AWS SA PRACTICE QUESTIONS

Reference:
https://docs.aws.amazon.com/
https:/tutorialsdojo.com/

1. QUESTION

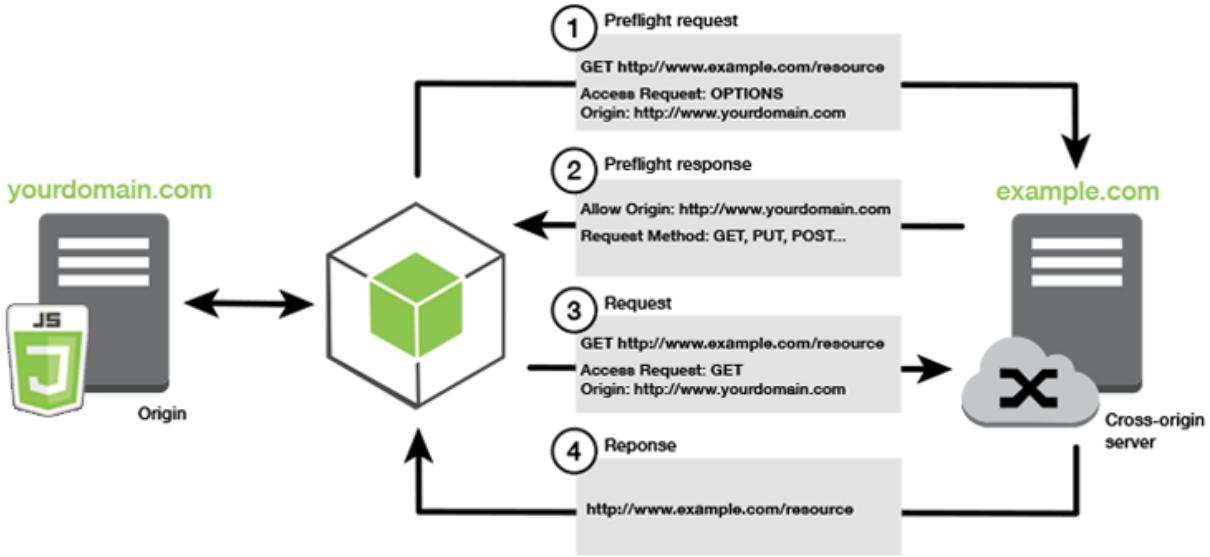
Category: CSAA – Design Secure Architectures

A Solutions Architect is hosting a website in an Amazon S3 bucket named `alxuser`. The users load the website using the following URL: `http://alxuser.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 (`alxuser.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-Zone Load Balancing.
- Enable Cross-Region Replication (CRR).
- Enable Cross-origin resource sharing (CORS) configuration in the bucket. (Correct)
- Enable cross-account access.

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/ManageCorsU

2. 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 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 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. (Correct)
- 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 another Amazon SQS queue for the other team. Grant Amazon S3 permission to send a notification to the second SQS queue.

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.

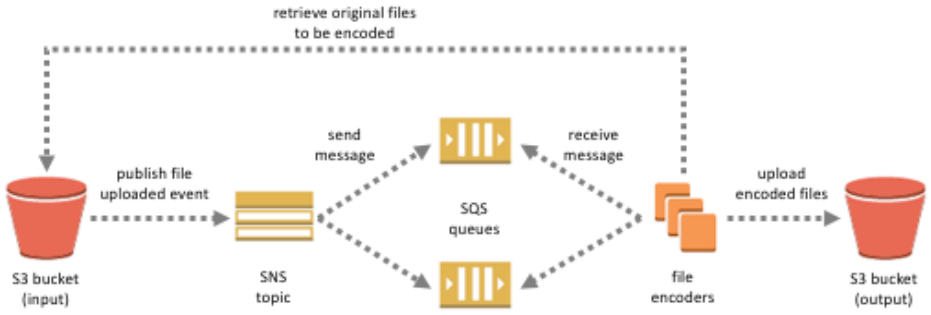
Amazon S3 supports the following destinations where it can publish events:

– 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

https://youtu.be/ft5R45lEUJ8

3. QUESTION

Category: CSAA – Design High-Performing Architectures

An e-commerce company operates a highly scalable web application that relies on an Amazon Aurora database. As their users multiply, they've noticed that the read replica struggles to keep up with the increasing read traffic, leading to performance bottlenecks during peak periods.

As a solutions architect, which of the following will address the issue with the most cost-effective solution?

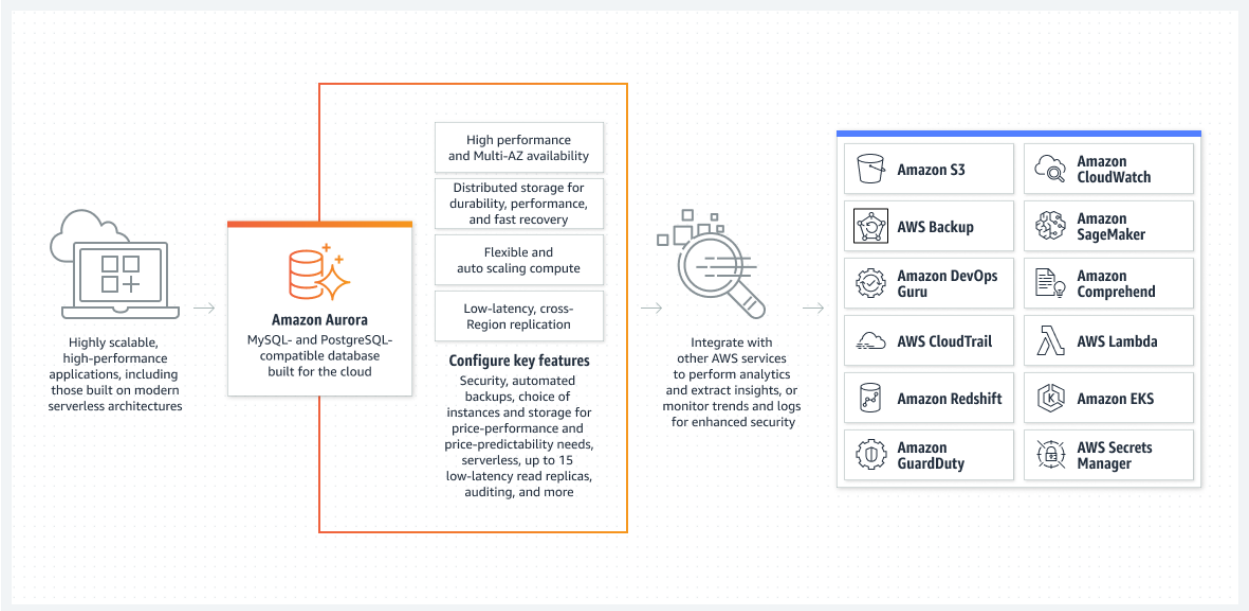
Increase the size of the Amazon Aurora DB cluster.

Use automatic scaling for the Amazon Aurora read replica using Aurora Auto Scaling. (Correct)

Implement read scaling with Amazon Aurora Global Database.

Set up a read replica that can operate across different regions.

Amazon Aurora is a cloud-based relational database service that provides better performance and reliability for database workloads. It is highly available and scalable, making it a great choice for businesses of any size. One of the key features of Amazon Aurora is Aurora Auto Scaling, which automatically adjusts the capacity of your Aurora database cluster based on the workload. This means that you don't have to worry about manually adjusting the ability of your database cluster to handle changes in demand. With Aurora Auto Scaling, you can be sure that your database cluster will always have the appropriate capacity to handle your workload while minimizing costs.



Aurora Auto Scaling is particularly useful for businesses that have fluctuating workloads. It ensures that your database cluster scales up or down as needed without manual intervention. This feature saves time and resources, allowing businesses to focus on other aspects of their operations. Aurora Auto Scaling is also

cost-effective, as it helps minimize unnecessary expenses associated with overprovisioning or underprovisioning database resources.

In this scenario, the company can benefit from using Aurora Auto Scaling. This solution allows the system to dynamically manage resources, effectively addressing the surge in read traffic during peak periods. This dynamic management of resources ensures that the company pays only for the extra resources when they are genuinely required.

Hence the correct answer is: **Use automatic scaling for the Amazon Aurora read replica using Aurora Auto Scaling.**

Increase the size of the Amazon Aurora DB cluster is incorrect because it's not economical to upsize the cluster just to alleviate the bottleneck during peak periods. A static increase in the DB cluster size results in constant costs, regardless of whether your database's resources are being fully utilized during off-peak periods or not.

Implement read scaling with Amazon Aurora Global Database is incorrect. Amazon Aurora Global Database is designed for globally distributed applications, allowing a single Amazon Aurora database to span multiple AWS Regions. While this can provide global availability, it introduces additional complexity and can be more expensive due to infrastructure and data transfer costs.

Set up a read replica that can operate across different regions is incorrect. Setting up a read replica that operates across different regions can provide read scalability and load-balancing benefits by distributing the read traffic across regions. However, it is not the most cost-effective solution in this scenario since it incurs additional costs associated with inter-region data replication. Moreover, the issue is not related to cross-region availability but rather the read replica's performance within the current region.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Integrating.AutoScaling.html

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/CHAP_AuroraOverview.html

4. 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.

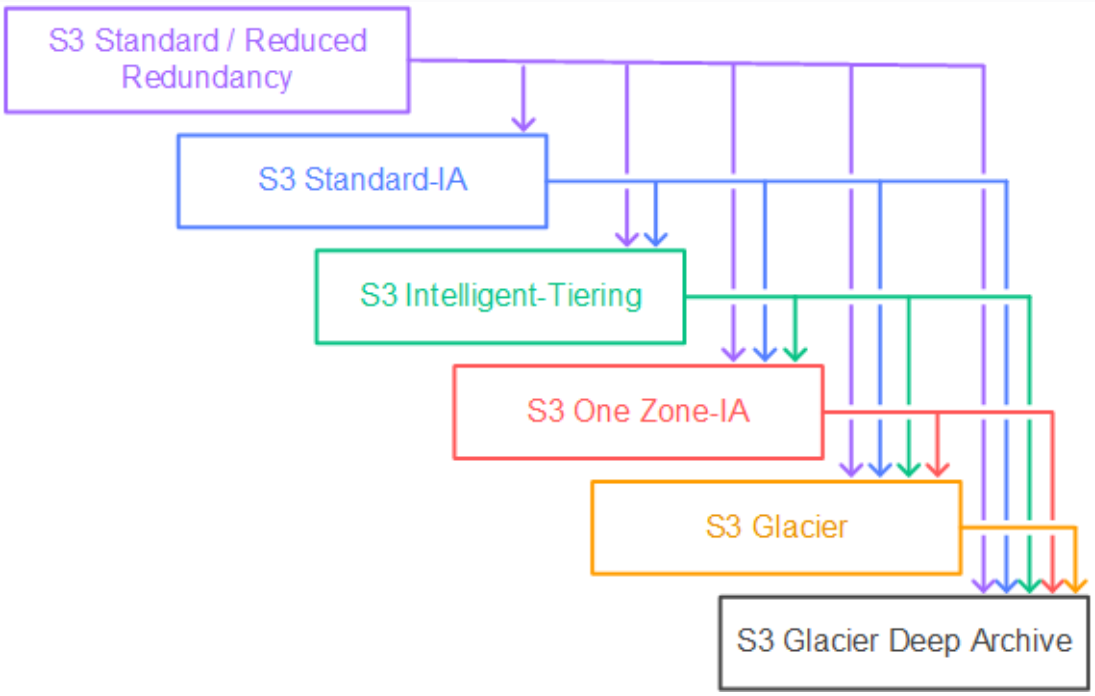
What should you do to meet the above requirement?

- **Attach an 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.** (Correct)
- **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.**

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/

5. QUESTION

Category: CSAA – Design High-Performing Architectures

A company wishes to query data that resides in multiple AWS accounts from a central data lake. Each account has its own Amazon S3 bucket that stores data unique to its business function. Users from different accounts must be granted access to the data lake based on their roles.

Which solution will minimize overhead and costs while meeting the required access patterns?

Use AWS Lake Formation to consolidate data from multiple accounts into a single account. (Correct)

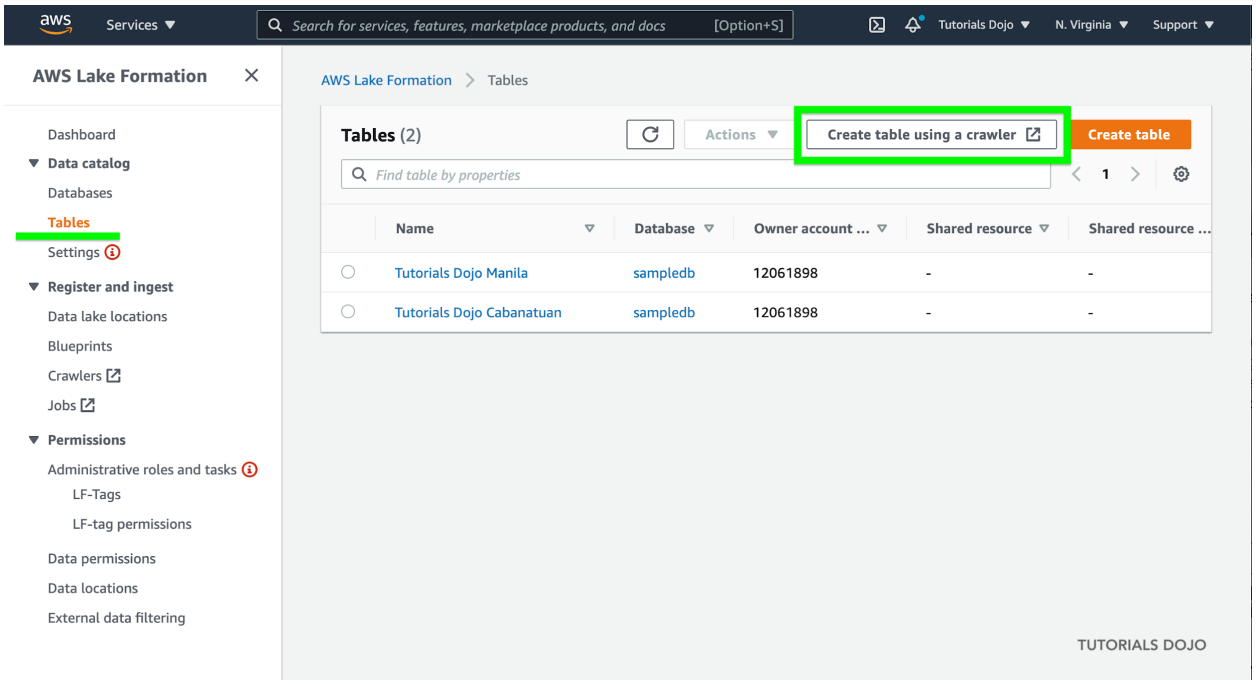
Create a scheduled Lambda function for transferring data from multiple accounts to the S3 buckets of a central account

Use AWS Control Tower to centrally manage each account's S3 buckets.

Use AWS Kinesis Firehose to consolidate data from multiple accounts into a single account.

AWS Lake Formation is a service that makes it easy to set up a secure data lake in days. A data lake is a centralized, curated, and secured repository that stores all your data, both in its original form and prepared for analysis. A data lake enables you to break down data silos and combine different types of analytics to gain insights and guide better business decisions.

Amazon S3 forms the storage layer for Lake Formation. If you already use S3, you typically begin by registering existing S3 buckets that contain your data. Lake Formation creates new buckets for the data lake and import data into them. AWS always stores this data in your account, and only you have direct access to it.



AWS Lake Formation is integrated with AWS Glue which you can use to create a data catalog that describes available datasets and their appropriate business applications. Lake Formation lets you define policies and control data access with simple "grant and revoke permissions to data" sets at granular levels. You can assign permissions to IAM users, roles, groups, and Active Directory users using federation. You specify permissions on catalog objects (like tables and columns) rather than on buckets and objects.

Thus, the correct answer is: **Use AWS Lake Formation to consolidate data from multiple accounts into a single account.**

The option that says: **Use AWS Kinesis Firehose to consolidate data from multiple accounts into a single account** is incorrect. Setting up a Kinesis Firehose in each and every account to move data into a single location is costly and impractical. A better approach is to set up cross-account sharing which is free with AWS Lake Formation.

The option that says: **Create a scheduled Lambda function for transferring data from multiple accounts to the S3 buckets of a central account** is incorrect. This could be done by utilizing the AWS SDK, but implementation would be difficult and quite challenging to manage. Remember that the scenario explicitly mentioned that the solution must minimize management overhead.

The option that says: **Use AWS Control Tower to centrally manage each account's S3 buckets** is incorrect because the AWS Central Tower service is primarily used to manage and govern multiple AWS accounts and not just S3 buckets. Using the AWS Lake Formation service is a more suitable choice.

References:

https://aws.amazon.com/blogs/big-data/building-securing-and-managing-data-lakes-with-aws-lake-formation/

https://docs.aws.amazon.com/lake-formation/latest/dg/how-it-works.html

Check out this AWS Control Tower Cheat Sheet:

https://tutorialsdojo.com/aws-control-tower/

6. QUESTION

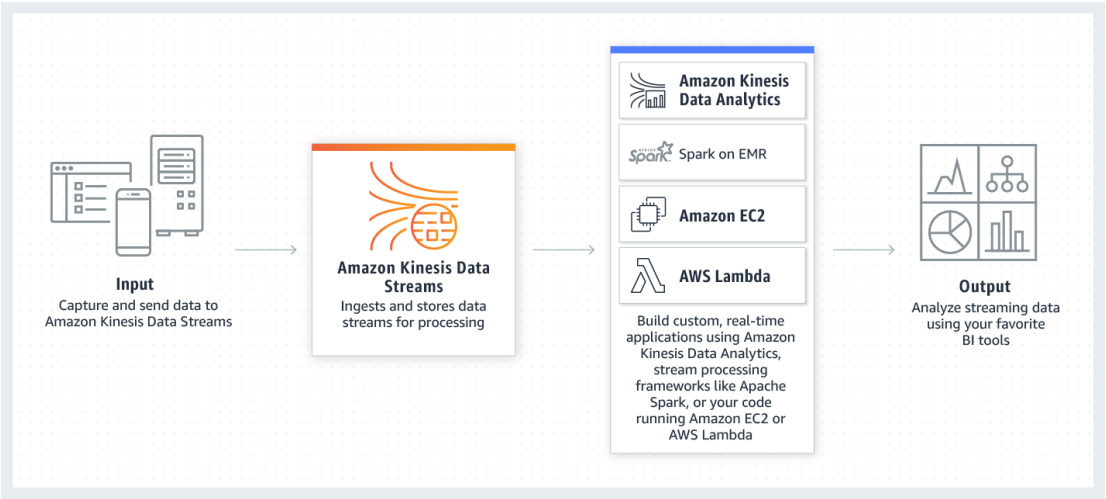
Category: CSAA – Design High-Performing Architectures

A healthcare organization wants to build a system that can predict drug prescription abuse. They will gather real-time data from multiple sources, which includes Personally Identifiable Information (PII). It's crucial that this sensitive information is anonymized prior to landing in a NoSQL database for further processing.

Which solution would meet the requirements?

- **Stream the data in an Amazon DynamoDB table. Enable DynamoDB Streams, and configure a function that performs anonymization on newly written items.**
- **Ingest real-time data using Amazon Kinesis Data Stream. Use a Lambda function to anonymize the PII, then store it in Amazon DynamoDB.** (Correct)
- **Create a data lake in Amazon S3 and use it as the primary storage for patient health data. Use an S3 trigger to run a Lambda function that performs anonymization. Send the anonymized data to Amazon DynamoDB**
- **Deploy an Amazon Kinesis Data Firehose stream to capture and transform the streaming data. Deliver the anonymized data to Amazon Redshift for analysis.**

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources.



Kinesis Data Streams integrates seamlessly with AWS Lambda, which can be utilized to transform and anonymize the Personally Identifiable Information (PII) in transit prior to storage. This ensures that sensitive information is appropriately anonymized at the earliest opportunity, significantly reducing the risk of any data breaches or privacy violations. Finally, the anonymized data is stored in Amazon DynamoDB, a NoSQL database suitable for handling the processed data.

Hence, the correct answer in this scenario is: **Ingest real-time data using Amazon Kinesis Data Stream. Use a Lambda function to anonymize the PII, then store it in Amazon DynamoDB.**

The option that says: **Create a data lake in Amazon S3 and use it as the primary storage for patient health data. Use an S3 trigger to run a Lambda function that performs anonymization. Send the anonymized data to Amazon DynamoDB** is incorrect. This approach doesn't guarantee the anonymization of data before it lands on DynamoDB. The data will first be stored in S3 and then anonymized, potentially exposing sensitive information. This violates the principle of ensuring PII is anonymized prior to storage.

The options that says: **Stream the data in an Amazon DynamoDB table. Enable DynamoDB Streams, and configure a function that performs anonymization on newly written items** is incorrect. DynamoDB streams operate on changes to data that has already been written to the database. Therefore, the PII will be stored in DynamoDB before the anonymization function is triggered, which is a potential privacy concern.

The options that says: **Deploy an Amazon Kinesis Data Firehose stream to capture and transform the streaming data. Deliver the anonymized data to Amazon Redshift for analysis** is incorrect. The requirement was to store the data in a NoSQL database. Amazon Redshift is a data warehousing solution built on a relational database model, not a NoSQL model, which makes this option unsuitable to meet the given requirements.

References:

https://aws.amazon.com/kinesis/data-streams/

https://docs.aws.amazon.com/lambda/latest/dg/with-kinesis.html

Check out this Amazon Kinesis Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Secure 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.

