.

Learn more about using custom SSL/TLS certificates with CloudFront.
Learn more about using ACM.

Custom SSL Client Support:

- Only Clients that Support Server Name Indication (SNI)

CloudFront serves your content over HTTPS only to clients that support SNI. Older browsers and other clients that do not support SNI can not access your content over HTTPS.
[Learn More](#)
- All Clients (\$600/month prorated charge applies. [Learn about pricing](#).)

CloudFront allocates dedicated IP addresses at each CloudFront edge location to serve your content over HTTPS. Any client can access your content.
[Learn More](#)

Some users may not be able to access your content because some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content. If you need to support non-SNI compliant browsers for HTTPS content, it is recommended to use the Dedicated IP Custom SSL feature.

Using Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic is incorrect because a Classic Load Balancer does not support Server Name Indication (SNI). You have to use an Application Load Balancer instead or a CloudFront web distribution to allow the SNI feature.

Using an Elastic IP and uploading multiple 3rd party certificates in your Application Load Balancer using the AWS Certificate Manager is incorrect because just like in the above, a Classic Load Balancer does not support Server Name Indication (SNI) and the use of an Elastic IP is not a suitable solution to allow multiple domains to serve SSL traffic. You have to use Server Name Indication (SNI).

The option that says: ***It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS*** is incorrect because AWS does support the use of Server Name Indication (SNI).

References:

<https://aws.amazon.com/about-aws/whats-new/2014/03/05/amazon-cloudfront-announces-sni-custom-ssl/>

<https://aws.amazon.com/blogs/security/how-to-help-achieve-mobile-app-transport-security-compliance-by-using-amazon-cloudfront-and-aws-certificate-manager/>

Check out this Amazon CloudFront Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudfront/>

SNI Custom SSL vs Dedicated IP Custom SSL:

<https://tutorialsdojo.com/sni-custom-ssl-vs-dedicated-ip-custom-ssl/>

AWS Security Services Overview – Secrets Manager, ACM, Macie:

<https://youtu.be/ogVamzF2Dzk>

55. QUESTION

Category: CSAA – Design High-Performing Architectures

A Solutions Architect created a new Standard-class S3 bucket to store financial reports that are not frequently accessed but should immediately be available when an auditor requests them. To save costs, the Architect changed the storage class of the S3 bucket from Standard to Infrequent Access storage class.

In Amazon S3 Standard – Infrequent Access storage class, which of the following statements are true? (Select TWO.)

- It is designed for data that requires rapid access when needed.
- It automatically moves data to the most cost-effective access tier without any operational overhead.
- It provides high latency and low throughput performance.
- It is designed for data that is accessed less frequently.
- Ideal to use for data archiving.

Correct

Amazon S3 Standard – Infrequent Access (Standard – IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard – IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee.

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads



This combination of low cost and high performance make Standard – IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard – IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of Standard
- Designed for durability of 99.999999999% of objects
- Designed for 99.9% availability over a given year
- Backed with the Amazon S3 Service Level Agreement for availability
- Supports SSL encryption of data in transit and at rest
- Lifecycle management for automatic migration of objects

Hence, the correct answers are:

- ***It is designed for data that is accessed less frequently.***
- ***It is designed for data that requires rapid access when needed.***

The option that says: ***It automatically moves data to the most cost-effective access tier without any operational overhead*** is incorrect as it actually refers to Amazon S3 – Intelligent Tiering, which is the only cloud storage class that delivers automatic cost savings by moving objects between different access tiers when access patterns change.

The option that says: ***It provides high latency and low throughput performance*** is incorrect as it should be “low latency” and “high throughput” instead. S3 automatically scales performance to meet user demands.

The option that says: ***Ideal to use for data archiving*** is incorrect because this statement refers to Amazon S3 Glacier. Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.

References:

<https://aws.amazon.com/s3/storage-classes/>

<https://aws.amazon.com/s3/faqs>

Check out this Amazon S3 Cheat Sheet:

<https://tutorialsdojo.com/amazon-s3/>

56. QUESTION

Category: CSAA – Design Resilient Architectures

A company has an enterprise web application hosted on Amazon ECS Docker containers that use an Amazon FSx for Lustre filesystem for its high-performance computing workloads. A warm standby environment is running in another AWS region for disaster recovery. A Solutions Architect was assigned to design a system that will automatically route the live traffic to the disaster recovery (DR) environment only in the event that the primary application stack experiences an outage.

What should the Architect do to satisfy this requirement?