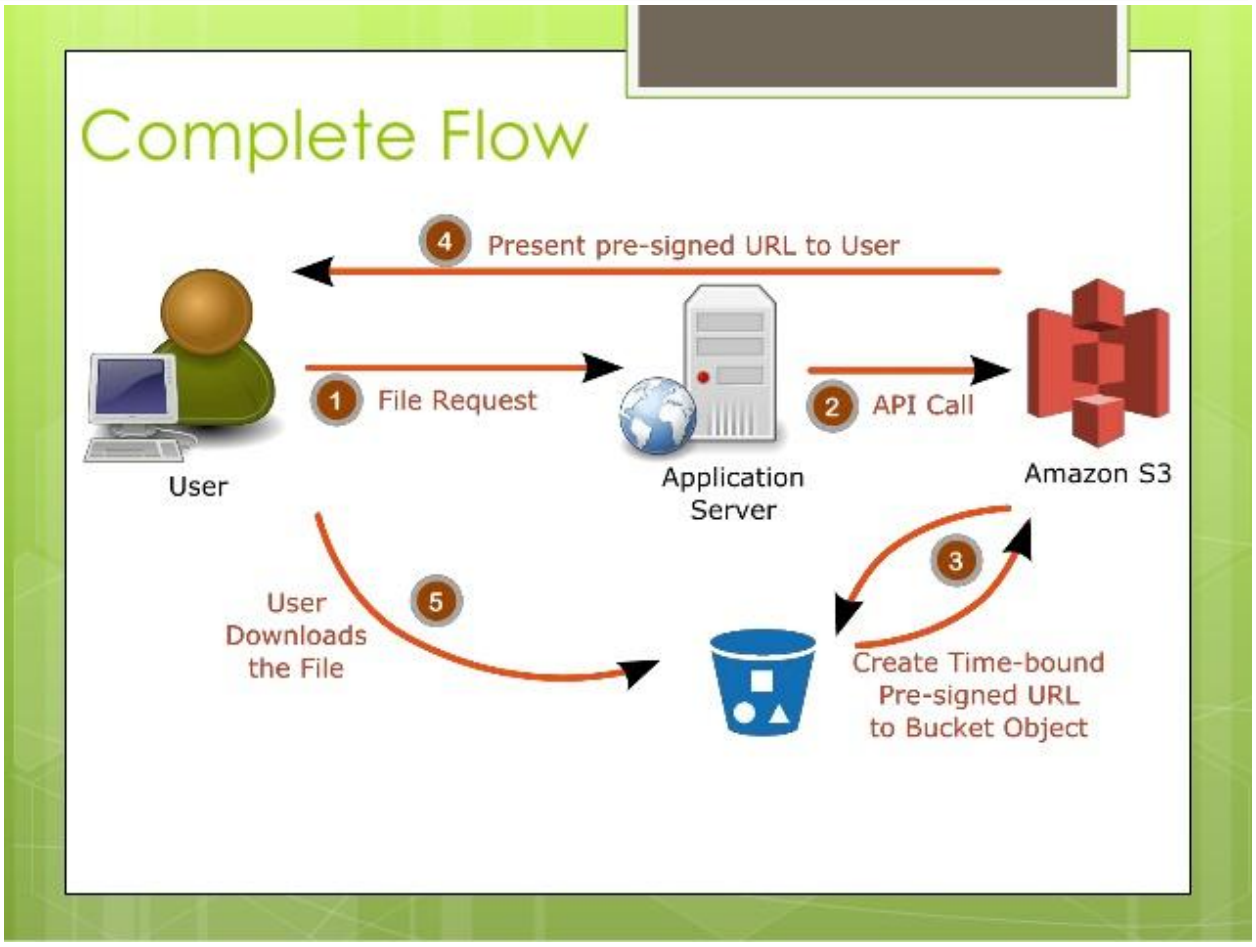
What is the MOST effective method to mitigate this issue?

- Store and privately serve the high-quality photos on Amazon WorkDocs instead.
- Block the IP addresses of the offending websites using NACL.
- Use CloudFront distributions for your photos.
- 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/

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/

8. QUESTION

Category: CSAA – Design Secure Architectures

A medical records company is planning to store sensitive clinical trial data in an Amazon S3 repository with the object-level versioning feature enabled. The Solutions Architect is tasked with ensuring that no object can be overwritten or deleted by any user in a period of one year only. To meet the strict compliance requirements, the root user of the company's AWS account must also be restricted from making any changes to an object in the S3 bucket.

Which of the following is the most secure way of storing the data in Amazon S3?

Enable S3 Object Lock in compliance mode with a retention period of one year. . (Correct)

Enable S3 Object Lock in compliance mode with a legal hold of one year.

Enable S3 Object Lock in governance mode with a retention period of one year.

Enable S3 Object Lock in governance mode with a legal hold of one year.

With S3 Object Lock, you can store objects using a write-once-read-many (WORM) model. Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. You can use Object Lock to help meet regulatory requirements that require WORM storage or to simply add another layer of protection against object changes and deletion.

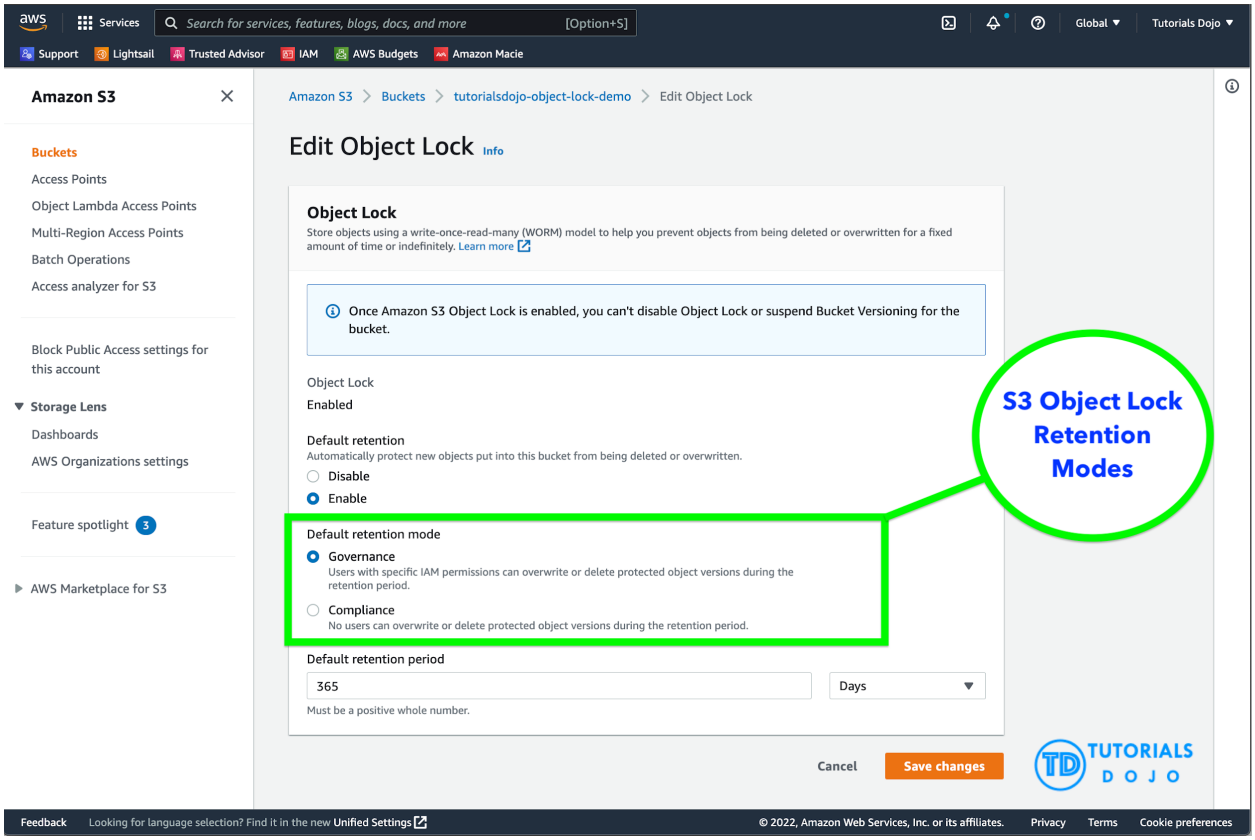
Before you lock any objects, you have to enable a bucket to use S3 Object Lock. You enable Object Lock when you create a bucket. After you enable Object Lock on a bucket, you can lock objects in that bucket. When you create a bucket with Object Lock enabled, you can't disable Object Lock or suspend versioning for that bucket.

S3 Object Lock provides two retention modes:

-Governance mode

-Compliance mode

These retention modes apply different levels of protection to your objects. You can apply either retention mode to any object version that is protected by Object Lock.



In governance mode, users can't overwrite or delete an object version or alter its lock settings unless they have special permissions. With governance mode, you protect objects against being deleted by most users, but you can still grant some users permission to alter the retention settings or delete the object if necessary. You can also use governance mode to test retention-period settings before creating a compliance-mode retention period.

In compliance mode, a protected object version can't be overwritten or deleted by any user, including the root user in your AWS account. When an object is locked in compliance mode, its retention mode can't be changed, and its retention period can't be shortened. Compliance mode helps ensure that an object version can't be overwritten or deleted for the duration of the retention period.

To override or remove governance-mode retention settings, a user must have the `s3:BypassGovernanceRetention` permission and must explicitly include `x-amz-bypass-governance-retention:true` as a request header with any request that requires overriding governance mode.

Legal Hold vs. Retention Period

With Object Lock, you can also place a legal hold on an object version. Like a retention period, a legal hold prevents an object version from being overwritten or deleted. However, a legal hold doesn't have an associated retention period and

remains in effect until removed. Legal holds can be freely placed and removed by any user who has the `s3:PutObjectLegalHold` permission.

Legal holds are independent from retention periods. As long as the bucket that contains the object has Object Lock enabled, you can place and remove legal holds regardless of whether the specified object version has a retention period set. Placing a legal hold on an object version doesn't affect the retention mode or retention period for that object version.

For example, suppose that you place a legal hold on an object version while the object version is also protected by a retention period. If the retention period expires, the object doesn't lose its WORM protection. Rather, the legal hold continues to protect the object until an authorized user explicitly removes it. Similarly, if you remove a legal hold while an object version has a retention period in effect, the object version remains protected until the retention period expires.

Hence, the correct answer is: **Enable S3 Object Lock in compliance mode with a retention period of one year.**

The option that says: **Enable S3 Object Lock in governance mode with a retention period of one year** is incorrect because in the governance mode, users can't overwrite or delete an object version or alter its lock settings unless they have special permissions or if a user has access to the root AWS user account. A better option to choose here is to use the compliance mode.

The option that says: **Enable S3 Object Lock in governance mode with a legal hold of one year** is incorrect. You cannot set a time period for a legal hold. You can only do this using the "retention period" option. Take note that a legal hold will still restrict users from changing the S3 objects even after the one-year retention period has elapsed. In addition, a governance mode will allow the root user to modify your S3 objects and override any existing settings.

The option that says: **Enable S3 Object Lock in compliance mode with a legal hold of one year** is incorrect. Although the choice of using the compliance mode is right, you still cannot set a one-year time period for the legal hold option. Keep in mind that the legal hold is independent of the retention period.

References:

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock-overview.html

Check out this Amazon S3 Cheat Sheet:

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

9. QUESTION

Category: CSAA – Design High-Performing Architectures

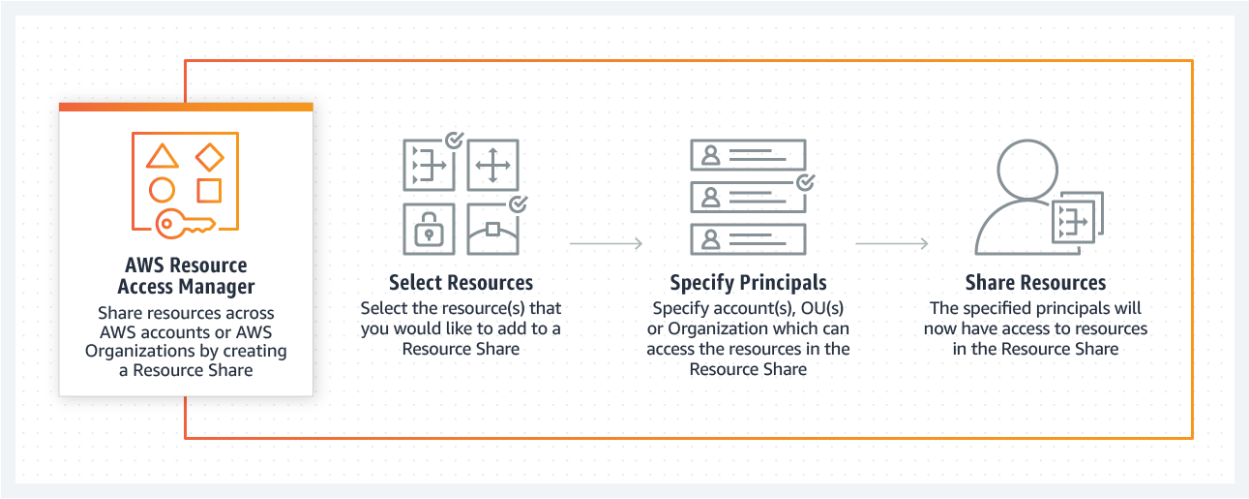
A global IT company with offices around the world has multiple AWS accounts. To improve efficiency and drive costs down, the Chief Information Officer (CIO) wants to set up a solution that centrally manages their AWS resources. This will allow them to procure AWS resources centrally and share resources 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.)

- **Consolidate all of the company's accounts using AWS Organizations.** (Correct)
- **Use AWS Control Tower to easily and securely share your resources with your AWS accounts.**
- **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.**
- **Consolidate all of the company's accounts using AWS ParallelCluster.**
- **Use the AWS Resource Access Manager (RAM) service to easily and securely share your resources with your AWS accounts.** (Correct)

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. 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

10. 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?

- **Disk Reads activity of an EC2 instance**
- **CPU Utilization of an EC2 instance**

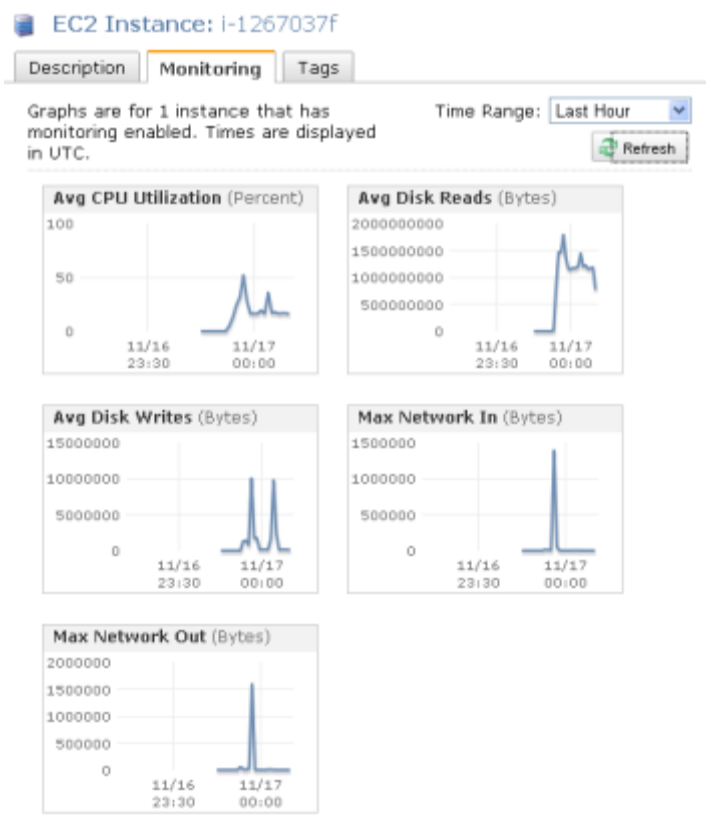
Network packets out of an EC2 instance

Memory Utilization of an EC2 instance (Correct)

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/

11. QUESTION

Category: CSAA – Design Secure Architectures

A company uses an Application Load Balancer (ALB) for its public-facing multi-tier web applications. The security team has recently reported that there has been a surge of SQL injection attacks lately, which causes critical data discrepancy issues. The same issue is also encountered by its other web applications in other AWS accounts that are behind an ALB. An immediate solution is required to prevent the remote injection of unauthorized SQL queries and protect their applications hosted across multiple accounts.

As a Solutions Architect, what solution would you recommend?

Use AWS Network Firewall to filter web vulnerabilities and brute force attacks using stateful rule groups across all Application Load Balancers on all AWS accounts. Refactor the web application to be less susceptible to SQL injection attacks based on the security assessment.

Use Amazon Macie to scan for vulnerabilities and unintended network exposure. Refactor the web application to be less susceptible to SQL injection attacks based on the security assessment. Utilize the AWS Audit Manager to reuse the security assessment across all AWS accounts.

Use Amazon GuardDuty and set up a managed rule to block request patterns associated with the exploitation of SQL databases, like SQL injection attacks. Associate it with the Application Load Balancer and utilize the AWS Security Hub service to reuse the managed rules across all the AWS accounts

Use AWS WAF and set up a managed rule to block request patterns associated with the exploitation of SQL databases, like SQL injection attacks. Associate it with the Application Load Balancer. Integrate AWS WAF with AWS Firewall Manager to reuse the rules across all the AWS accounts. (Correct)

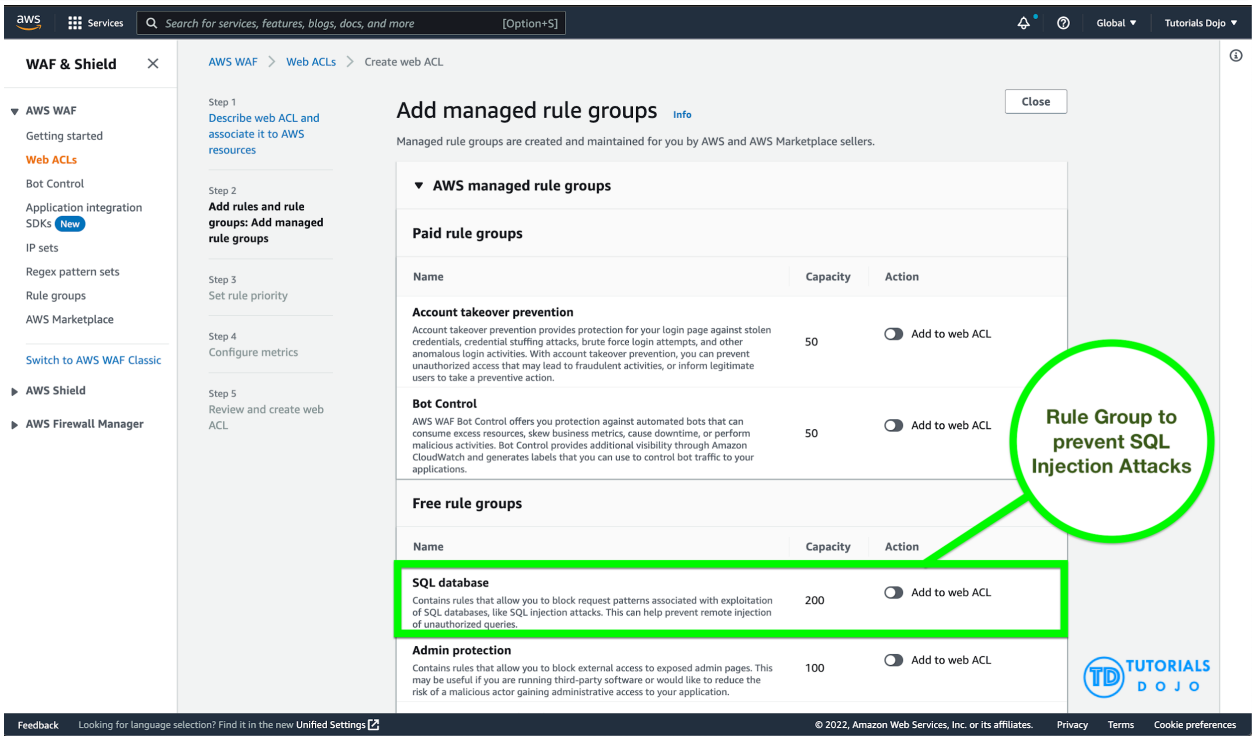
AWS WAF is a web application firewall that lets you monitor the HTTP(S) requests that are forwarded to an Amazon CloudFront distribution, an Amazon API Gateway REST API, an Application Load Balancer, or an AWS AppSync GraphQL API.

-Web ACLs – You use a web access control list (ACL) to protect a set of AWS resources. You create a web ACL and define its protection strategy by adding rules. Rules define criteria for inspecting web requests and specify how to handle requests that match the criteria. You set a default action for the web ACL that indicates whether to block or allow through those requests that pass the rules inspections.

-Rules – Each rule contains a statement that defines the inspection criteria and an action to take if a web request meets the criteria. When a web request meets the criteria, that's a match. You can configure rules to block matching requests, allow them through, count them, or run CAPTCHA controls against them.

-Rules groups – You can use rules individually or in reusable rule groups. AWS Managed Rules and AWS Marketplace sellers provide managed rule groups for your use. You can also define your own rule groups.

AWSManagedRulesSQLiRuleSet – The SQL database rule group contains rules to block request patterns associated with the exploitation of SQL databases, like SQL injection attacks. This can help prevent remote injection of unauthorized queries. Evaluate this rule group for use if your application interfaces with an SQL database.



AWS WAF is easy to deploy and protect applications deployed on either Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts

all your origin servers, Amazon API Gateway for your REST APIs, or AWS AppSync for your GraphQL APIs. There is no additional software to deploy, DNS configuration, SSL/TLS certificate to manage, or need for a reverse proxy setup.

With AWS Firewall Manager integration, you can centrally define and manage your rules and reuse them across all the web applications that you need to protect.

Therefore, the correct answer is: **Use AWS WAF and set up a managed rule to block request patterns associated with the exploitation of SQL databases, like SQL injection attacks. Associate it with the Application Load Balancer. Integrate AWS WAF with AWS Firewall Manager to reuse the rules across all the AWS accounts.**

The option that says: **Use Amazon GuardDuty and set up a managed rule to block request patterns associated with the exploitation of SQL databases, like SQL injection attacks. Associate it with the Application Load Balancer and utilize the AWS Security Hub service to reuse the managed rules across all the AWS accounts**is incorrect because Amazon GuardDuty is only a threat detection service and cannot directly be integrated with the Application Load Balancer.

The options that says: **Use AWS Network Firewall to filter web vulnerabilities and brute force attacks using stateful rule groups across all Application Load Balancers on all AWS accounts. Refactor the web application to be less susceptible to SQL injection attacks based on the security assessment**is incorrect because AWS Network Firewall is a managed service that is primarily used to deploy essential network protections for all of your Amazon Virtual Private Clouds (VPCs) and not particularly to your Application Load Balancers. Take note that the AWS Network Firewall is account-specific by default and needs to be integrated with the AWS Firewall Manager to easily share the firewall across your other AWS accounts. In addition, refactoring the web application will require an immense amount of time.