- Set up a CloudWatch Alarm to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the `ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record.
- Set up a Weighted routing policy configuration in Route 53 by adding health checks on both the primary stack and the DR environment. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`.
- Set up a CloudWatch Events rule to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the `ChangeResourceRecordSets` API call using the function to initiate the failover to the secondary DNS record.
- Set up a failover routing policy configuration in Route 53 by adding a health check on the primary service endpoint. Configure Route 53 to direct the DNS queries to the secondary record when the primary resource is unhealthy. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the `Evaluate Target Health` option by setting it to `Yes`.

Correct

Use an active-passive failover configuration when you want a primary resource or group of resources to be available majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Route 53 includes only the healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries.

To create an active-passive failover configuration with one primary record and one secondary record, you just create the records and specify **Failover** for the routing policy. When the primary resource is healthy, Route 53 responds to DNS queries using the primary record. When the primary resource is unhealthy, Route 53 responds to DNS queries using the secondary record.

You can configure a health check that monitors an endpoint that you specify either by IP address or by domain name. At regular intervals that you specify, Route 53 submits automated requests over the Internet to your application, server, or other resource to verify that it's reachable, available, and functional. Optionally, you can configure the health check to make requests similar to those that your users make, such as requesting a web page from a specific URL.

Create Record Set

Name:	example.com.
Type:	A – IPv4 address
Alias:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Alias Target:	.us-west-2.elb.amazon
Alias Hosted Zone ID:	[REDACTED]
Routing Policy:	Failover
Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. Learn More	
Failover Record Type:	<input checked="" type="radio"/> Primary <input type="radio"/> Secondary
Set ID:	Primary
Evaluate Target Health:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Create Record Set	

When Route 53 checks the health of an endpoint, it sends an HTTP, HTTPS, or TCP request to the IP address and port that you specified when you created the health check. For a health check to succeed, your router and firewall rules must allow inbound traffic from the IP addresses that the Route 53 health checkers use.

Hence, the correct answer is: ***Set up a failover routing policy configuration in Route 53 by adding a health check on the primary service endpoint. Configure Route 53 to direct the DNS queries to the secondary record when the primary resource is unhealthy. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the * Evaluate Target Health * option by setting it to * Yes . ****

The option that says: ***Set up a Weighted routing policy configuration in Route 53 by adding health checks on both the primary stack and the DR environment. Configure the network access control list and the route table to allow Route 53 to send requests to the endpoints specified in the health checks. Enable the Evaluate Target Health option by setting it to * Yes **** is incorrect because Weighted routing simply lets you associate multiple resources with a single domain name (tutorialsdojo.com) or subdomain name (blog.tutorialsdojo.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of software, but not for a failover configuration. Remember that the scenario says that the solution should automatically route the live traffic to the disaster recovery (DR) environment only in the event that the primary

application stack experiences an outage. This configuration is incorrectly distributing the traffic on both the primary and DR environment.

The option that says: ***Set up a CloudWatch Alarm to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the ChangeResourceRecordSets API call using the function to initiate the failover to the secondary DNS record*** is incorrect because setting up a CloudWatch Alarm and using the Route 53 API is not applicable nor useful at all in this scenario. Remember that CloudWatch Alarm is primarily used for monitoring CloudWatch metrics. You have to use a Failover routing policy instead.

The option that says: ***Set up a CloudWatch Events rule to monitor the primary Route 53 DNS endpoint and create a custom Lambda function. Execute the ChangeResourceRecordSets API call using the function to initiate the failover to the secondary DNS record*** is incorrect because the Amazon CloudWatch Events service is commonly used to deliver a near real-time stream of system events that describe changes in **some** Amazon Web Services (AWS) resources. There is no direct way for CloudWatch Events to monitor the status of your Route 53 endpoints. You have to configure a health check and a failover configuration in Route 53 instead to satisfy the requirement in this scenario.

References:

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-types.html>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/health-checks-types.html>

<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-router-firewall-rules.html>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