The options that says: **Use Amazon Macie to scan for vulnerabilities and unintended network exposure. Refactor the web application to be less susceptible to SQL injection attacks based on the security assessment. Utilize the AWS Audit Manager to reuse the security assessment across all AWS accounts** is incorrect because Amazon Macie is only used for data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data. Just like before, refactoring the web application will require an immense amount of time. The use of the AWS Audit Manager is not relevant as well. The AWS Audit Manager simply helps you continually audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/how-aws-waf-works.html

https://docs.aws.amazon.com/waf/latest/developerguide/fms-chapter.html

https://docs.aws.amazon.com/waf/latest/developerguide/aws-managed-rule-groups-use-case.html#aws-managed-rule-groups-use-case-sql-db

Check out this AWS Web Application Firewall Cheat Sheet:

https://tutorialsdojo.com/aws-waf

12. QUESTION

Category: CSAA – Design Secure 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.

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. (Correct)

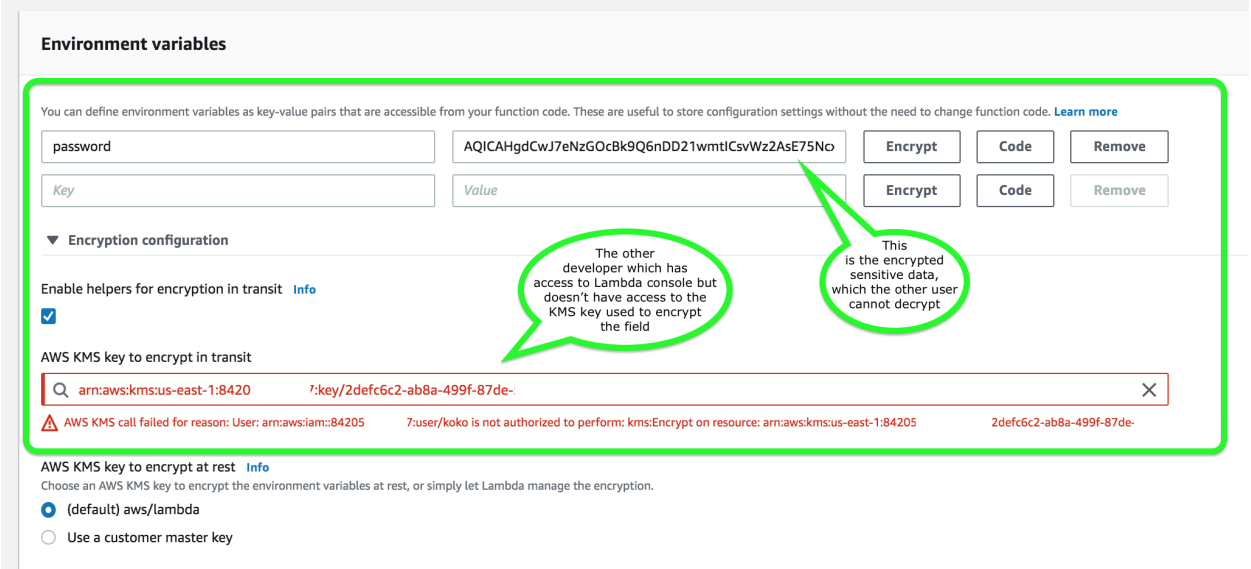
AWS Lambda does not provide encryption for the environment variables. Deploy your code to an EC2 instance instead.

Enable SSL encryption that leverages on AWS CloudHSM to store and encrypt the sensitive information.

There is no need to do anything because, by default, AWS Lambda already encrypts the environment variables using the AWS Key Management Service.

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 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

13. 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.

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.

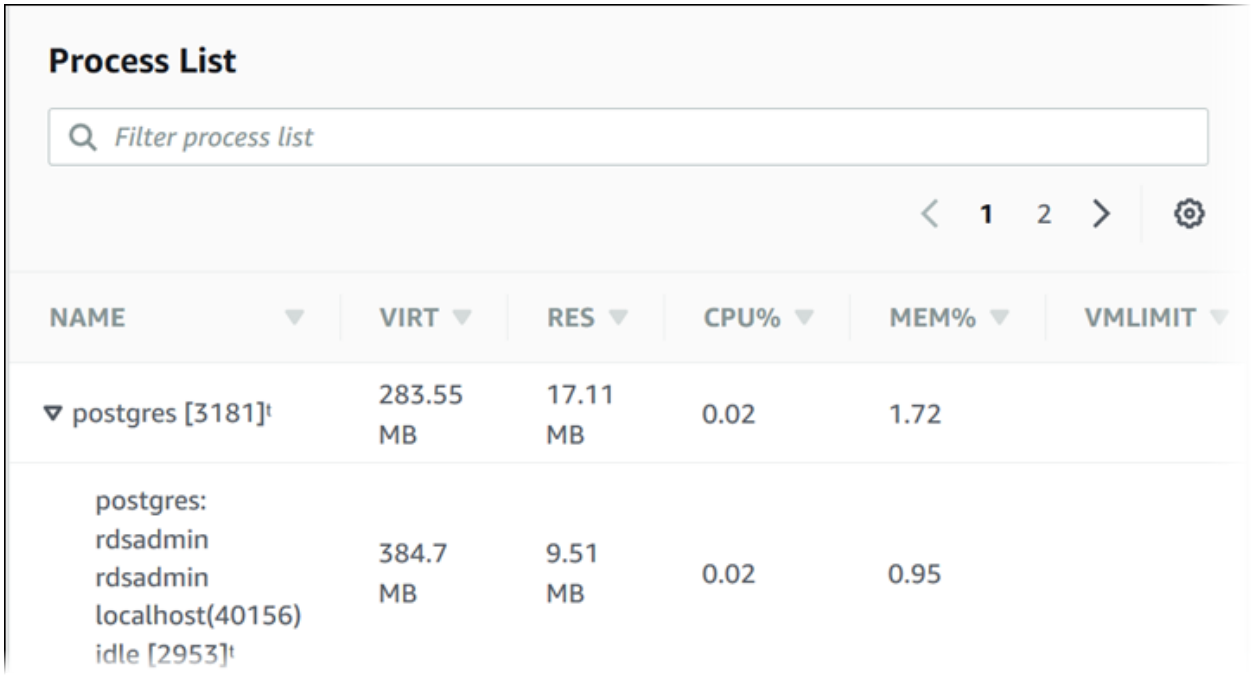
Enable Enhanced Monitoring in RDS. (Correct)

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.

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.



Using Amazon CloudWatch to monitor the CPU Utilization of your database is incorrect. 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. 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-cloudwatch

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/

14. QUESTION

Category: CSAA – Design High-Performing Architectures

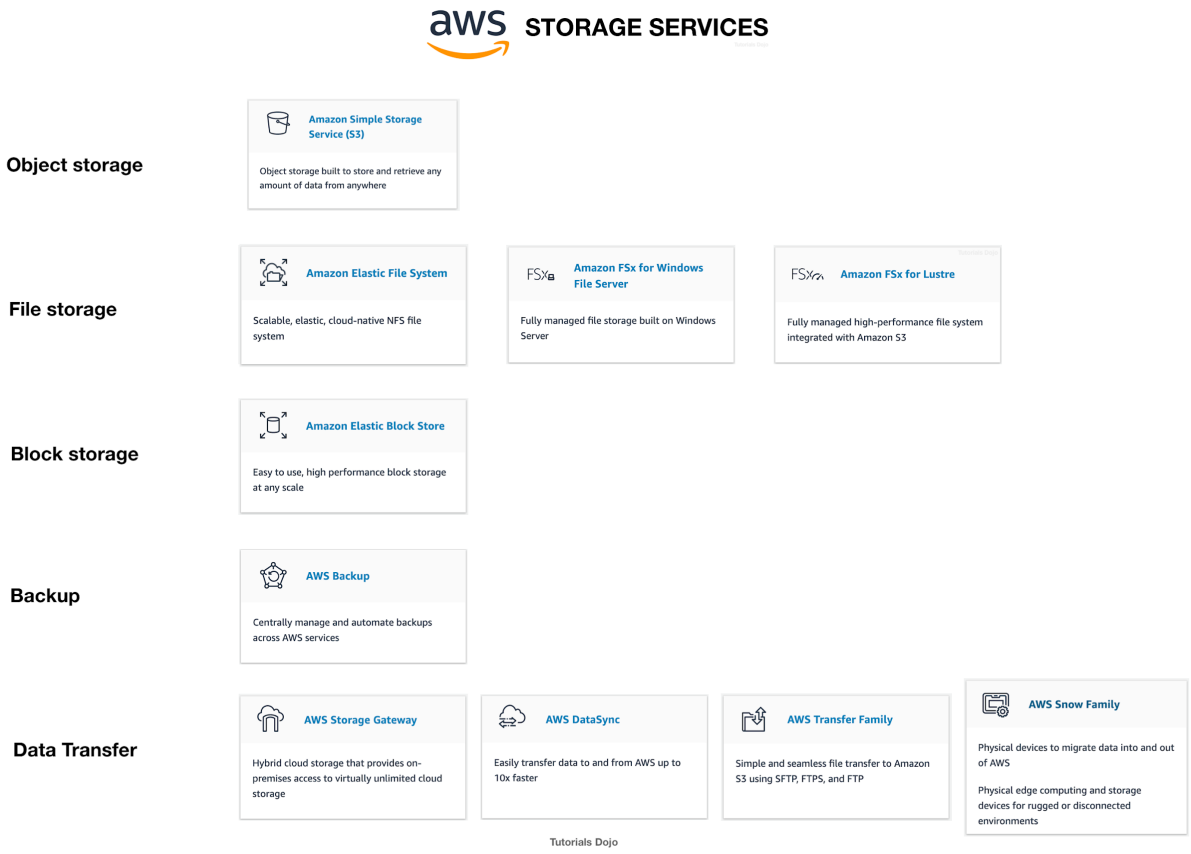
An AI-powered Forex trading application consumes thousands of data sets to train its machine learning model. The application's workload requires a high-performance, parallel hot storage to process the training datasets concurrently. It also needs cost-effective cold storage to archive those datasets that yield low profit.

Which of the following Amazon storage services should the developer use?

- **Use Amazon FSx For Lustre and Amazon 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 FSx For Lustre and Amazon S3 for hot and cold storage respectively.** (Correct)

Use Amazon Elastic File System and Amazon S3 for hot and cold storage respectively.

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.



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 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 for storing hot data (data that are frequently accessed) used in I/O-intensive workloads. EBS has a storage option called "Cold HDD," but due to its price, it is not ideal for data archiving. EBS Cold HDD is much more expensive than Amazon S3 Glacier / Glacier Deep Archive and is often utilized in applications where sequential cold data is read less frequently.

Using Amazon Elastic File System and Amazon S3 for hot and cold storage respectively is incorrect. 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/

15. QUESTION

Category: CSAA – Design Resilient Architectures

A logistics company plans to automate its order management application. The company wants to use SFTP file transfer in uploading business-critical documents. Since the files are confidential, the files need to be highly available and must be encrypted at rest. The files must also be automatically deleted a month after they are created.

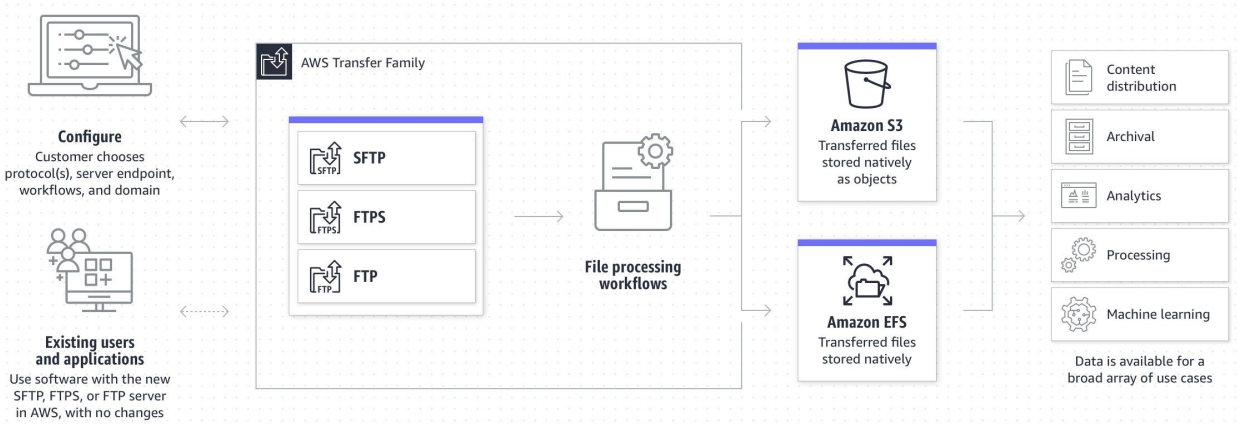
Which of the following options should be implemented to meet the company requirements with the least operation overhead?

- **Create an Amazon Elastic Filesystem (EFS) file system and enable encryption. Configure AWS Transfer for SFTP to securely upload files to the EFS file system. Apply an EFS lifecycle policy to delete files after 30 days.**
- **Create an Amazon S3 bucket with encryption enabled. Configure AWS Transfer for SFTP to securely upload files to the S3 bucket. Configure the retention policy on the SFTP server to delete files after a month.**
- **Create an Amazon S3 bucket with encryption enabled. Launch an AWS Transfer for SFTP endpoint to securely upload files to the S3 bucket. Configure an S3 lifecycle rule to delete files after a month.** (Correct)

Provision an Amazon EC2 instance and install the SFTP service. Mount an encrypted EFS file system on the EC2 instance to store the uploaded files. Add a cron job to delete the files older than a month.

AWS Transfer for SFTP enables you to easily move your file transfer workloads that use the Secure Shell File Transfer Protocol (SFTP) to AWS without needing to modify your applications or manage any SFTP servers.

To get started with AWS Transfer for SFTP (AWS SFTP) you create an SFTP server and map your domain to the server endpoint, select authentication for your SFTP clients using service-managed identities, or integrate your own identity provider, and select your Amazon S3 buckets to store the transferred data. Your existing users can continue to operate with their existing SFTP clients or applications. Data uploaded or downloaded using SFTP is available in your Amazon S3 bucket, and can be used for archiving or processing in AWS.



An Amazon S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

Transition actions – These actions define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after creating them.

Expiration actions – These actions define when objects expire. Amazon S3 deletes expired objects on your behalf.

Therefore, the correct answer is: **Create an Amazon S3 bucket with encryption enabled. Launch an AWS Transfer for SFTP endpoint to securely upload files to the S3 bucket. Configure an S3 lifecycle rule to delete files after a month.** You can use S3 as the storage service for your AWS Transfer SFTP-enabled server.

The option that says: **Create an Amazon S3 bucket with encryption enabled. Configure AWS Transfer for SFTP to securely upload files to the S3 bucket. Configure the retention policy on the SFTP server to delete files after a month** is incorrect. The 30-day retention policy must be configured on the Amazon S3 bucket. There is no retention policy option on AWS Transfer for SFTP.

The option that says: **Create an Amazon Elastic Filesystem (EFS) file system and enable encryption. Configure AWS Transfer for SFTP to securely upload files to the EFS file system. Apply an EFS lifecycle policy to delete files after 30 days** is incorrect. This may be possible, however, the EFS lifecycle management doesn't delete objects. It can only transition files in and out of the "Infrequent Access" tier.

The option that says: **Provision an Amazon EC2 instance and install the SFTP service. Mount an encrypted EFS file system on the EC2 instance to store the uploaded files. Add a cron job to delete the files older than a month** is incorrect. This option is possible however, it entails greater operational overhead since you need to manage the EC2 instance and SFTP service.

References:

https://aws.amazon.com/aws-transfer-family/

https://docs.aws.amazon.com/transfer/latest/userguide/create-server-sftp.html

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html

Check out these AWS Transfer Family and Amazon S3 Cheat Sheets:

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

https://tutorialsdojo.com/aws-transfer-family/

16. QUESTION

Category: CSAA – Design Secure 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/*"
  }
 ]
}
```

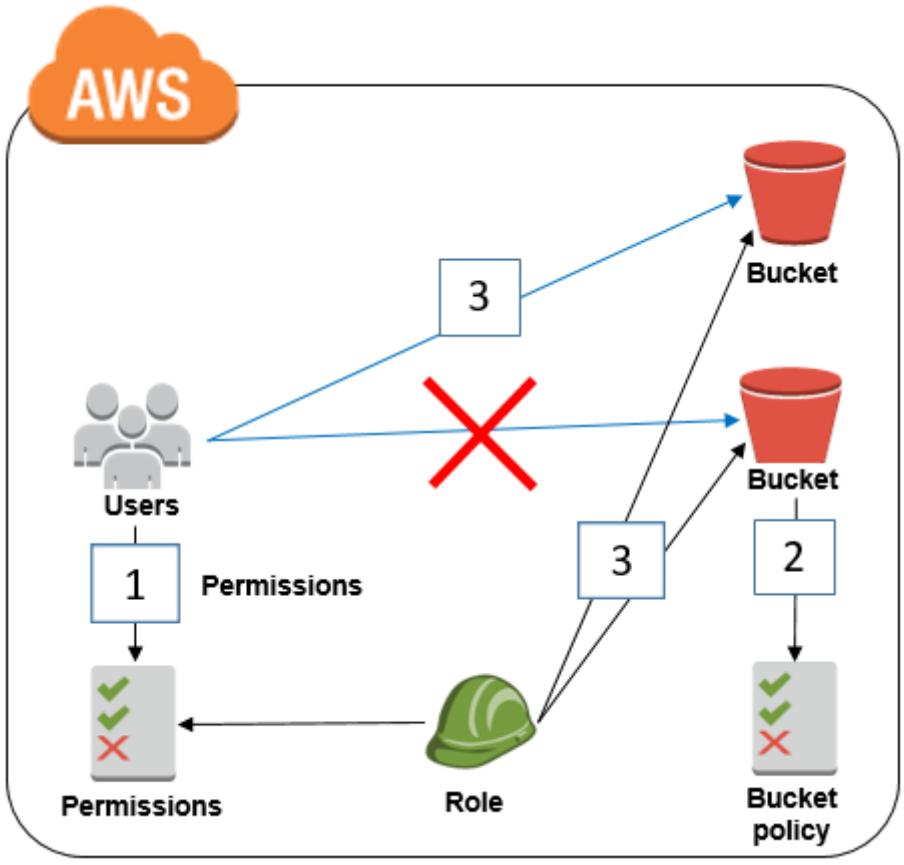
What does the above IAM policy allow? (Select THREE.)

- **An IAM user with this IAM policy is allowed to read and delete objects from the `boracay` S3 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.** (Correct)
- **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 read objects from the `boracay` S3 bucket.** (Correct)

An IAM user with this IAM policy is allowed to read objects from all S3 buckets owned by the account. (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/

17. QUESTION

Category: CSAA – Design Resilient Architectures

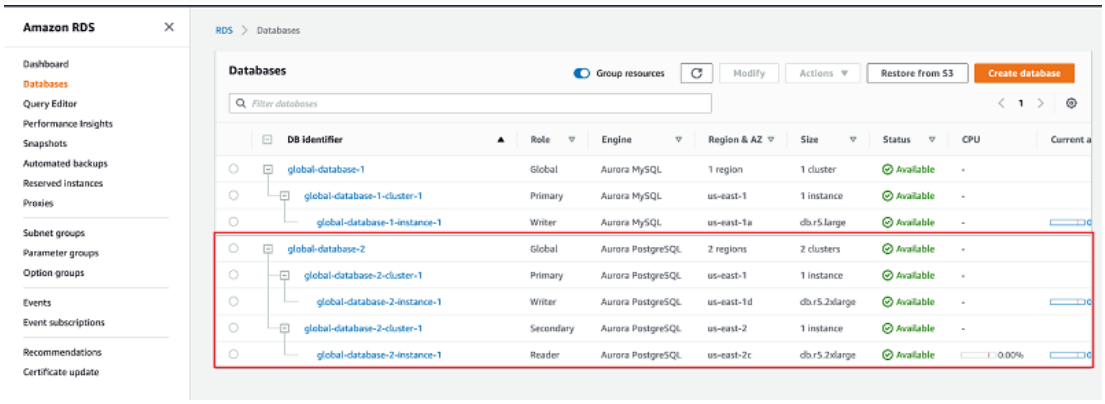
A Solutions Architect needs to set up a relational database and come up with a disaster recovery plan to mitigate multi-region failure. The solution requires a Recovery Point Objective (RPO) of 1 second and a Recovery Time Objective (RTO) of less than 1 minute.

Which of the following AWS services can fulfill this requirement?

- **Amazon Quantum Ledger Database (Amazon QLDB)**
- **Amazon RDS for PostgreSQL with cross-region read replicas**
- **Amazon Timestream**
- **Amazon Aurora Global Database** (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.



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 provide you a strong foundation for a global business continuity plan.

Hence, the correct answer is: **Amazon Aurora Global Database**.

Amazon Quantum Ledger Database (Amazon QLDB) is incorrect because it is stated in the scenario that the Solutions Architect needs to create a relational database and not a ledger database. An Amazon Quantum Ledger Database (QLDB) is a fully managed ledger database that provides a transparent, immutable, and cryptographically verifiable transaction log. Moreover, QLDB cannot provide an RPO of 1 second and an RTO of less than 1 minute.

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 cross-region RDS Read Replica replication is not as fast as Amazon Aurora Global Databases.

Amazon Timestream is incorrect because this is a serverless time series database service that is commonly used for IoT and operational applications. The most suitable solution for this scenario is to use the Amazon Aurora Global Database since it can provide the required RPO and RTO.

References:

https://aws.amazon.com/rds/aurora/global-database/

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-global-database.html

18. QUESTION

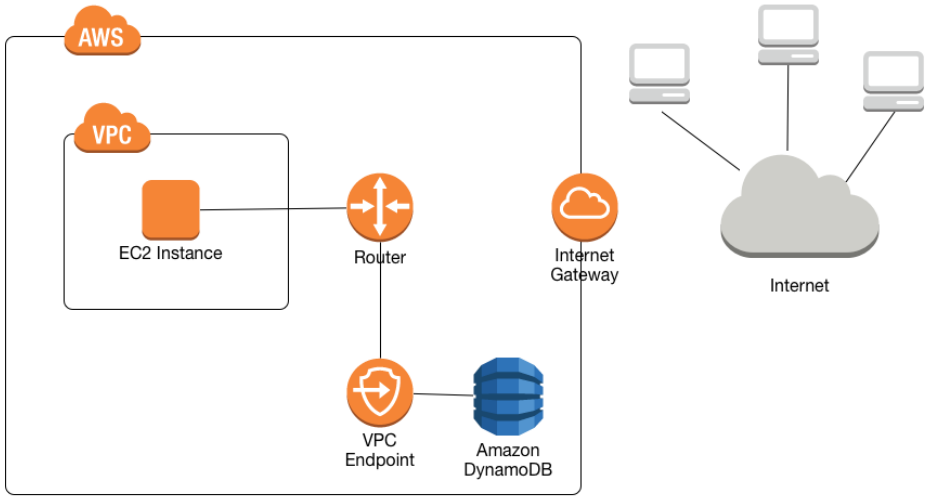
Category: CSAA – Design High-Performing Architectures

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 Direct Connect to route all access to S3 and DynamoDB via private endpoints.**
- **Use AWS VPN CloudHub 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.** (Correct)
- **Use AWS Transit Gateway to route all access to S3 and DynamoDB via private endpoints.**

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 High-Performing Architectures

An online learning company hosts its Microsoft .NET e-Learning application on a Windows Server in its on-premises data center. The application uses an Oracle Database Standard Edition as its backend database.

The company wants a high-performing solution to migrate this workload to the AWS cloud to take advantage of the cloud's high availability. The migration process should minimize development changes, and the environment should be easier to manage.

Which of the following options should be implemented to meet the company requirements? (Select TWO.)

- **Rehost the on-premises .NET application to an AWS Elastic Beanstalk Multi-AZ environment which runs in multiple Availability Zones.** (Correct)
- **Use AWS Application Migration Service (AWS MGN) to migrate the on-premises Oracle database server to a new Amazon EC2 instance.**
- **Refactor the application to .NET Core and run it as a serverless container service using Amazon Elastic Kubernetes Service (Amazon EKS) with AWS Fargate.**

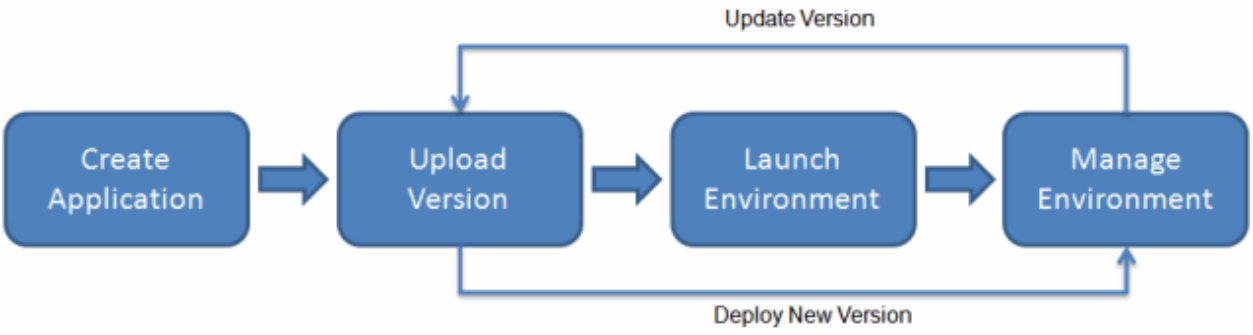
Migrate the Oracle database to Amazon RDS for Oracle in a Multi-AZ deployment by using AWS Database Migration Service (AWS DMS). (Correct)

Provision and replatform the application to Amazon Elastic Container Service (Amazon ECS) with Amazon EC2 worker nodes. Use the Windows Server Amazon Machine Image (AMI) and deploy the .NET application using to the ECS cluster via the Amazon ECS Anywhere service.

AWS Database Migration Service (AWS DMS) is a cloud service that makes it easy to migrate relational databases, data warehouses, NoSQL databases, and other types of data stores. You can use AWS DMS to migrate your data into the AWS Cloud or between combinations of cloud and on-premises setups.

With AWS DMS, you can perform one-time migrations, and you can replicate ongoing changes to keep sources and targets in sync. If you want to migrate to a different database engine, you can use the AWS Schema Conversion Tool (AWS SCT) to translate your database schema to the new platform. You then use AWS DMS to migrate the data.

AWS Elastic Beanstalk reduces management complexity without restricting choice or control. You simply upload your application, and Elastic Beanstalk automatically handles the details of capacity provisioning, load balancing, scaling, and application health monitoring. Elastic Beanstalk supports applications developed in Go, Java, .NET, Node.js, PHP, Python, and Ruby. When you deploy your application, Elastic Beanstalk builds the selected supported platform version and provisions one or more AWS resources, such as Amazon EC2 instances, to run your application.



AWS Elastic Beanstalk for .NET makes it easier to deploy, manage, and scale your ASP.NET web applications that use Amazon Web Services. Elastic Beanstalk for .NET is available to anyone who is developing or hosting a web application that uses IIS.

The option that says: **Migrate the Oracle database to Amazon RDS for Oracle in a Multi-AZ deployment by using AWS Database Migration Service (AWS DMS)** is correct. AWS DMS can help migrate on-premises databases to the AWS Cloud.

The option that says: **Rehost the on-premises .NET application to an AWS Elastic Elastic Beanstalk Multi-AZ environment which runs in multiple Availability Zones** is correct. AWS Beanstalk reduces the operational overhead by taking care of provisioning the needed resources for your application.

The option that says: **Refactor the application to .NET Core and run it as a serverless container service using Amazon Elastic Kubernetes Service (Amazon EKS) with AWS Fargate** is incorrect. This will take significant changes to the application as you will refactor, or do a code change to, the codebase in order for it to become a serverless container application. Remember that the scenario explicitly mentioned that the migration process should minimize development changes. A better solution is to rehost the on-premises .NET application to an AWS Elastic Beanstalk Multi-AZ environment, which doesn't require any code changes.

The option that says: **Use AWS Application Migration Service (AWS MGN) to migrate the on-premises Oracle database server to a new Amazon EC2 instance** is incorrect. Amazon RDS supports standard Oracle databases so it would be better to use AWS DMS for the database migration, not AWS MGN.

The option that says: **Provision and replatform the application to Amazon Elastic Container Service (Amazon ECS) with Amazon EC2 worker nodes. Use the Windows Server Amazon Machine Image (AMI) and deploy the .NET application using to the ECS cluster via the Amazon ECS Anywhere service** is incorrect. This may be possible but not recommended for this scenario because you will have to manage the underlying EC2 instances of your Amazon ECS cluster that will run the application. It would be better to use Elastic Beanstalk to take care of provisioning the resources for your .NET application. Keep in mind that doing a replatform-type migration like this one entails significant development changes, which is not suitable with the requirements given in the scenario.

References:

https://docs.aws.amazon.com/dms/latest/userguide/Welcome.html

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create_deploy_NET.html

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html

Check out these AWS DMS and AWS Beanstalk Cheat Sheets:

https://tutorialsdojo.com/aws-database-migration-service/

https://tutorialsdojo.com/aws-elastic-beanstalk/

20. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

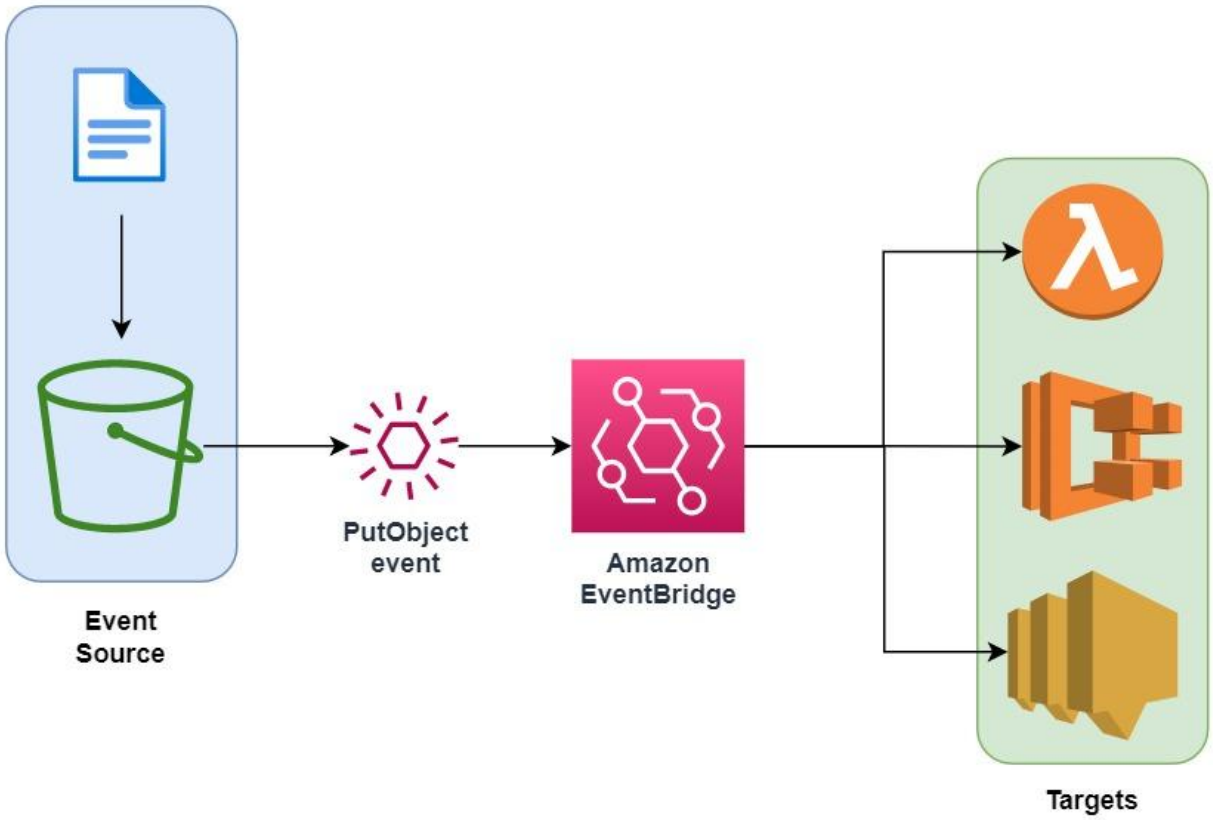
A company is using AWS Fargate to run a batch job whenever an object is uploaded to an Amazon S3 bucket. The minimum ECS task count is initially set to 1 to save on costs and should only be increased based on new objects uploaded to the S3 bucket.

Which is the most suitable option to implement with the LEAST amount of effort?

- **Set up an Amazon EventBridge (Amazon CloudWatch Events) rule to detect S3 object PUT operations and set the target to a Lambda function that will run the `StartTask` API command.**
- **Set up an Amazon EventBridge (Amazon CloudWatch Events) rule to detect S3 object PUT operations and set the target to the ECS cluster to run a new ECS task.** (Correct)
- **Set up an alarm in Amazon CloudWatch to monitor S3 object-level operations that are recorded on CloudTrail. Create an Amazon EventBridge (Amazon CloudWatch Events) rule that triggers the ECS cluster when new CloudTrail events are detected.**
- **Set up an alarm in CloudWatch to monitor S3 object-level operations recorded on CloudTrail. Set two alarm actions to update the ECS task count to scale-out/scale-in depending on the S3 event.**

Amazon EventBridge (Amazon CloudWatch Events) is a serverless event bus that makes it easy to connect applications together. It uses data from your own