57. QUESTION

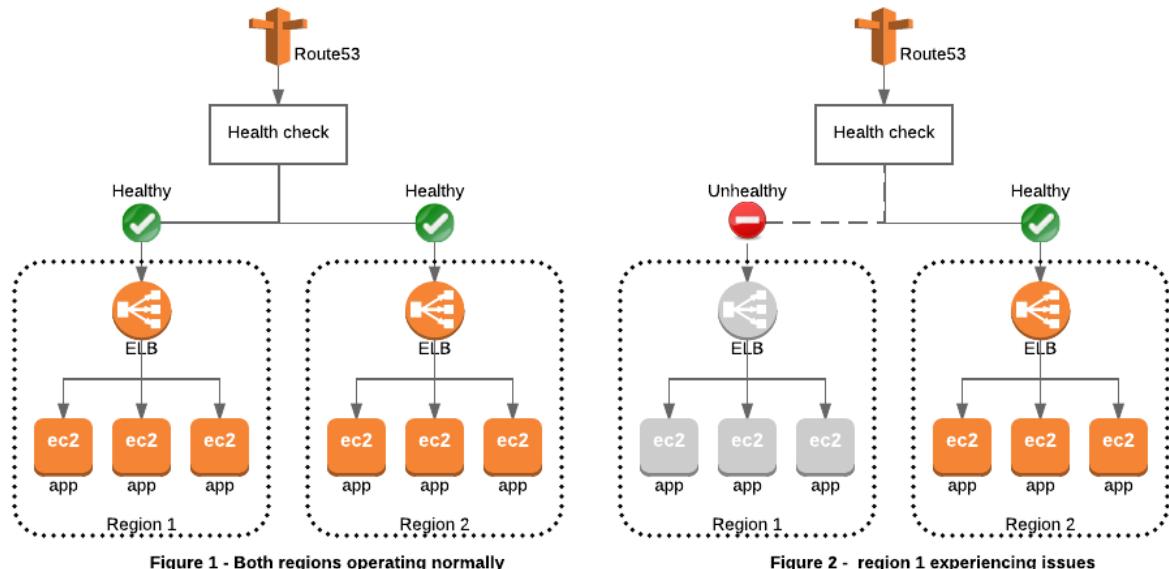
Category: CSAA – Design Resilient Architectures

A company has a dynamic web app written in MEAN stack that is going to be launched in the next month. There is a probability that the traffic will be quite high in the first couple of weeks. In the event of a load failure, how can you set up DNS failover to a static website?

- Duplicate the exact application architecture in another region and configure DNS weight-based routing.
- Enable failover to an application hosted in an on-premises data center.
- Add more servers in case the application fails.
- **Use Route 53 with the failover option to a static S3 website bucket or CloudFront distribution.**

Correct

For this scenario, ***using Route 53 with the failover option to a static S3 website bucket or CloudFront distribution*** is correct. You can create a new Route 53 with the failover option to a static S3 website bucket or CloudFront distribution as an alternative.



Duplicating the exact application architecture in another region and configuring DNS weight-based routing is incorrect because running a duplicate system is not a cost-effective solution. Remember that you are trying to build a failover mechanism for your web app, not a distributed setup.

Enabling failover to an application hosted in an on-premises data center is incorrect. Although you can set up failover to your on-premises data center, you are not maximizing the AWS environment such as using Route 53 failover.

Adding more servers in case the application fails is incorrect because this is not the best way to handle a failover event. If you add more servers only in case the application fails, then there would be a period of downtime in which your application is unavailable. Since there are no running servers on that period, your application will be unavailable for a certain period of time until your new server is up and running.

Reference:

<https://aws.amazon.com/premiumsupport/knowledge-center/fail-over-s3-r53/>

<http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html>

Check out this Amazon Route 53 Cheat Sheet:

<https://tutorialsdojo.com/amazon-route-53/>

58. QUESTION

Category: CSAA – Design High-Performing Architectures

One of your EC2 instances is reporting an unhealthy system status check. The operations team is looking for an easier way to monitor and repair these instances instead of fixing them manually.

How will you automate the monitoring and repair of the system status check failure in an AWS environment?

- Buy and implement a third party monitoring tool.
- **Create CloudWatch alarms that stop and start the instance based on status check alarms.**
- Write a python script that queries the EC2 API for each instance status check
- Write a shell script that periodically shuts down and starts instances based on certain stats.

Correct

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your EC2 instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

Writing a python script that queries the EC2 API for each instance status check, ***writing a shell script that periodically shuts down and starts instances based on certain stats***, and ***buying and implementing a third party monitoring tool*** are all incorrect because it is unnecessary to go through such lengths when CloudWatch Alarms already has such a feature for you, offered at a low cost.

Reference:

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/UsingAlarmActions.html>

Check out this Amazon CloudWatch Cheat Sheet:

<https://tutorialsdojo.com/amazon-cloudwatch/>

59. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company has two On-Demand EC2 instances inside the Virtual Private Cloud in the same Availability Zone but are deployed to different subnets. One EC2 instance is running a database and the other EC2 instance a web application that connects with the database. You need to ensure that these two instances can communicate with each other for the system to work properly.