applications, integrated software as a service (SaaS) applications, and AWS services. This simplifies the process of building event-driven architectures by decoupling event producers from event consumers. This allows producers and consumers to be scaled, updated, and deployed independently. Loose coupling improves developer agility in addition to application resiliency.



You can use Amazon EventBridge (Amazon CloudWatch Events) to run Amazon ECS tasks when certain AWS events occur. You can set up an EventBridge rule that runs an Amazon ECS task whenever a file is uploaded to a certain Amazon S3 bucket using the Amazon S3 PUT operation.

Hence, the correct answer is: **Set up an Amazon EventBridge (Amazon CloudWatch Events) rule to detect S3 object PUT operations and set the target to the ECS cluster to run a new ECS task.**

The option that says: **Set up an Amazon EventBridge (Amazon CloudWatch Events) rule to detect S3 object PUT operations and set the target to a Lambda function that will run the `StartTask` API command** 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 tasks directly as targets 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 Amazon CloudWatch to monitor S3 object-level operations that are recorded on CloudTrail. Create an Amazon EventBridge (Amazon CloudWatch Events) rule that triggers the ECS cluster when new CloudTrail events are detected** is incorrect because using CloudTrail and CloudWatch Alarm creates an unnecessary complexity to what you want to achieve. Amazon EventBridge (Amazon 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.

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/

21. 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.

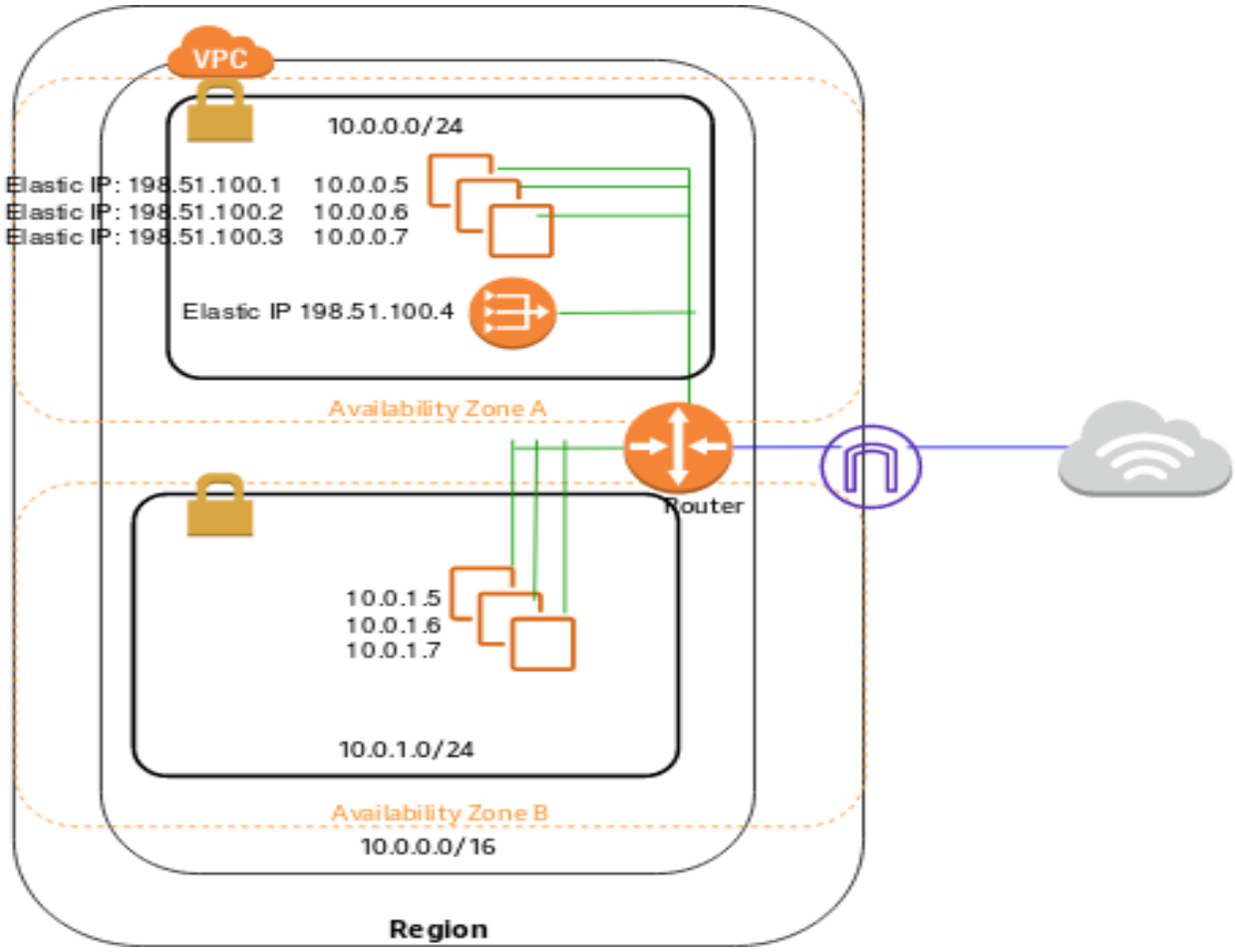
Create two NAT Gateways in each availability zone. Configure the route table in each public subnet to ensure that instances use the NAT Gateway in the same availability zone.

Create 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 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 (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/

22. QUESTION

Category: CSAA – Design Secure Architectures

A payment processing company plans to migrate its on-premises application to an Amazon EC2 instance. An IPv6 CIDR block is attached to the company's Amazon VPC. Strict security policy mandates that the production VPC must only allow outbound communication over IPv6 between the instance and the internet but should prevent the internet from initiating an inbound IPv6 connection. The new architecture should also allow traffic flow inspection and traffic filtering.

What should a solutions architect do to meet these requirements?

Launch the EC2 instance to a private subnet and attach an Egress-Only Internet Gateway to the VPC to allow outbound IPv6 communication to the internet. Use AWS Network Firewall to set up the required rules for traffic inspection and traffic filtering. (Correct)

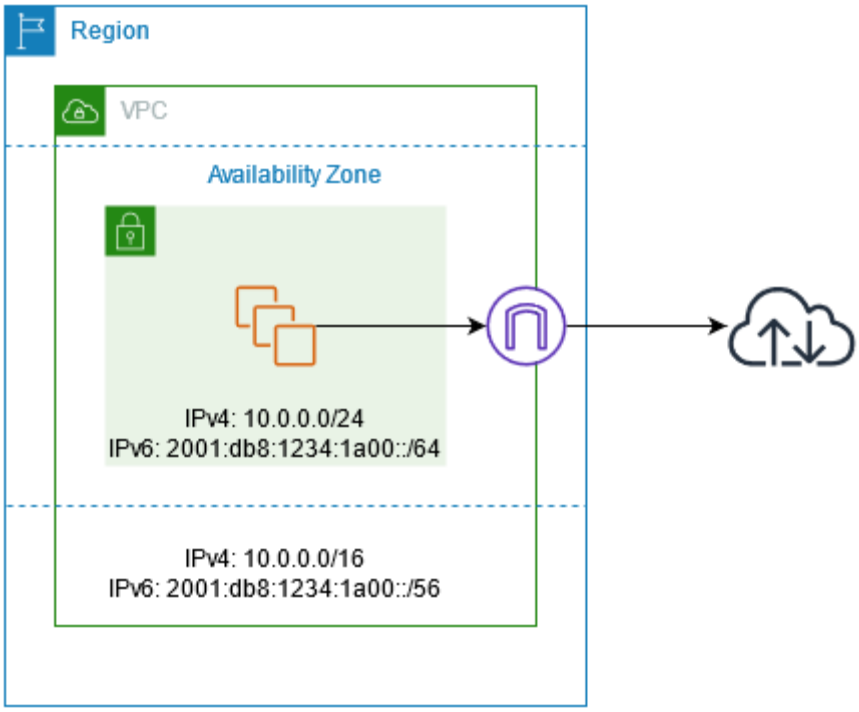
Launch the EC2 instance to a private subnet and attach a NAT Gateway to the VPC to allow outbound IPv6 communication to the

internet. Use AWS Firewall Manager to set up the required rules for traffic inspection and traffic filtering.

Launch the EC2 instance to a private subnet and attach AWS PrivateLink interface endpoint to the VPC to control outbound IPv6 communication to the internet. Use Amazon GuardDuty to set up the required rules for traffic inspection and traffic filtering.

Launch the EC2 instance to a public subnet and attach an Internet Gateway to the VPC to allow outbound IPv6 communication to the internet. Use Traffic Mirroring to set up the required rules for traffic inspection and traffic filtering.

An egress-only internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the internet and prevents it from initiating an IPv6 connection with your instances.

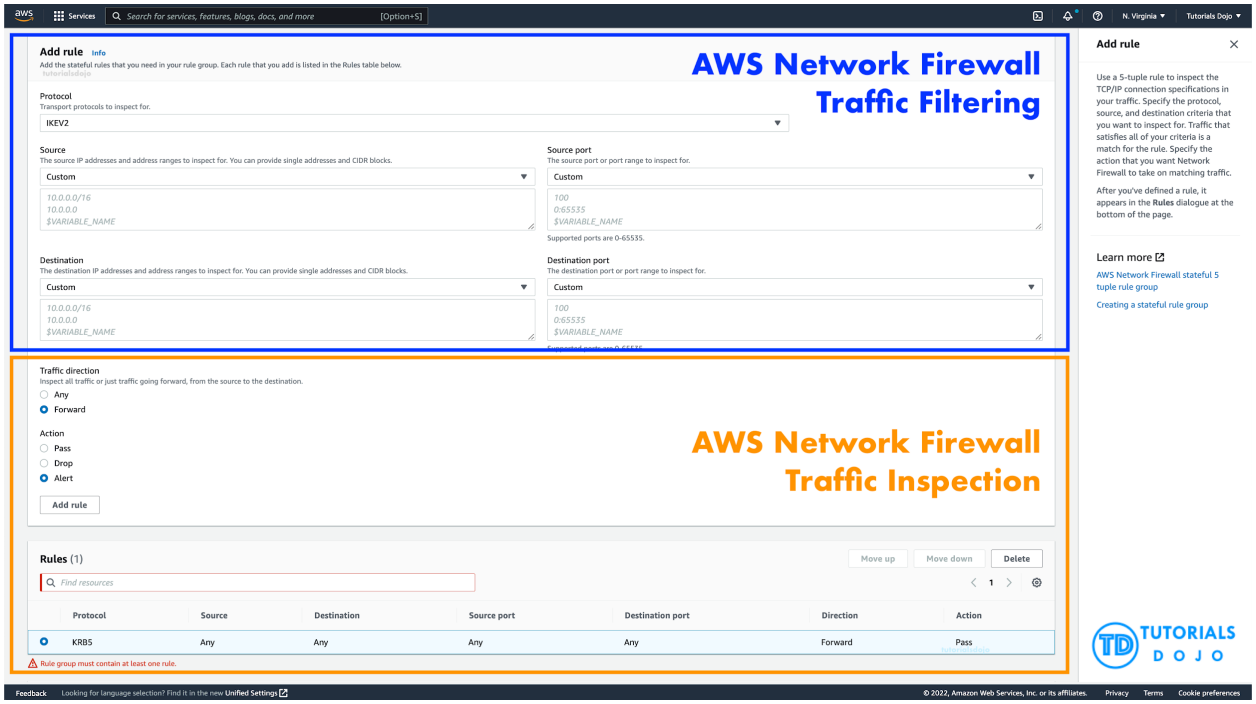


Destination	Target
10.0.0.0/16	Local
2001:db8:1234:1a00:/64	Local
::/0	*eigw-id*

IPv6 addresses are globally unique and are therefore public by default. If you want your instance to be able to access the internet, but you want to prevent resources on the internet from initiating communication with your instance, you can use an egress-only internet gateway.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. Use a public subnet for resources that must be connected to the internet and a private subnet for resources that won't be connected to the internet.

AWS Network Firewall is a managed service that makes it easy to deploy essential network protections for all of your Amazon Virtual Private Clouds (VPCs). The service can be set up with just a few clicks and scales automatically with your network traffic, so you don't have to worry about deploying and managing any infrastructure. AWS Network Firewall includes features that provide protection from common network threats.



AWS Network Firewall's stateful firewall can incorporate context from traffic flows, like tracking connections and protocol identification, to enforce policies such as preventing your VPCs from accessing domains using an unauthorized protocol. AWS Network Firewall's intrusion prevention system (IPS) provides active traffic flow inspection so you can identify and block vulnerability exploits using signature-based detection. AWS Network Firewall also offers web filtering that can stop traffic to known bad URLs and monitor fully qualified domain names.

In this scenario, you can use an egress-only internet gateway to allow outbound IPv6 communication to the internet and then use the AWS Network Firewall to set up the required rules for traffic inspection and traffic filtering.

Hence, the correct answer for the scenario is: **Launch the EC2 instance to a private subnet and attach an Egress-Only Internet Gateway to the VPC to allow outbound IPv6 communication to the internet. Use AWS Network Firewall to set up the required rules for traffic inspection and traffic filtering.**

The option that says: **Launch the EC2 instance to a private subnet and attach AWS PrivateLink interface endpoint to the VPC to control outbound IPv6 communication to the internet. Use Amazon GuardDuty to set up the required rules for traffic inspection and traffic filtering** is incorrect because the AWS PrivateLink (which is also known as VPC Endpoint) is just a highly available, scalable technology that enables you to privately connect your VPC to the AWS services as if they were in your VPC. This service is not capable of controlling outbound IPv6 communication to the Internet. Furthermore, the Amazon GuardDuty service doesn't have the features to do traffic inspection or filtering.

The option that says: **Launch the EC2 instance to a public subnet and attach an Internet Gateway to the VPC to allow outbound IPv6 communication to the internet. Use Traffic Mirroring to set up the required rules for traffic inspection and traffic filtering** is incorrect because an Internet Gateway does not limit or control any outgoing IPv6 connection. Take note that the requirement is to prevent the Internet from initiating an inbound IPv6 connection to your instance. This solution allows all kinds of traffic to initiate a connection to your EC2 instance hence, this option is wrong. In addition, the use of Traffic Mirroring is not appropriate as well. This is just an Amazon VPC feature that you can use to copy network traffic from an elastic network interface of type interface, not to filter or inspect the incoming/outgoing traffic.

The option that says: **Launch the EC2 instance to a private subnet and attach a NAT Gateway to the VPC to allow outbound IPv6 communication to the internet. Use AWS Firewall Manager to set up the required rules for traffic inspection and traffic filtering** is incorrect. While NAT Gateway has a NAT64 feature that translates an IPv6 address to IPv4, it will not prevent inbound IPv6 traffic from reaching the EC2 instance. You have to use the egress-only Internet Gateway instead. Moreover, the AWS Firewall Manager is neither capable of doing traffic inspection nor traffic filtering.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/egress-only-internet-gateway.html

https://docs.aws.amazon.com/vpc/latest/userguide/configure-subnets.html

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Internet_Gateway.html

Check out this Amazon VPC Cheat Sheet:

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

23. QUESTION

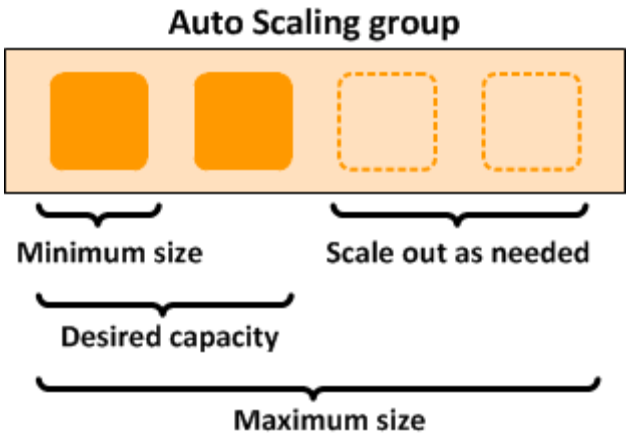
Category: CSAA – Design Resilient Architectures

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 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B. (Correct)
- 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.

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/

24. QUESTION

Category: CSAA – Design Secure 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 KMS-managed customer master key.**
- **Use S3 server-side encryption with a KMS managed key.**
- **Use S3 client-side encryption with a client-side master key.** (Correct)

Use S3 server-side encryption with customer provided key.

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.

✅ Correct Answer: Client-Side Encryption with Client-Side Master Key
This method means:

You encrypt the data before uploading it to S3.

You manage the encryption keys entirely outside AWS.

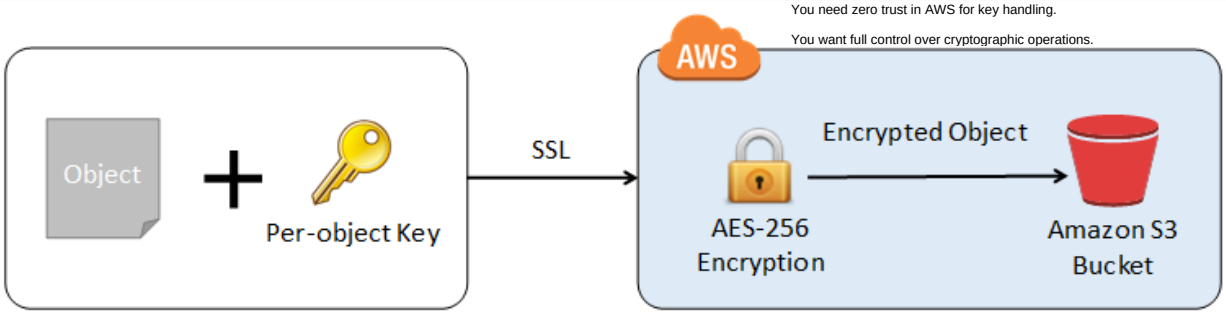
AWS only stores the already encrypted data, and has zero access to the keys.

This satisfies the requirement that nothing sensitive is exposed to AWS, not even the key material.

Is this outdated or still valid?
No, it's still valid and relevant in 2025. While AWS has introduced more advanced key management options (like AWS KMS with external key stores or CloudHSM), the client-side encryption model remains the gold standard when:

You need zero trust in AWS for key handling.

You want full control over cryptographic operations.



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/UsingClientSide

25. QUESTION

Category: CSAA – Design Resilient Architectures

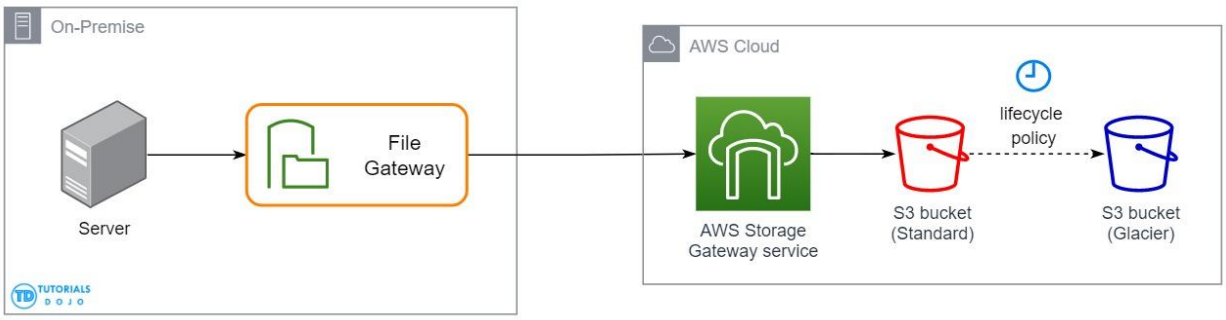
A company has a hybrid cloud architecture that connects their on-premises data center and cloud infrastructure in AWS. They require a durable storage backup for their corporate documents stored on-premises and a local cache that provides low latency access to their recently accessed data to reduce data 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?