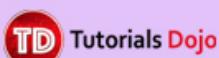
What are the things you have to check so that these EC2 instances can communicate inside the VPC?
(Select TWO.)

- Ensure that the EC2 instances are in the same Placement Group.
- Check the Network ACL if it allows communication between the two subnets.
- Check if both instances are the same instance class.
- Check if the default route is set to a NAT instance or Internet Gateway (IGW) for them to communicate.
- Check if all security groups are set to allow the application host to communicate to the database on the right port and protocol.

Incorrect

First, the Network ACL should be properly set to allow communication between the two subnets. The security group should also be properly configured so that your web server can communicate with the database server.

Security Group	Network Access Control List
Acts as a firewall for associated Amazon EC2 instances	Acts as a firewall for associated subnets
Controls both inbound and outbound traffic at the instance level You can secure your VPC instances using only security groups	Controls both inbound and outbound traffic at the subnet level Network ACLs are an additional layer of defense.
Supports allow rules only	Supports allow rules and deny rules
Stateful (Return traffic is automatically allowed, regardless of any rules)	Stateless (Return traffic must be explicitly allowed by rules)
Evaluates all rules before deciding whether to allow traffic	Evaluates rules in number order when deciding whether to allow traffic, starting with the lowest numbered rule.
Applies only to the instance that is associated to it	Applies to all instances in the subnet it is associated with
Has separate rules for inbound and outbound traffic	Has separate rules for inbound and outbound traffic
A newly created security group denies all inbound traffic by default	A newly created nACL denies all inbound traffic by default
A newly created security group has an outbound rule that allows all outbound traffic by default	A newly created nACL denies all outbound traffic default
Instances associated with a security group can't talk to each other unless you add rules allowing it	Each subnet in your VPC must be associated with a network ACL. If none is associated, the default nACL is selected.
Security groups are associated with network interfaces	You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.



Hence, these are the correct answers:

1. ***Check if all security groups are set to allow the application host to communicate to the database on the right port and protocol.***
2. ***Check the Network ACL if it allows communication between the two subnets.***

The option that says: ***Check if both instances are the same instance class*** is incorrect because the EC2 instances do not need to be of the same class in order to communicate with each other.

The option that says: ***Check if the default route is set to a NAT instance or Internet Gateway (IGW) for them to communicate*** is incorrect because an Internet gateway is primarily used to communicate to the Internet.

The option that says: ***Ensure that the EC2 instances are in the same Placement Group*** is incorrect because Placement Group is mainly used to provide low-latency network performance necessary for tightly-coupled node-to-node communication.

Reference:

http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

Check out this Amazon VPC Cheat Sheet:

<https://tutorialsdojo.com/amazon-vpc/>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>

60. QUESTION

Category: CSAA – Design Resilient Architectures

As part of the Business Continuity Plan of your company, your IT Director instructed you to set up an automated backup of all of the EBS Volumes for your EC2 instances as soon as possible.

What is the fastest and most cost-effective solution to automatically back up all of your EBS Volumes?

- For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically.
- Set your Amazon Storage Gateway with EBS volumes as the data source and store the backups in your on-premises servers through the storage gateway.
- Use an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes.
- **Use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots.**

Incorrect

You can use **Amazon Data Lifecycle Manager (Amazon DLM)** to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes. Automating snapshot management helps you to:

- Protect valuable data by enforcing a regular backup schedule.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.

Combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon DLM provides a complete backup solution for EBS volumes at no additional cost.

Hence, ***using Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots*** is the correct answer as it is the fastest and most cost-effective solution that provides an automated way of backing up your EBS volumes.

The option that says: ***For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically*** is incorrect because even though this is a valid solution, you would still need additional time to create a scheduled job that calls the "create-snapshot" command. It would be better to use Amazon Data Lifecycle Manager (Amazon DLM) instead as this provides you the fastest solution which enables you to automate the creation, retention, and deletion of the EBS snapshots without having to write custom shell scripts or creating scheduled jobs.

Setting your Amazon Storage Gateway with EBS volumes as the data source and storing the backups in your on-premises servers through the storage gateway is incorrect as the **Amazon Storage Gateway is used only for creating a backup of data from your on-premises server and not from the Amazon Virtual Private Cloud.**

Using an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes is incorrect as there is no such thing as EBS-cycle policy in Amazon S3.

References:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html>

Check out this Amazon EBS Cheat Sheet:

<https://tutorialsdojo.com/amazon-ebs/>

Amazon EBS Overview – SSD vs HDD:

<https://youtu.be/LW7x8wyLFvw>

61. QUESTION

Category: CSAA – Design High-Performing Architectures

An insurance company plans to implement a message filtering feature in their web application. To implement this solution, they need to create separate Amazon SQS queues for each type of quote request. The entire message processing should not exceed 24 hours.

As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

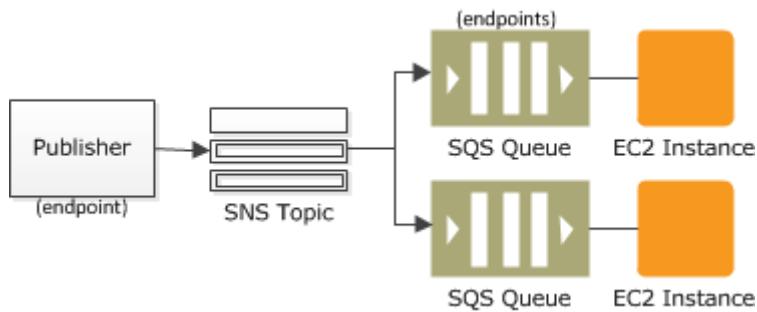
- Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type.
- Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type.
- Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type.
- Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type.

Incorrect

Amazon SNS is a fully managed pub/sub messaging service. With Amazon SNS, you can use topics to simultaneously distribute messages to multiple subscribing endpoints such as Amazon SQS queues, AWS Lambda functions, HTTP endpoints, email addresses, and mobile devices (SMS, Push).

Amazon SQS is a message queue service used by distributed applications to exchange messages through a polling model. It can be used to decouple sending and receiving components without requiring each component to be concurrently available.

A fanout scenario occurs when a message published to an SNS topic is replicated and pushed to multiple endpoints, such as Amazon SQS queues, HTTP(S) endpoints, and Lambda functions. This allows for parallel asynchronous processing.



For example, you can develop an application that publishes a message to an SNS topic whenever an order is placed for a product. Then, two or more SQS queues that are subscribed to the SNS topic receive identical notifications for the new order. An Amazon Elastic Compute Cloud (Amazon EC2) server instance attached to one of the SQS queues can handle the processing or fulfillment of the order. And you can attach another Amazon EC2 server instance to a data warehouse for analysis of all orders received.

By default, an Amazon SNS topic subscriber receives every message published to the topic. You can use Amazon SNS message filtering to assign a filter policy to the topic subscription, and the subscriber will only receive a message that they are interested in. Using Amazon SNS and Amazon SQS together, messages can be delivered to applications that require immediate notification of an event. This method is known as fanout to Amazon SQS queues.

Hence, the correct answer is: ***Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type.***

The option that says: ***Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type*** is incorrect because this option will distribute the same messages on all SQS queues instead of its designated queue. You need to fan-out the messages to multiple SQS queues using a filter policy in Amazon SNS subscriptions to allow parallel asynchronous processing. By doing so, the entire message processing will not exceed 24 hours.

The option that says: ***Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type*** is incorrect because to implement the solution asked in the scenario, you only need to use one Amazon SNS topic. To publish it to the designated SQS queue, you must set a filter policy that allows you to fanout the messages. If you didn't set a filter policy in Amazon SNS, the subscribers would receive all the messages published to the SNS topic. Thus, using multiple SNS topics is not an appropriate solution for this scenario.

The option that says: ***Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type*** is incorrect because Amazon KDS is not a message filtering service. You should use Amazon SNS and SQS to distribute the topic to the designated queue.

References:

<https://aws.amazon.com/getting-started/hands-on/filter-messages-published-to-topics/>

<https://docs.aws.amazon.com/sns/latest/dg/sns-message-filtering.html>

<https://docs.aws.amazon.com/sns/latest/dg/sns-sqs-as-subscriber.html>

Check out this Amazon SNS and SQS Cheat Sheets:

<https://tutorialsdojo.com/amazon-sns/>

<https://tutorialsdojo.com/amazon-sqs/>

Amazon SNS Overview:

<https://youtu.be/ft5R45IEUJ8>

62. QUESTION

Category: CSAA – Design Resilient Architectures

A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture that is composed of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used by an online accounting application that requires path-based routing, host-based routing, and bi-directional communication channels using WebSockets.

Which is the most suitable type of Elastic Load Balancer that will satisfy the given requirement?