- 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.
- 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. (Correct)
- 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.

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/ (Correct)

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

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

26. QUESTION

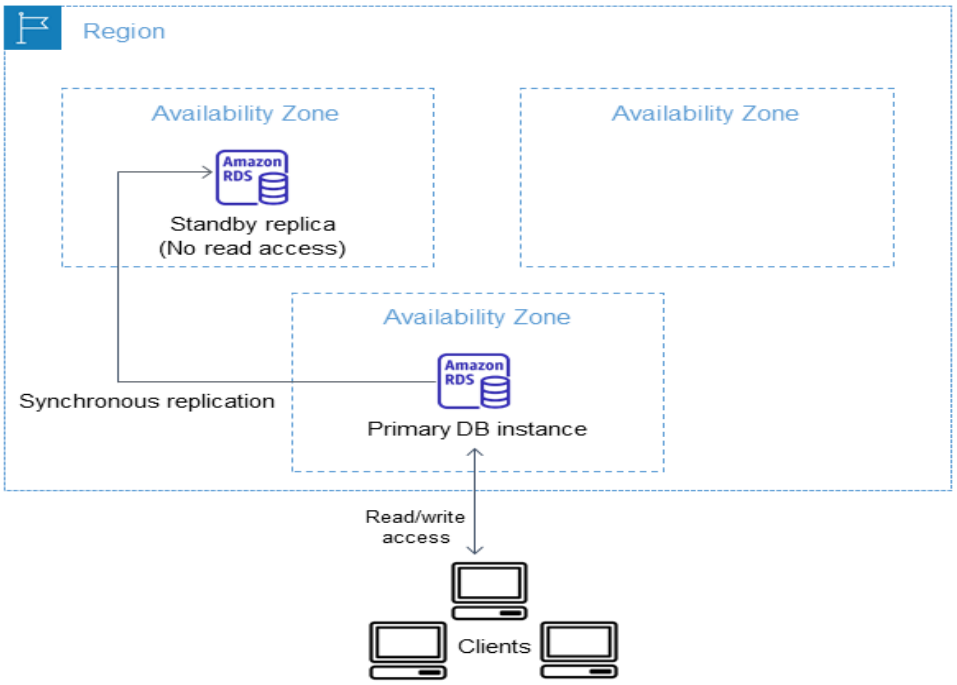
Category: CSAA – Design Resilient Architectures

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 combination of actions would meet the requirement? (Select TWO.)

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

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. You can use AWS DMS to move the on-premises database to AWS with minimal downtime and zero data loss. It supports over 20 engines, including Oracle to Aurora MySQL, MySQL to RDS for MySQL, SQL Server to Aurora PostgreSQL, MongoDB to DocumentDB, Oracle to Redshift, and S3.

Hence, the correct answers are:

– Create an Oracle database in RDS with Multi-AZ deployments.

– Migrate the Oracle database to AWS using the AWS Database Migration Service.

The option that says: **Launching an Oracle database instance in RDS with Recovery Manager (RMAN)** is incorrect because Oracle RMAN is not supported in RDS.

The option that says: **Convert the database schema using the AWS Schema Conversion Tool** is incorrect. AWS Schema Conversion Tool is typically used for heterogeneous migrations where you're moving from one type of database to another (e.g., Oracle to PostgreSQL). In the scenario, the migration is homogenous,

meaning it's an Oracle-to-Oracle migration. As a result, there's no need to convert the schema since you're staying within the same database type.

The option that says: **Migrate the Oracle database to a non-cluster Amazon Aurora with a single instance** is incorrect. While a single-instance Aurora can be a feasible solution for non-critical applications or environments like development or testing, it's not suitable for applications that demand high availability.

References:

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

https://aws.amazon.com/dms/

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/

27. 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?

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

28. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

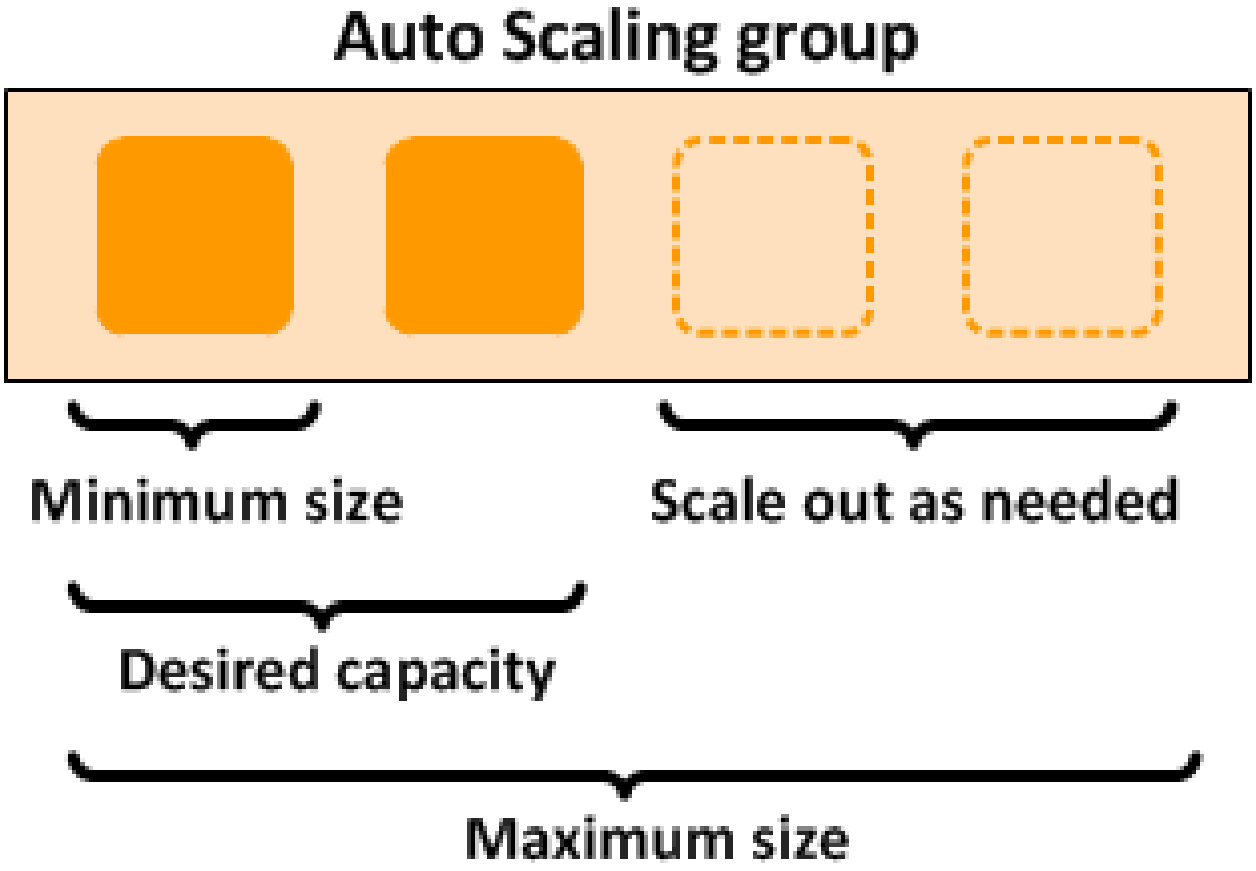
A company hosted a web application in an Auto Scaling group of EC2 instances. The IT manager is concerned about the over-provisioning of the resources that can cause higher operating costs. A Solutions Architect has been instructed to create a cost-effective solution without affecting the performance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

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

- **Use target tracking scaling.** (Correct)
- **Use suspend and resume scaling.**
- **Use scheduled scaling.**
- **Use simple scaling.**

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/

29. 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.** (Correct)
- **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.

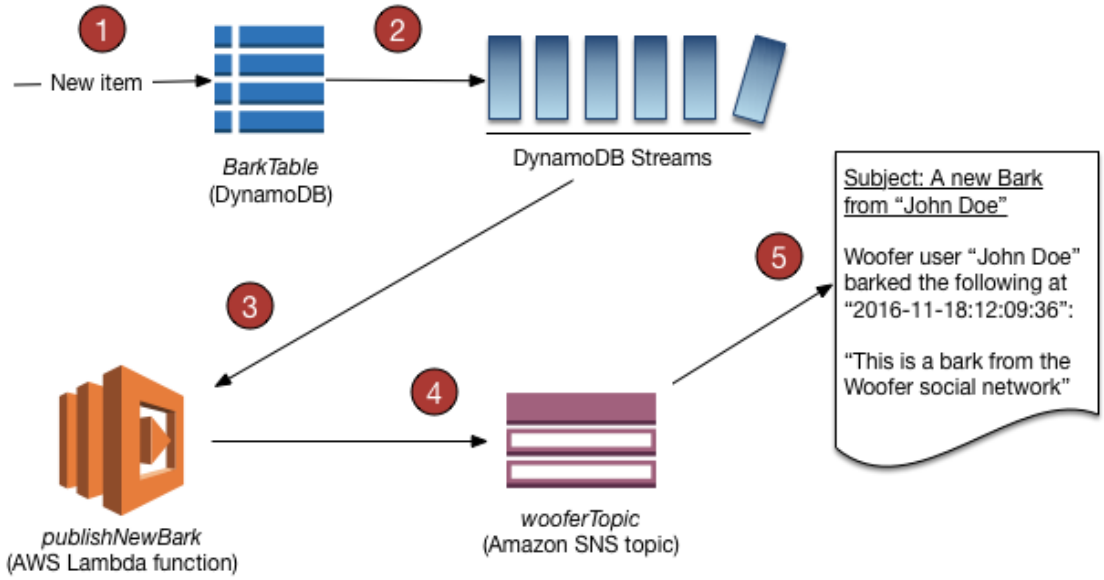
A DynamoDB stream is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

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. 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/

30. QUESTION

~~Category: CSAA – Design Secure Architectures~~

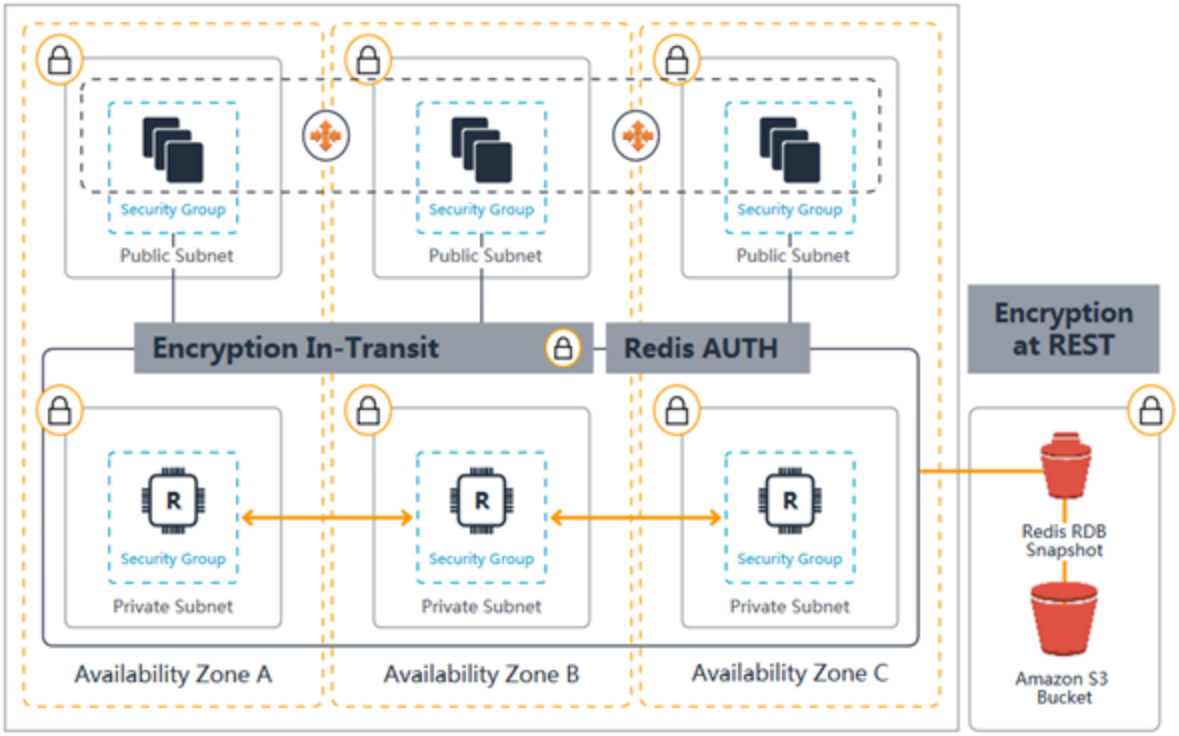
~~A company is designing a banking portal that uses Amazon ElastiCache for Redis as its distributed session management component. Since the other Cloud Engineers in your department have access to your ElastiCache cluster, you have to secure the session data in the portal by requiring them to enter a 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?~~

- **Enable the in-transit encryption for Redis replication groups.**
- **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.**
- **Authenticate the users using Redis AUTH by creating a new Redis Cluster with both the `--transit-encryption-enabled` and `--auth-token` parameters enabled.** (Correct)
- **Set up a Redis replication group and enable the `AtRestEncryptionEnabled` parameter.**

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/

31. 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 launched from the oldest launch configuration** (Correct)
- **The EC2 instance which has been running for the longest time**
- **The EC2 instance which has the least number of user sessions**

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:

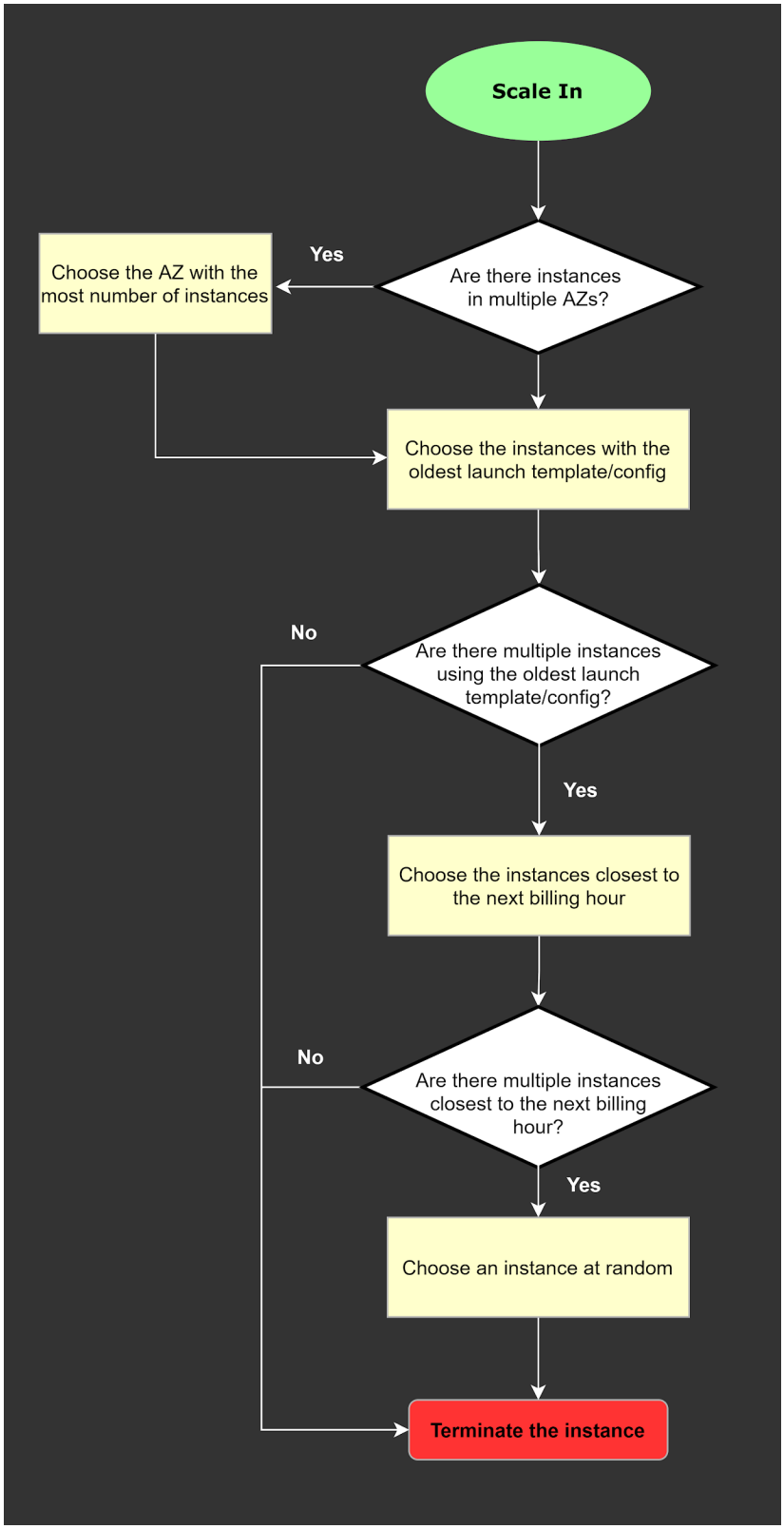
1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that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


Scale In
Are there instances in multiple AZs?
Yes
Choose the AZ with the most number of instances
Choose the instances with the oldest launch template/config
Are there multiple instances using the oldest launch template/config?
No
Yes
Choose the instances closest to the next billing hour
Are there multiple instances closest to the next billing hour?
No
Yes
Choose an instance at random
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/

32. QUESTION

Category: CSAA – Design Secure 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.

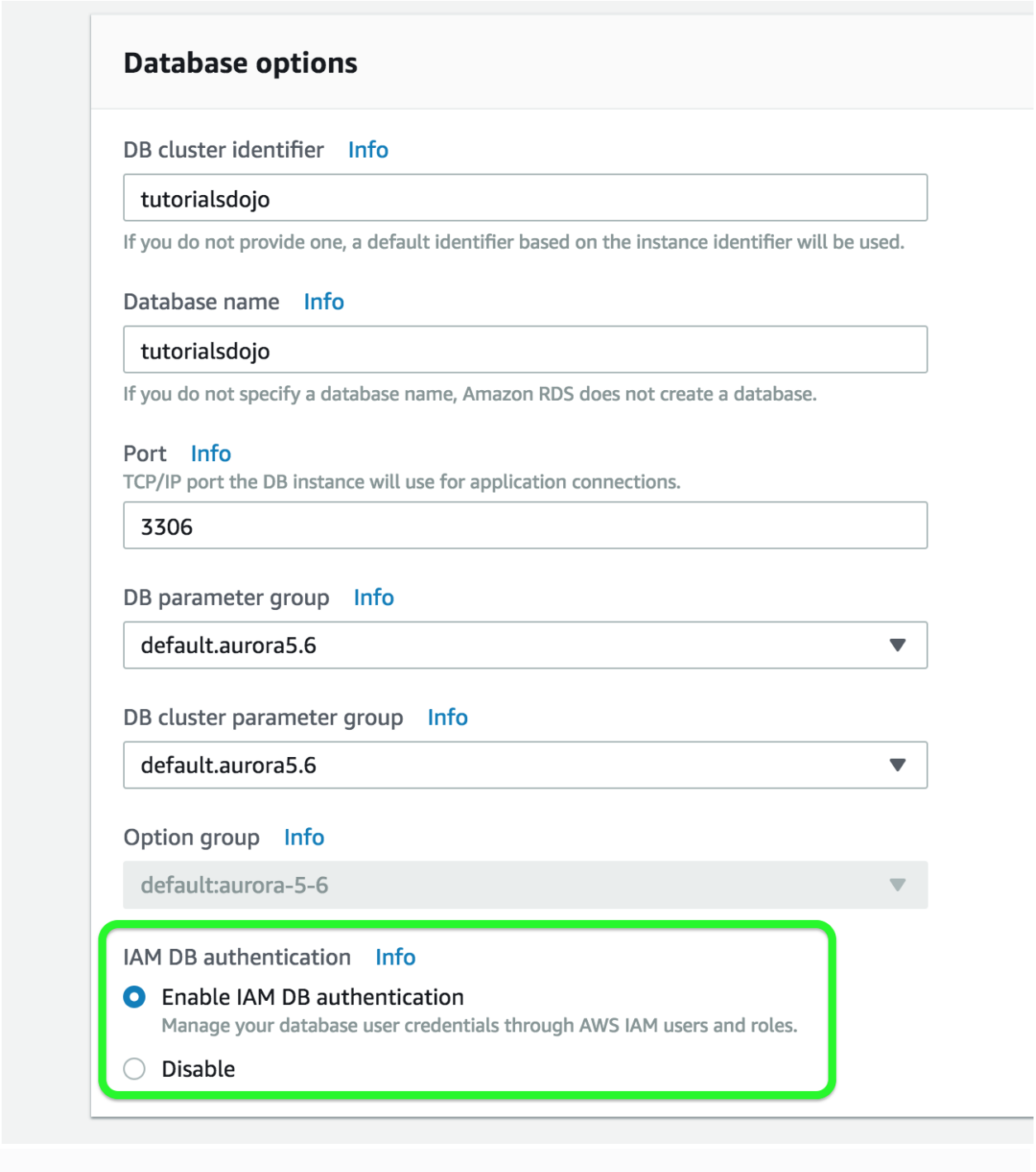
As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- **Enable the IAM DB Authentication.** (Correct)
- **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.**

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.



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/

33. QUESTION

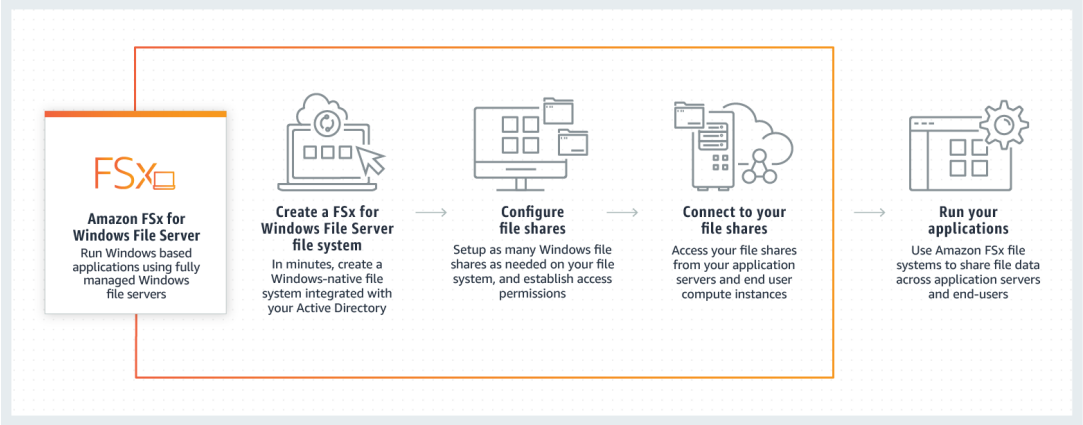
Category: CSAA – Design High-Performing Architectures

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 AWS Storage Gateway.**
- **Migrate the existing file share configuration to Amazon FSx for Windows File Server.** (Correct)
- **Migrate the existing file share configuration to Amazon EFS.**
- **Migrate the existing file share configuration to Amazon EBS.**

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/

34. 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?

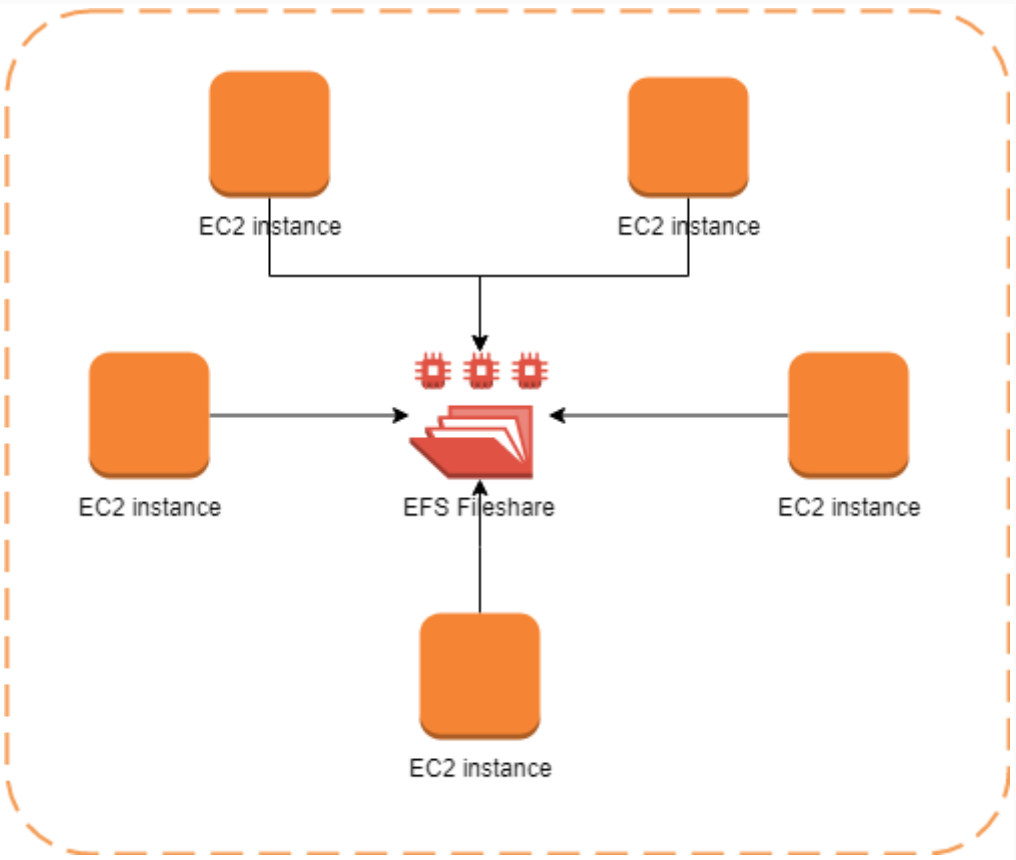
- **Create an S3 bucket and use this as the storage for the CMS**
- **Use EFS** (Correct)
- **Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes**
- **Use ElastiCache**

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/

35. QUESTION

Category: CSAA – Design Resilient Architectures

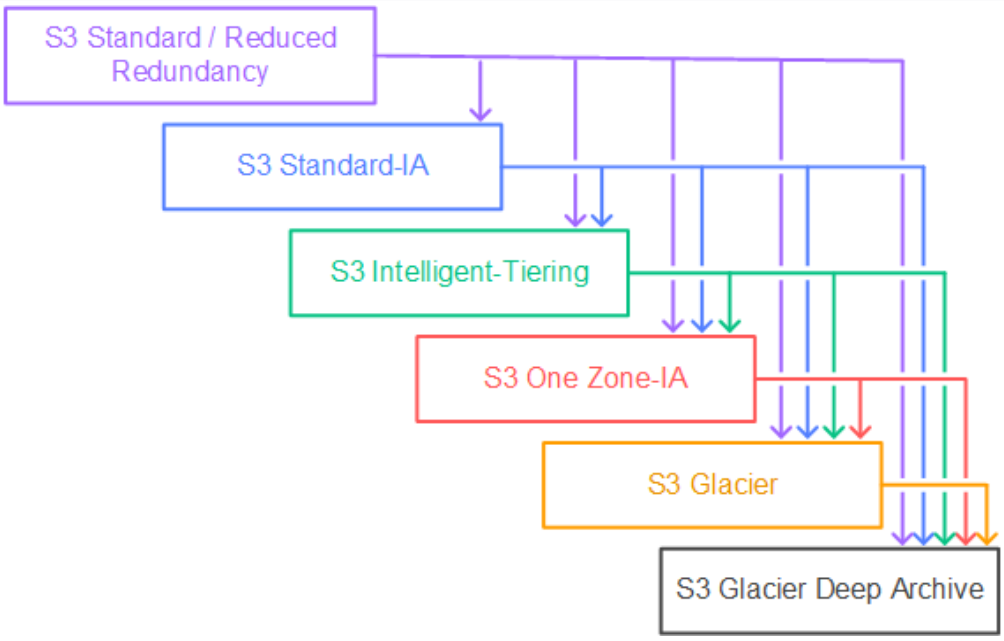
A company plans to host a web application in an Auto Scaling group of Amazon EC2 instances. The application will be used globally by users to upload and store several types of files. Based on user trends, files that are older than 2 years must be stored in a different storage class. The Solutions Architect 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.** (Correct)
- **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 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 S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier 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/

36. 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 AWS Lambda. Configure the Lambda function to fan out the event notifications to multiple Amazon SQS queues to update the processing system.

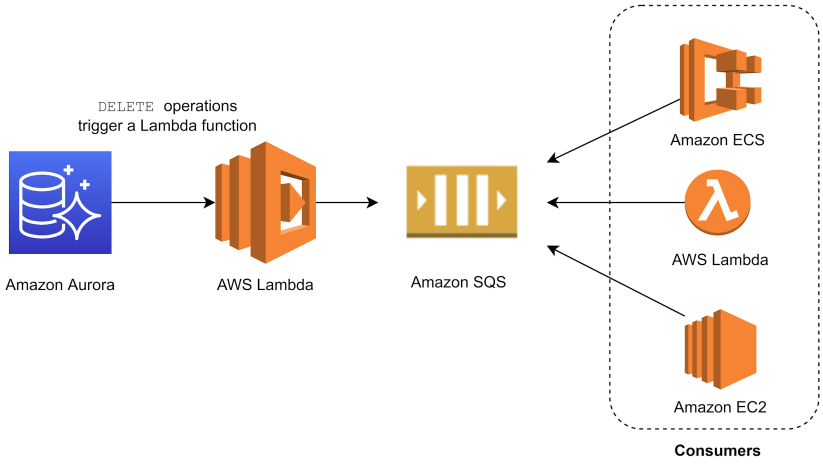
Create an RDS event subscription and send the notifications to AWS Lambda. Configure the Lambda function to fanout the event

notifications to multiple Amazon SQS queues to update the target groups.

Create an RDS event subscription and send the notifications to 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.~~ (Correct)

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/

37. QUESTION

Category: CSAA – Design Secure Architectures

A business has recently migrated its applications to AWS. The audit team must be able to assess whether the services the company is using meet common security and regulatory standards. A solutions architect needs to provide the team with a report of all compliance-related documents for their account.

Which action should a solutions architect consider?

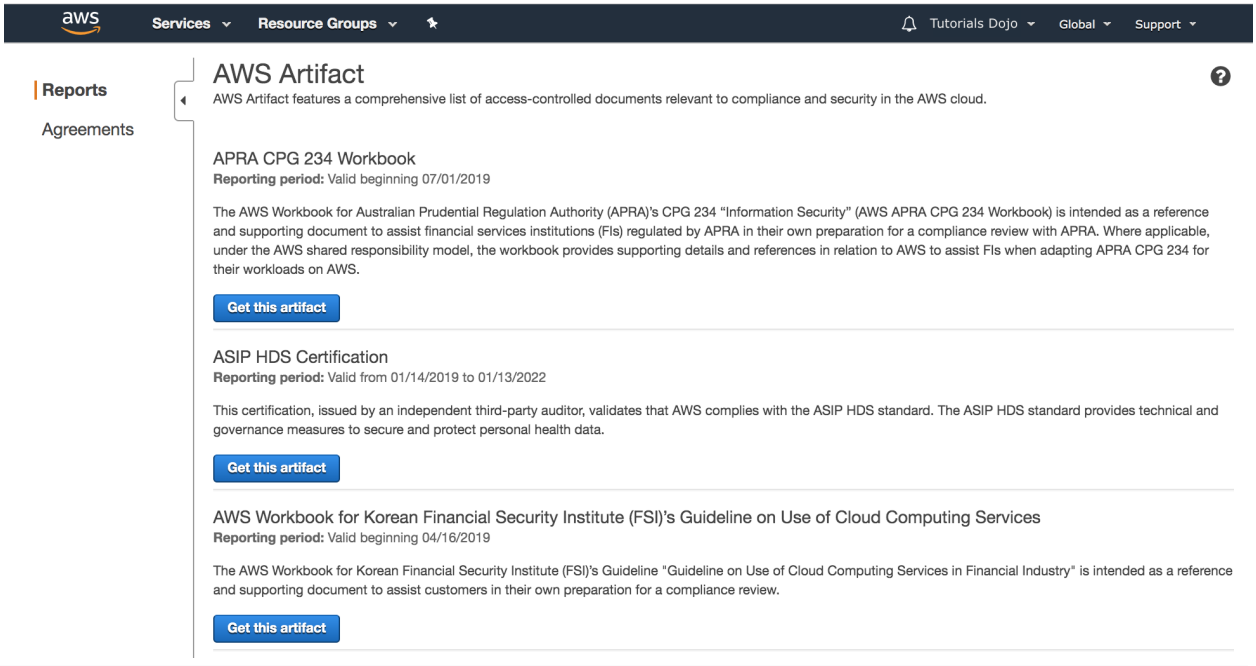
Use AWS Artifact to view the security reports as well as other AWS compliance-related information. (Correct)

View all of the AWS security compliance reports from AWS Security Hub.

Run an Amazon Inspector assessment job to download all of the AWS compliance-related information.

Run an Amazon Macie job to view the Service Organization Control (SOC), Payment Card Industry (PCI), and other compliance reports from AWS Certificate Manager (ACM).

AWS Artifact is your go-to, central resource for compliance-related information that matters to you. It provides on-demand access to AWS' security and compliance reports and select online agreements. Reports available in AWS Artifact include our Service Organization Control (SOC) reports, Payment Card Industry (PCI) reports, and certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of AWS security controls. Agreements available in AWS Artifact include the Business Associate Addendum (BAA) and the Nondisclosure Agreement (NDA).



All AWS Accounts have access to AWS Artifact. Root users and IAM users with admin permissions can download all audit artifacts available to their accounts by agreeing to the associated terms and conditions. You will need to grant IAM users with non-admin permissions access to AWS Artifact using IAM permissions. This

allows you to grant a user access to AWS Artifact while restricting access to other services and resources within your AWS Account.

Hence, the correct answer in this scenario is: **Use AWS Artifact to view the security reports as well as other AWS compliance-related information.**

The option that says: **Run an Amazon Inspector assessment job to download all of the AWS compliance-related information** is incorrect. Amazon Inspector is simply a security tool for detecting vulnerabilities in AWS workloads. For this scenario, it is better to use the readily-available security reports in AWS Artifact instead.