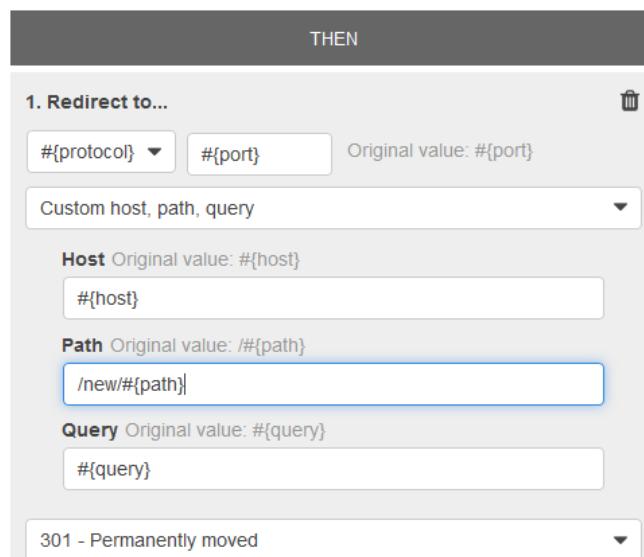
- Either a Classic Load Balancer or a Network Load Balancer
- Classic Load Balancer
- Network Load Balancer
- Application Load Balancer

Correct

Elastic Load Balancing supports three types of load balancers. You can select the appropriate load balancer based on your application needs.

If you need flexible application management and TLS termination then it is recommended to use Application Load Balancer. If extreme performance and static IP is needed for your application then it is recommend that you use Network Load Balancer. If your application is built within the EC2 Classic network then you should use Classic Load Balancer.

An **Application Load Balancer** functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.



Application Load Balancers support path-based routing, host-based routing, and support for containerized applications hence, ***Application Load Balancer*** is the correct answer.

Network Load Balancer, ***Classic Load Balancer***, and ***either a Classic Load Balancer or a Network Load Balancer*** are all incorrect as none of these support path-based routing and host-based routing, unlike an Application Load Balancer.

References:

<https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits>

<https://aws.amazon.com/elasticloadbalancing/faqs/>

AWS Elastic Load Balancing Overview:

<https://youtu.be/UBI5dw59DO8>

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

<https://tutorialsdojo.com/aws-elastic-load-balancing-elb/>

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

<https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/>

63. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect is building a cloud infrastructure where EC2 instances require access to various AWS services such as S3 and Redshift. The Architect will also need to provide access to system administrators so they can deploy and test their changes.

Which configuration should be used to ensure that the access to the resources is secured and not compromised? (Select TWO.)

- Assign an IAM role to the Amazon EC2 instance.
- Enable Multi-Factor Authentication.
- Store the AWS Access Keys in the EC2 instance.
- Store the AWS Access Keys in ACM.
- Assign an IAM user for each Amazon EC2 Instance.

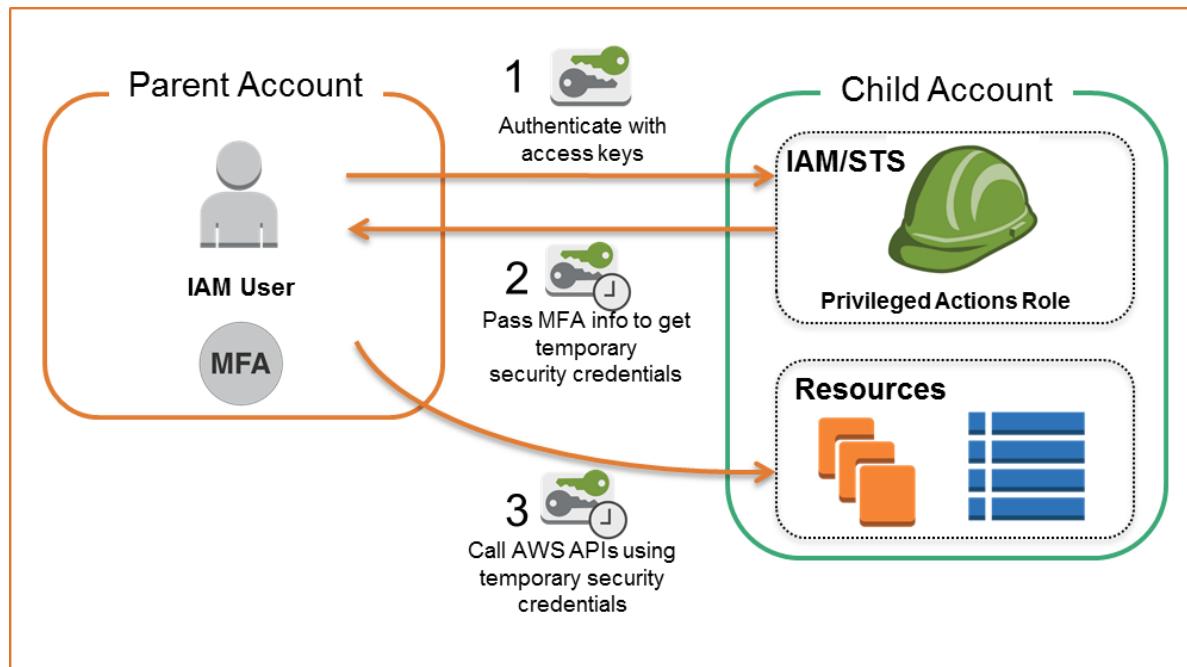
Incorrect

In this scenario, the correct answers are:

- Enable Multi-Factor Authentication

- Assign an IAM role to the Amazon EC2 instance

Always remember that you should associate IAM roles to EC2 instances and not an IAM user, for the purpose of accessing other AWS services. IAM roles are designed so that your applications can securely make API requests from your instances, without requiring you to manage the security credentials that the applications use. Instead of creating and distributing your AWS credentials, you can delegate permission to make API requests using IAM roles.



AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of your user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password (the first factor—what they know), as well as for an authentication code from their AWS MFA device (the second factor—what they have). Taken together, these multiple factors provide increased security for your AWS account settings and resources. You can enable MFA for your AWS account and for individual IAM users you have created under your account. MFA can also be used to control access to AWS service APIs.

Storing the AWS Access Keys in the EC2 instance is incorrect. This is not recommended by AWS as it can be compromised. Instead of storing access keys on an EC2 instance for use by applications that run on the instance and make AWS API requests, you can use an IAM role to provide temporary access keys for these applications.

Assigning an IAM user for each Amazon EC2 Instance is incorrect because there is no need to create an IAM user for this scenario since IAM roles already provide greater flexibility and easier management.

Storing the AWS Access Keys in ACM is incorrect because ACM is just a service that lets you easily provision, manage, and deploy public and private SSL/TLS certificates for use with AWS services and your internal connected resources. It is not used as a secure storage for your access keys.

References:

<https://aws.amazon.com/iam/details/mfa/>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html>

Check out this AWS IAM Cheat Sheet:

<https://tutorialsdojo.com/aws-identity-and-access-management-iam/>

64. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A company has a requirement to move 80 TB data warehouse to the cloud. It would take 2 months to transfer the data given their current bandwidth allocation.

Which is the most cost-effective service that would allow you to quickly upload their data into AWS?

- **AWS Snowball Edge**
- AWS Direct Connect
- Amazon S3 Multipart Upload
- AWS Snowmobile

Correct

AWS Snowball Edge is a type of Snowball device with on-board storage and compute power for select **AWS capabilities**. Snowball Edge can undertake local processing and edge-computing workloads in addition to transferring data between your local environment and the AWS Cloud.

Each Snowball Edge device can transport data at speeds faster than the internet. This transport is done by shipping the data in the appliances through a regional carrier. The appliances are rugged shipping containers, complete with E Ink shipping labels. The AWS Snowball Edge device differs from the standard Snowball because it can bring the power of the AWS Cloud to your on-premises location, with local storage and compute functionality.

Snowball Edge devices have three options for device configurations – storage optimized, compute optimized, and with GPU.



Hence, the correct answer is: ***AWS Snowball Edge.***

AWS Snowmobile is incorrect because this **is an Exabyte-scale data transfer service used to move extremely large amounts of data to AWS.** It is not suitable for transferring a small amount of data, like 80 TB in this scenario. You can transfer up to 100PB per Snowmobile, a 45-foot long ruggedized shipping container, pulled by a semi-trailer truck. A more cost-effective solution here is to order a Snowball Edge device instead.

AWS Direct Connect is incorrect because it **is primarily used to establish a dedicated network connection from your premises network to AWS.** This is not suitable for one-time data transfer tasks, like what is depicted in the scenario.

Amazon S3 Multipart Upload is incorrect because this feature simply enables you to upload large objects in multiple parts. It still uses the same Internet connection of the company, which means that the transfer will still take time due to its current bandwidth allocation.