The option that says: **Run an Amazon Macie job to view the Service Organization Control (SOC), Payment Card Industry (PCI), and other compliance reports from AWS Certificate Manager (ACM)** is incorrect because ACM is just a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with AWS services and your internal connected resources. This service does not store certifications or compliance-related documents.

The option that says: **View all of the AWS security compliance reports from AWS Security Hub** is incorrect because AWS Security Hub only provides you a comprehensive view of your high-priority security alerts and security posture across your AWS accounts.

References:

https://aws.amazon.com/artifact/getting-started/

https://docs.aws.amazon.com/artifact/latest/ug/what-is-aws-artifact.html

38. QUESTION

Category: CSAA – Design Resilient Architectures

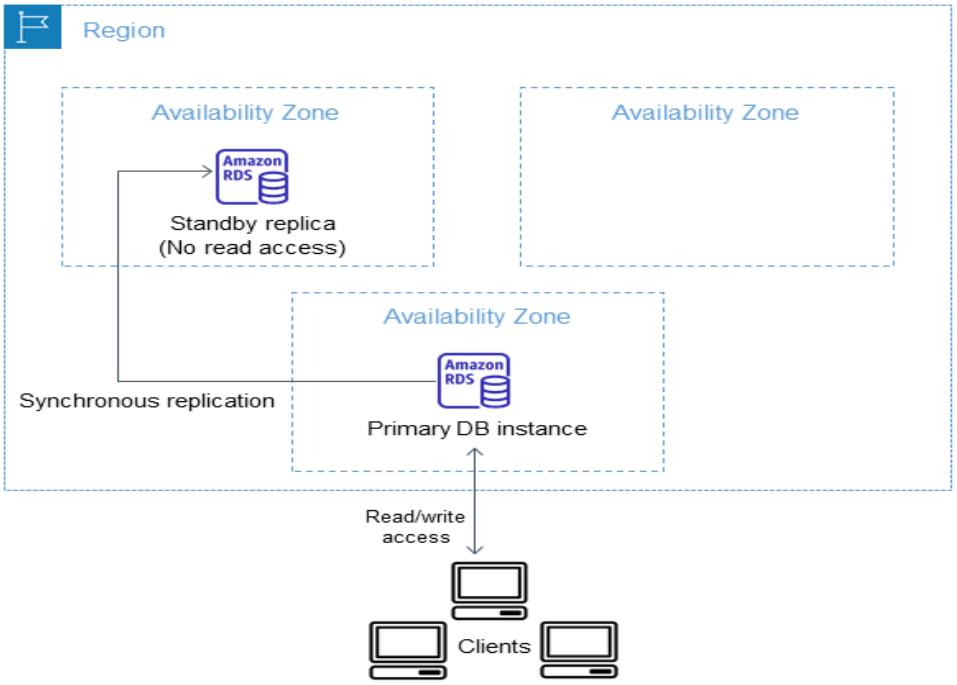
There are a lot of outages in the Availability Zone of your RDS database instance to the point that you have lost access to the database. What could you do to prevent losing access to your database in case that this event happens again?

- **Enable Multi-AZ failover** (Correct)
- **Create a read replica**

Make a snapshot of the database

Increase the database instance size

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-saa-c02/

39. 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?

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. (Correct)

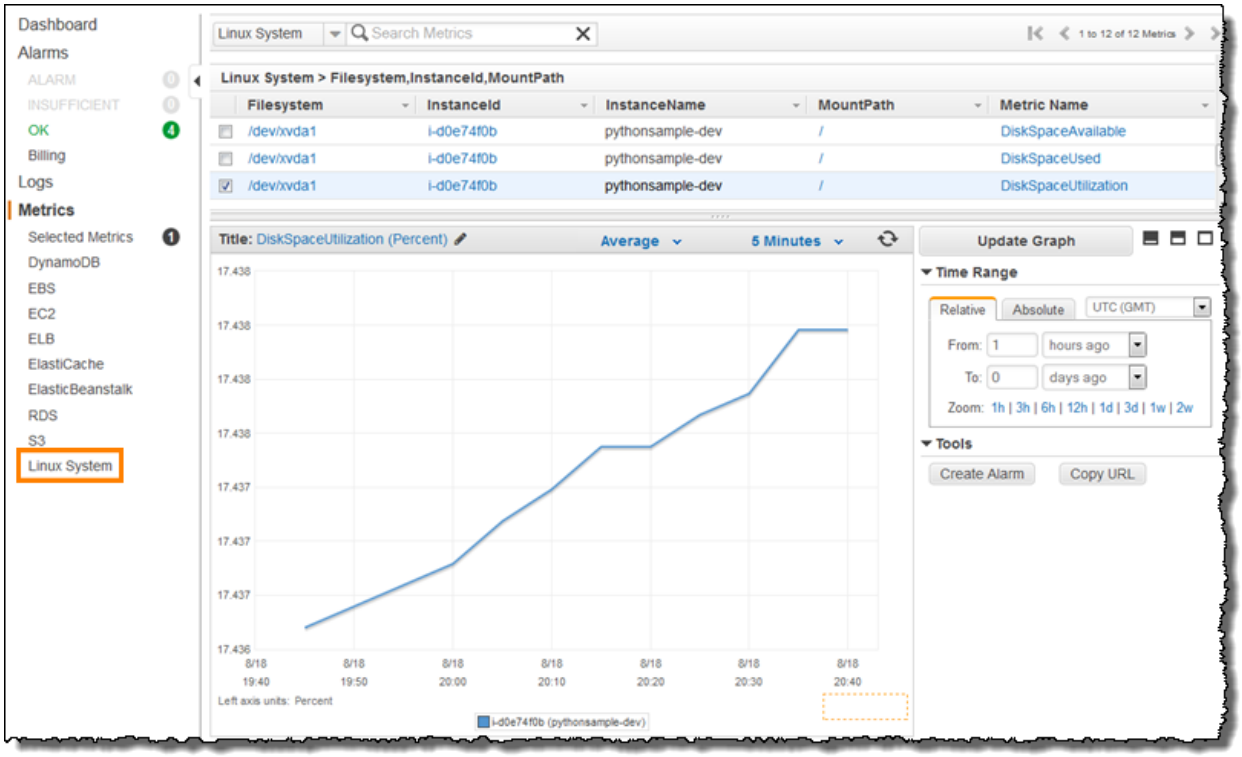
Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard.

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.

Use Amazon Inspector and install the Inspector agent to all EC2 instances.

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/

40. QUESTION

Category: CSAA – Design Secure 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?

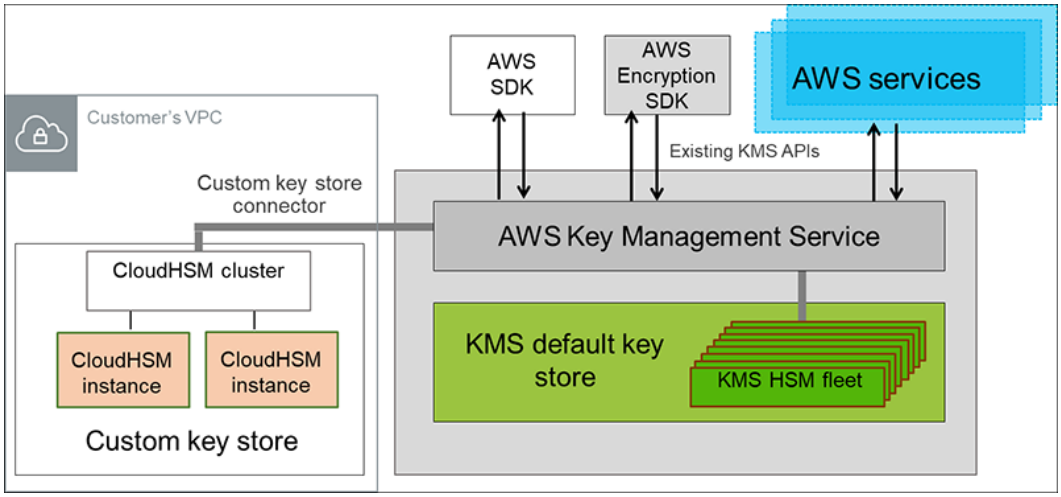
Use AWS Key Management Service to create AWS-owned CMKs and store the non-extractable key material in AWS CloudHSM.

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-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 AWS CloudHSM. (Correct)

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/

41. QUESTION

Category: CSAA – Design High-Performing Architectures

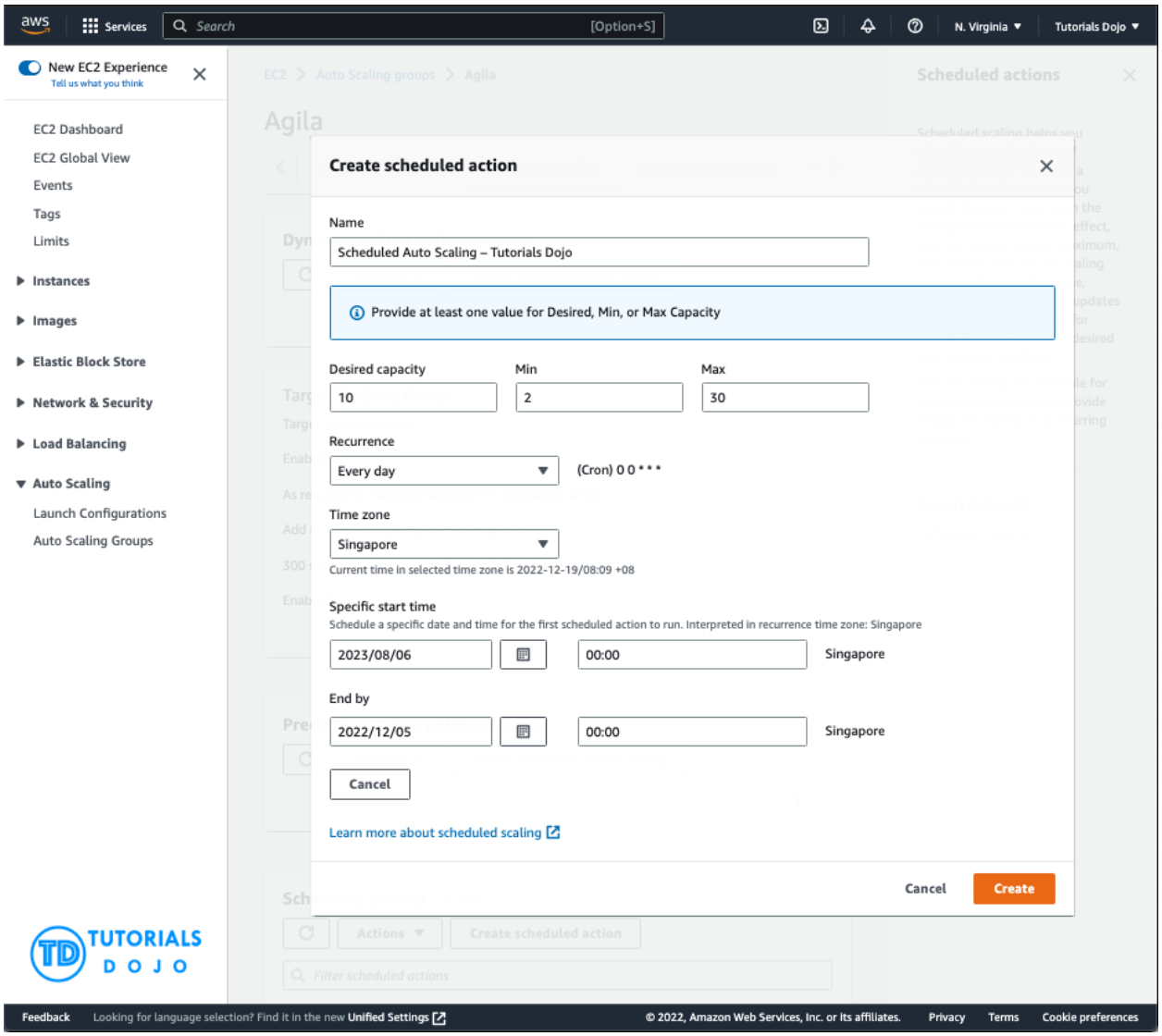
A tech company has a CRM application hosted on an Auto Scaling group of On-Demand EC2 instances with different instance types and sizes. The application is extensively used during office hours from 9 in the morning to 5 in the afternoon. Their users are complaining that the performance of the application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

Which of the following is the MOST operationally efficient solution to implement to ensure the application works properly at the beginning of the day?

- **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)
- **Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization.**
- **Configure a Predictive scaling policy for the Auto Scaling group to automatically adjust the number of Amazon EC2 instances**

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.

The following options are both incorrect. 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:

-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

The option that says: **Configure a Predictive scaling policy for the Auto Scaling group to automatically adjust the number of Amazon EC2 instances** is incorrect. Although this type of scaling policy can be used in this scenario, it is not the most operationally efficient option. Take note that the scenario mentioned that the Auto Scaling group consists of Amazon EC2 instances with different instance types and sizes. Predictive scaling assumes that your Auto Scaling group is homogenous, which means that all EC2 instances are of equal capacity. The forecasted capacity can be inaccurate if you are using a variety of EC2 instance sizes and types on your Auto Scaling group.

References:

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

https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-scheduled-scaling.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/ec2-auto-scaling-predictive-scaling.html#predictive-scaling-limitations

Check out this AWS Auto Scaling Cheat Sheet:

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

42. QUESTION

Category: CSAA – Design Secure 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?

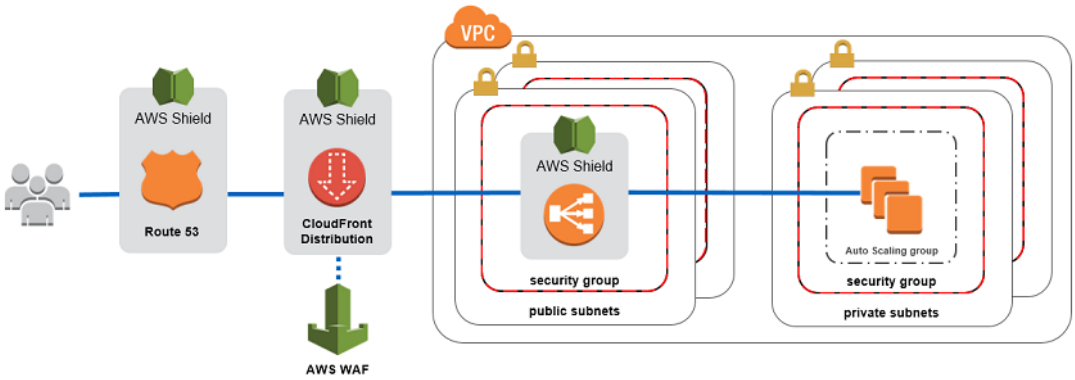
A combination of Security Groups and Network Access Control Lists to only allow authorized traffic to access your VPC.

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 24×7 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

43. QUESTION

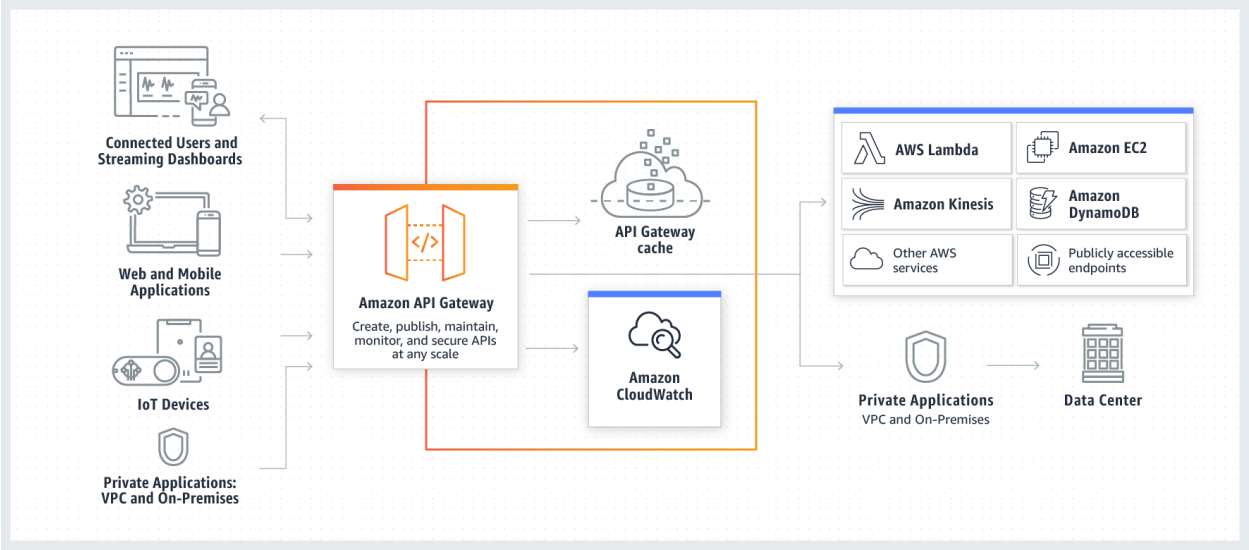
Category: CSAA – Design High-Performing Architectures

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** (Correct)

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/

44. QUESTION

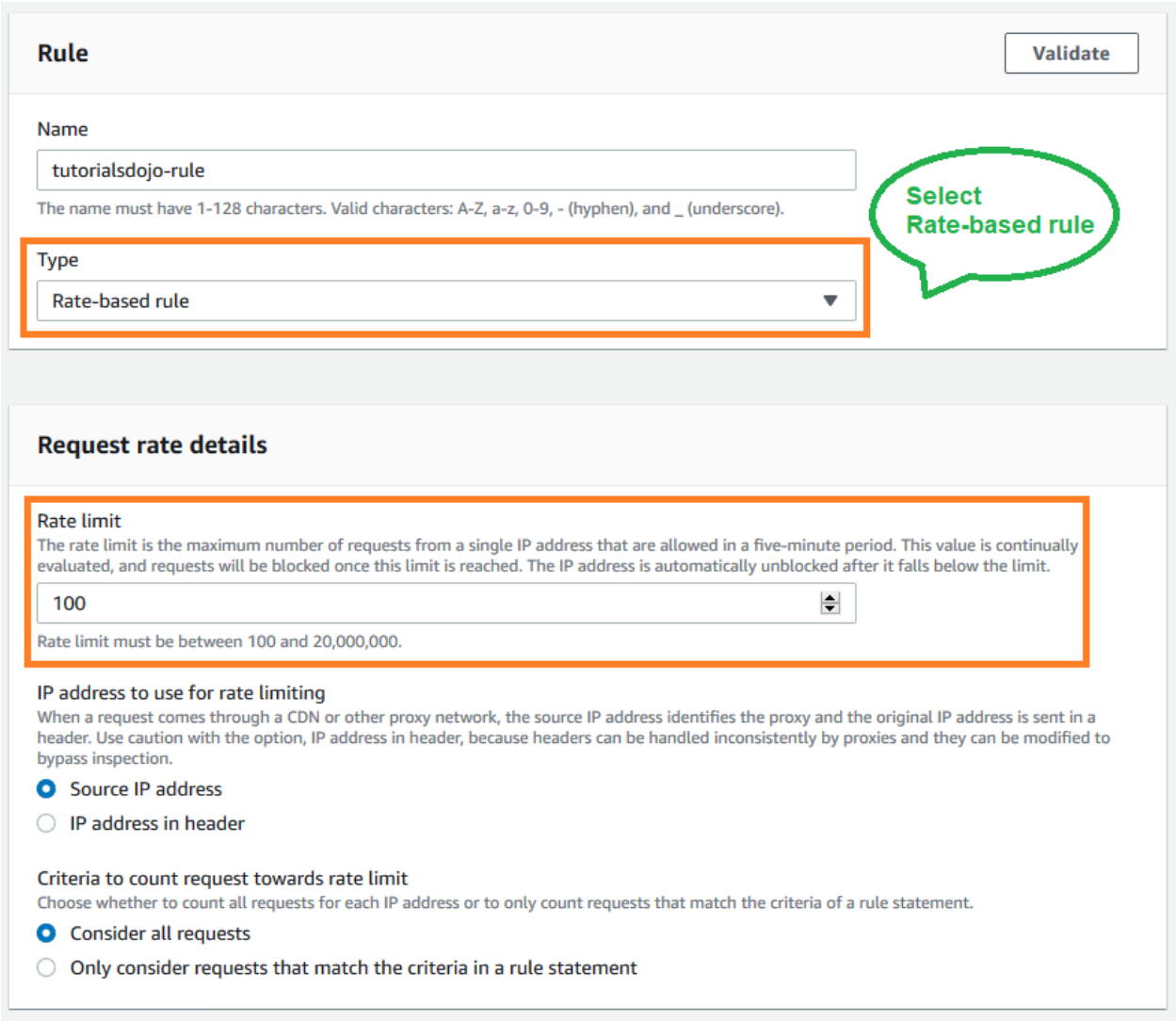
Category: CSAA – Design Secure Architectures

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 custom rule in the security group 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.** (Correct)
- **Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests.**
- **Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer.**

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.



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

45. QUESTION

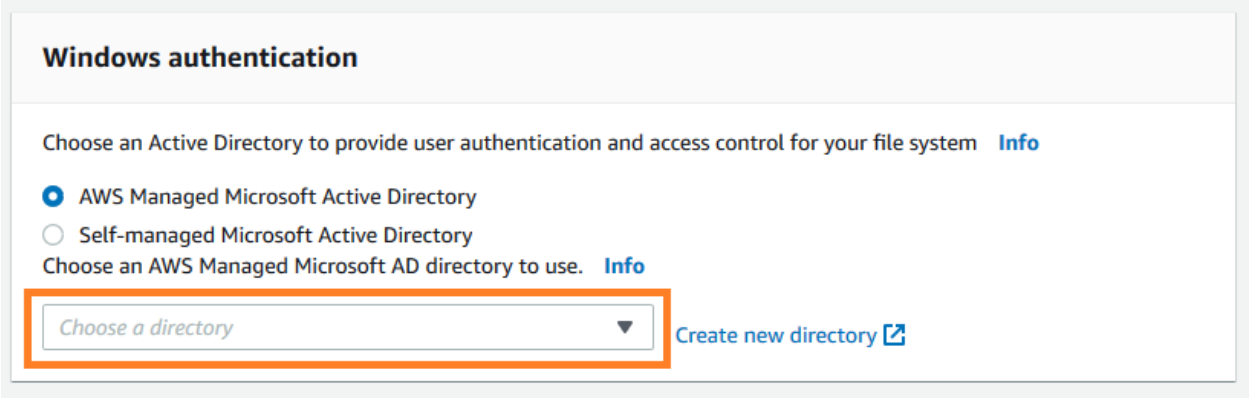
Category: CSAA – Design Resilient Architectures

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 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.**
- **Create a file system using Amazon FSx for Windows File Server and join it to an Active Directory domain in AWS.** (Correct)
- **Launch an Amazon EC2 Windows Server to mount a new S3 bucket as a file volume.**

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.



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/

46. QUESTION

Category: CSAA – Design Secure 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.

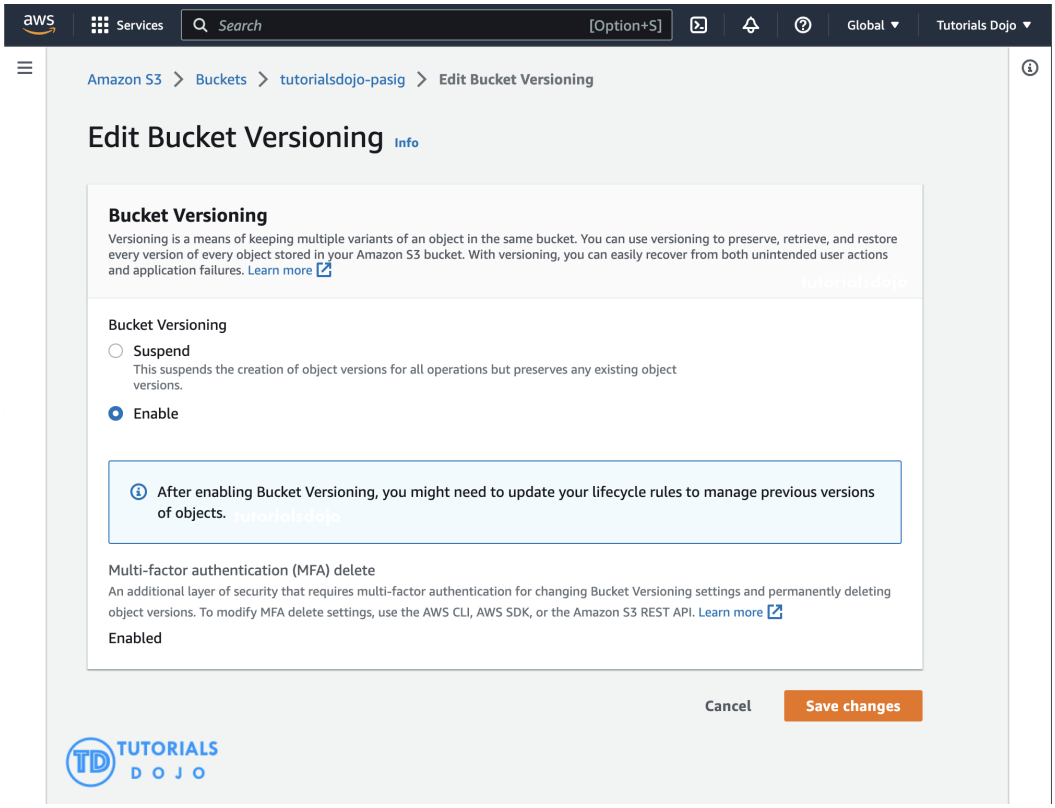
What can the DevOps engineers do to prevent this from happening again?

- **Create an IAM bucket policy that disables delete operation.**
- **Set up a signed URL for all users.**
- **Use S3 Infrequently Accessed storage to store the data.**
- **Enable S3 Versioning and Multi-Factor Authentication Delete on the bucket.** (Correct)

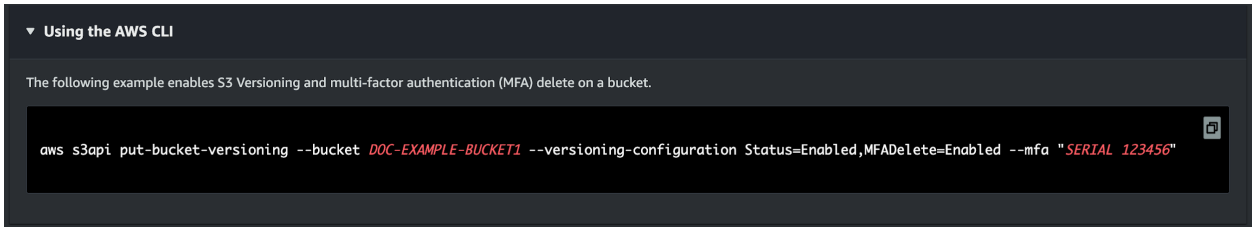
To avoid accidental deletion in Amazon S3 bucket, you can:

– 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/

47. QUESTION

Category: CSAA – Design Secure 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?

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

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/

48. QUESTION

Category: CSAA – Design High-Performing Architectures

A retail company receives raw `.csv` data files into its Amazon S3 bucket from various sources on an hourly basis. The average file size of these data files is 2 GB.

An automated process must be set up to convert these `.csv` files to a more efficient Apache Parquet format and store the output files in another S3 bucket. Additionally, the conversion process must be automatically triggered whenever a new file is uploaded into the S3 bucket.

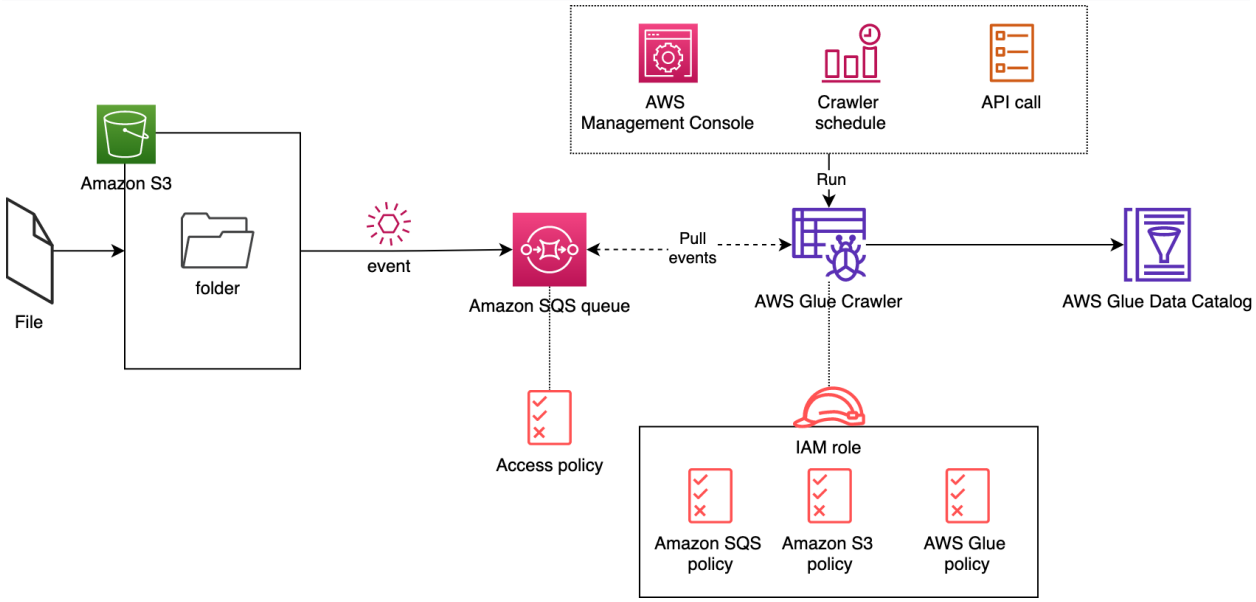
Which of the following options must be implemented to meet these requirements with the LEAST operational overhead?

- **Utilize an AWS Glue extract, transform, and load (ETL) job to process and convert the `.csv` files to Apache Parquet format and then store the output files into the target S3 bucket. Set up an S3 Event Notification to track every `S3 PUT` event and invoke the ETL job in AWS Glue through Amazon SQS.** (Correct)
- **Create an ETL (Extract, Transform, Load) job and a Data Catalog table in AWS Glue. Configure the AWS Glue crawler to run on a schedule to check for new files in the S3 bucket every hour and convert them to Parquet format.**
- **Set up an Apache Spark job running in an Amazon EC2 instance and create an Amazon EventBridge (Amazon CloudWatch Events) rule to**

monitor `S3 PUT` events in the S3 bucket. Configure AWS Lambda to invoke the Spark job for every new `.csv` file added via a Function URL.

Use a Lambda function triggered by an `S3 PUT` event to convert the `.csv` files to Parquet format. Use the AWS Transfer Family with SFTP service to move the output files to the target S3 bucket.

AWS Glue is a powerful ETL service that easily moves data between different data stores. By using AWS Glue, you can easily create and manage ETL jobs to transfer data from various sources, such as Amazon S3, Amazon RDS, and Amazon Redshift. Additionally, AWS Glue enables you to transform your data as needed to fit your specific needs. One of the key advantages of AWS Glue is its automatic schema discovery and mapping, which allows you to easily map data from different sources with different schemas.



When working with big data processing, it is often necessary to convert data from one format to another to optimize processing efficiency. Apache Parquet is a columnar storage format that is designed to provide higher efficiency and performance for big data processing. By storing and processing large amounts of data with high compression rates and faster query times, Parquet can offer significant benefits to the company. Fortunately, Parquet is compatible with many data processing frameworks such as Spark, Hive, and Hadoop, making it a versatile format for big data processing. By using AWS Glue and other AWS services, you can easily convert their .csv files to the more efficient Apache Parquet format and store

the output files in an S3 bucket, making it easy to access and process large amounts of data.

Hence the correct answer is: **Utilize an AWS Glue extract, transform, and load (ETL) job to process and convert the `.csv` files to Apache Parquet format and then store the output files into the target S3 bucket. Set up an S3 Event Notification to track every `S3 PUT` event and invoke the ETL job in AWS Glue through Amazon SQS.**