References:

<https://docs.aws.amazon.com/snowball/latest/ug/whatissnowball.html>

<https://docs.aws.amazon.com/snowball/latest/ug/device-differences.html>

Check out this AWS Snowball Edge Cheat Sheet:

<https://tutorialsdojo.com/aws-cheat-sheet-aws-snowball-edge/>

Here is a quick introduction on AWS Snowball Edge:

<https://youtu.be/bxSD1Nha2k8>

Using AWS Snowball Edge and AWS DMS for Database Migration:

<https://youtu.be/6Hw--HE8ILg>

65. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

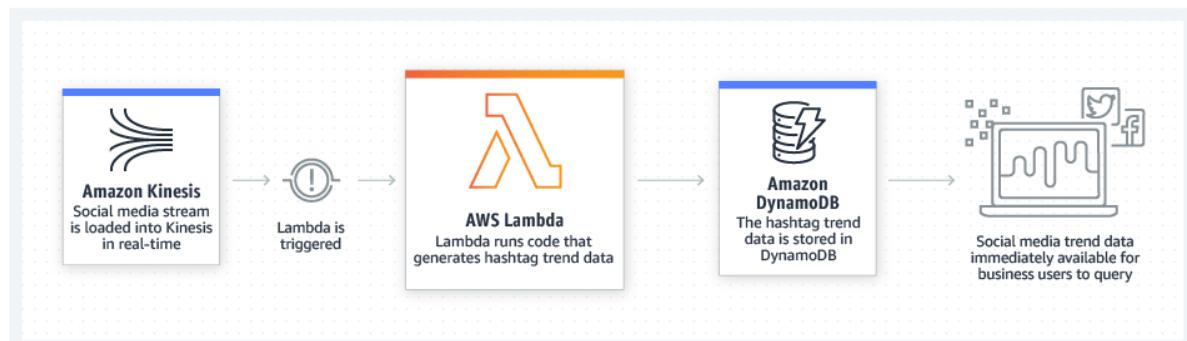
A company is building an internal application that serves as a repository for images uploaded by a couple of users. Whenever a user uploads an image, it would be sent to Kinesis Data Streams for processing before it is stored in an S3 bucket. If the upload was successful, the application will return a prompt informing the user that the operation was successful. The entire processing typically takes about 5 minutes to finish.

Which of the following options will allow you to asynchronously process the request to the application from upload request to Kinesis, S3, and return a reply in the most cost-effective manner?

- Use a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests.
- Use a combination of SQS to queue the requests and then asynchronously process them using On-Demand EC2 Instances.
- Replace the Kinesis Data Streams with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.
- Use a combination of SNS to buffer the requests and then asynchronously process them using On-Demand EC2 Instances.

Incorrect

AWS Lambda supports the synchronous and asynchronous invocation of a Lambda function. You can control the invocation type only when you invoke a Lambda function. When you use an AWS service as a trigger, the invocation type is predetermined for each service. You have no control over the invocation type that these event sources use when they invoke your Lambda function. Since processing only takes 5 minutes, Lambda is also a cost-effective choice.



You can use an AWS Lambda function to process messages in an Amazon Simple Queue Service (Amazon SQS) queue. Lambda event source mappings support standard queues and first-in, first-out (FIFO) queues. With Amazon SQS, you can offload tasks from one component of your application by sending them to a queue and processing them asynchronously.

Kinesis Data Streams is a real-time data streaming service that requires the provisioning of shards. Amazon SOS is a cheaper option because you only pay for what you use. Since there is no requirement for real-time processing in the scenario given, replacing Kinesis Data Streams with Amazon SOS would save more costs.

Hence, the correct answer is: ***Replace the Kinesis stream with an Amazon SQS queue. Create a Lambda function that will asynchronously process the requests.***

Using a combination of Lambda and Step Functions to orchestrate service components and asynchronously process the requests is incorrect. The AWS Step Functions service lets you coordinate multiple AWS services into serverless workflows so you can build and update apps quickly. Although this can be a valid solution, it is not cost-effective since the application does not have a lot of components to orchestrate. Lambda functions can effectively meet the requirements in this scenario without using Step Functions. This service is not as cost-effective as Lambda.

Using a combination of SQS to queue the requests and then asynchronously processing them using On-Demand EC2 Instances and ***Using a combination of SNS to buffer the requests and then asynchronously processing them using On-Demand EC2 Instances*** are both incorrect as using On-Demand EC2 instances is not cost-effective. It is better to use a Lambda function instead.

References:

<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

<https://docs.aws.amazon.com/lambda/latest/dg/lambda-invocation.html>

<https://aws.amazon.com/blogs/compute/new-aws-lambda-controls-for-stream-processing-and-asynchronous-invocations/>

AWS Lambda Overview – Serverless Computing in AWS:

<https://youtu.be/bPVX1zHwAnY>

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

<https://tutorialsdojo.com/aws-certified-solutions-architect-associate/>