The option that says: **Use a Lambda function triggered by an `S3 PUT` event to convert the CSV files to Parquet format. Use the AWS Transfer Family with SFTP service to move the output files to the target S3 bucket** is incorrect. The conversion of the CSV files to Parquet format by using a combination of a Lambda function and S3 event notification would work; however, this is not the most efficient solution when handling large amounts of data. The Lambda function has a maximum execution time limit which means that converting large files may result in timeout issues. Using the AWS Transfer Family with SFTP service to move the output files to the target S3 bucket is unnecessary too. Moreover, reading the records has to be delivered via a data stream since a Lambda function has a memory limit. This entails additional effort compared with using AWS Glue.

The option that says: **Set up an Apache Spark job running in an Amazon EC2 instance and create an Amazon EventBridge (Amazon CloudWatch Events) rule to monitor `S3 PUT` events in the S3 bucket. Configure AWS Lambda to invoke the Spark job for every new `.csv` file added via a Function URL** is incorrect. Running Spark on EC2 instances requires manual provisioning, monitoring, and maintenance, leading to time and additional costs. Additionally, using Amazon EventBridge (Amazon CloudWatch Events) to trigger the Spark job through a Function URL adds complexity and potential points of failure. Thus, this option introduces unnecessary complexity and operational overhead.

The option that says: **Create an ETL (Extract, Transform, Load) job and a Data Catalog table in AWS Glue. Configure the AWS Glue crawler to run on a schedule to check for new files in the S3 bucket every hour and convert them to Parquet format** is incorrect. Although it is right to create an ETL job using AWS Glue, triggering the job on a scheduled basis rather than being triggered automatically by a new file upload is not ideal. It is not as efficient as using an S3 event trigger to initiate the conversion process immediately upon file upload.

References:

https://aws.amazon.com/blogs/big-data/run-aws-glue-crawlers-using-amazon-s3-event-notifications/

https://docs.aws.amazon.com/glue/latest/dg/aws-glue-programming-etl-format-parquet-home.html

https://docs.aws.amazon.com/athena/latest/ug/glue-athena.html

Check out this AWS Glue Cheat Sheet:

https://tutorialsdojo.com/aws-glue/

49. QUESTION

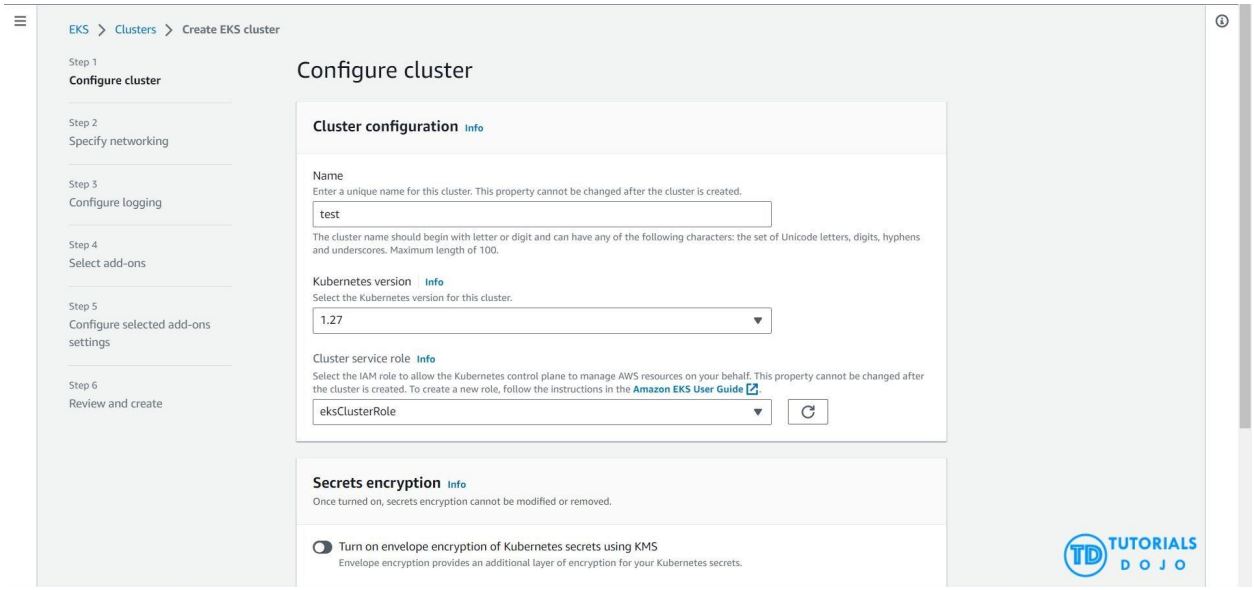
Category: CSAA – Design Resilient Architectures

A company has recently migrated its microservices-based application to Amazon Elastic Kubernetes Service (Amazon EKS). As part of the migration, the company must ensure that all sensitive configuration data and credentials, such as database passwords and API keys, are stored securely and encrypted within the Amazon EKS cluster's etcd key-value store.

What is the most suitable solution to meet the company's requirements?

- **Use Amazon EKS default options and the Amazon Elastic Block Store (Amazon EBS) Container Storage Interface (CSI) driver as an add-on to securely store sensitive data within the Amazon EKS cluster.**
- **Enable default Amazon EBS volume encryption for the account with a new AWS KMS key to ensure encryption of sensitive data within the Amazon EKS cluster.**
- **Use AWS Secrets Manager with a new AWS KMS key to securely manage and store sensitive data within the EKS cluster's etcd key-value store.** (Correct)
- **Enable secret encryption with a new AWS KMS key on an existing Amazon EKS cluster to encrypt sensitive data stored in the EKS cluster's etcd key-value store.**

Amazon Elastic Kubernetes Service (EKS) simplifies deploying, managing, and scaling containerized applications on Kubernetes clusters. AWS Secret Manager is a tool to securely store and collect sensitive information, such as database passwords and API keys.



By using AWS Secret Manager with a new AWS KMS key, you can add an extra layer of security to your EKS cluster's etcd key-value store. To do this, you need to create a new KMS key in the AWS KMS console, then create a new secret in the AWS Secret Manager console, specifying the new KMS key as the encryption key. Finally, you can configure your EKS cluster to use the new secret by creating a Kubernetes secret object that references the AWS Secret Manager secret.

This integration ensures that sensitive data is encrypted at rest and accessible only to authorized users or applications. By using AWS Secret Manager and a new AWS KMS key, you can ensure that your EKS cluster's etcd key-value store is secure and your sensitive data is protected.

Hence the correct answer is: **Use AWS Secrets Manager with a new AWS KMS key to securely manage and store sensitive data within the EKS cluster's etcd key-value store.**

The option that says: **Enable secret encryption with a new AWS KMS key on an existing Amazon EKS cluster to encrypt sensitive data stored in the EKS cluster's etcd key-value store** is incorrect. While enabling secret encryption with a new AWS KMS key on an existing Amazon EKS cluster would add encryption for secrets at rest, it doesn't specifically address the requirement of storing sensitive configuration data and credentials securely within the Amazon EKS cluster's etcd key-value store. Encryption of secrets at rest is essential, but it doesn't guarantee that the sensitive data stored in etcd is appropriately encrypted and managed.

The option that says: **Enable default Amazon EBS volume encryption for the account with a new AWS KMS key to ensure encryption of sensitive data within the Amazon EKS cluster** is incorrect. Enabling default Amazon EBS volume encryption is a way to ensure that data at rest in EBS volumes is encrypted. However, the EBS volumes are primarily used for persistent storage of the worker nodes. They are not directly related to the storage of sensitive configuration data and credentials within the EKS cluster's etcd key-value store.

The option that says: **Use Amazon EKS default options and the Amazon Elastic Block Store (Amazon EBS) Container Storage Interface (CSI) driver as an add-on to store sensitive data within the Amazon EKS cluster securely** is incorrect. Amazon EBS CSI driver enables Amazon Elastic Block Store (EBS) volumes as persistent storage for Kubernetes applications running on the Amazon EKS. While this can provide secure persistent storage for your microservices, it does not address the specific requirement of securely storing sensitive data within the EKS cluster's etcd key-value store.

References:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_csi_driver.html

https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html

https://docs.aws.amazon.com/eks/latest/userguide/security.html

Check out these Amazon Elastic Kubernetes Service and AWS Secrets Manager Cheat Sheets:

https://tutorialsdojo.com/amazon-elastic-kubernetes-service-eks/

https://tutorialsdojo.com/aws-secrets-manager/

50. QUESTION

Category: CSAA – Design Secure Architectures

A company has a web application that uses Amazon CloudFront to distribute its images, videos, and other static contents stored in its S3 bucket to its users around the world. The company has recently introduced a new member-only access feature 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?

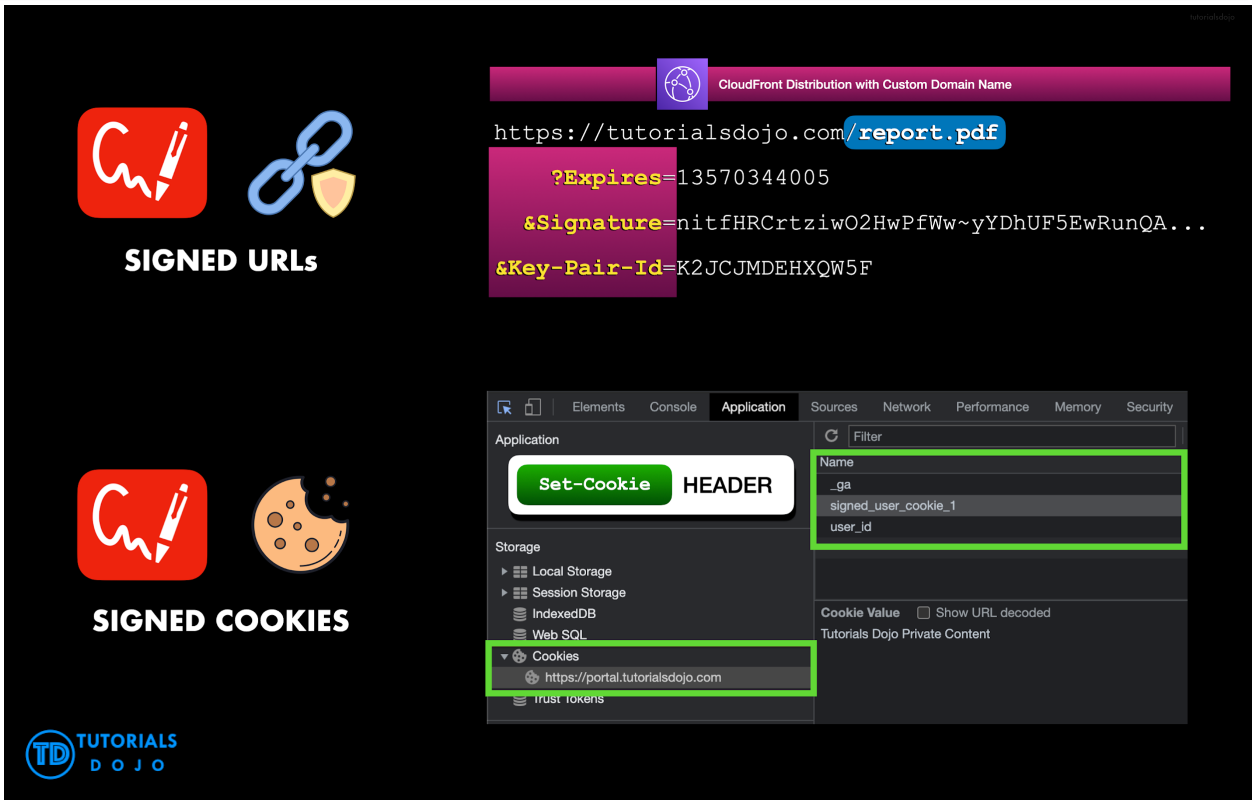
- Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member.
- Create a Signed URL with a custom policy which 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)

Many companies that distribute content over the internet want to restrict access to documents, business data, media streams, or content that is intended for selected users, for example, users who have paid a fee. To securely serve this private content by using CloudFront, you can do the following:

– Require that your users access your private content by using special CloudFront signed URLs or signed cookies.

– Require that your users access your content by using CloudFront URLs, not URLs that access content directly on the origin server (for example, Amazon S3 or a private HTTP server). Requiring CloudFront URLs isn't necessary, but we recommend it to prevent users from bypassing the restrictions that you specify in signed URLs or signed cookies.

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 that 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 in 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 URLs.

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.

References:

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/

51. QUESTION

Category: CSAA – Design Secure 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?

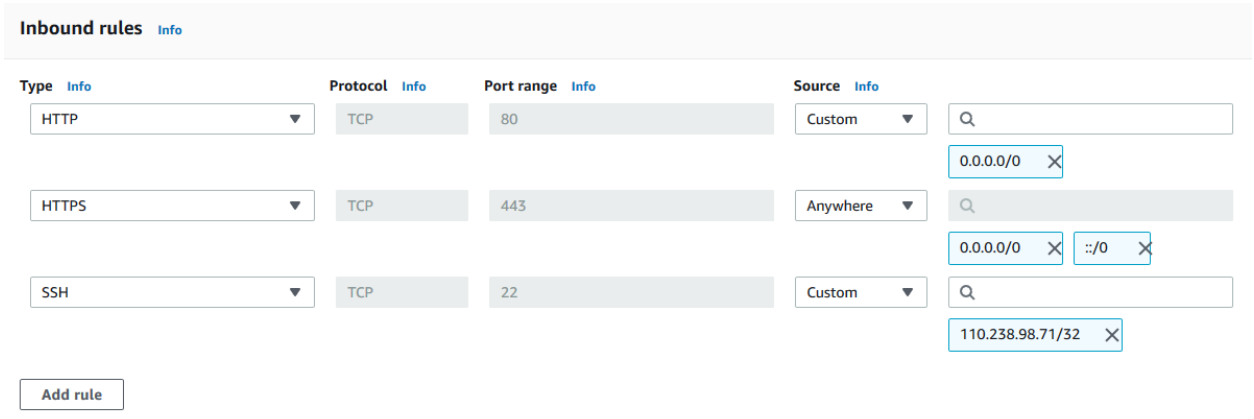
Security Group Inbound Rule: Protocol – TCP, Port Range – 22, Source 110.238.98.71/32 (Correct)

Security Group Inbound Rule: Protocol – UDP, Port Range – 22, Source 110.238.98.71/32

Security Group Outbound Rule: Protocol – TCP, Port Range – 22, Destination 110.238.98.71/32

Security Group Outbound Rule: Protocol – UDP, Port Range – 22, Destination 0.0.0.0/0

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.



The requirement is to only allow the individual IP of the client and not the entire network. The /32 CIDR notation denotes a single IP address. Take note that the SSH

protocol uses TCP, not UDP, and runs on port 22 (default). In the scenario, we can create a security group with an inbound rule allowing incoming traffic from the specified IP address on port 22.

Security groups are stateful, meaning they automatically allow return traffic associated with the client who initiated the connection to the instance. Therefore, any return traffic from the specified IP address on port 22 will be allowed to pass through the security group, regardless of whether or not there is an explicit outbound rule allowing it.

Hence, the correct answer is: **Security Group Inbound Rule: Protocol – TCP, Port Range – 22, Source 110.238.98.71/32**

Security Group Inbound Rule: Protocol – UDP, Port Range – 22, Source 110.238.98.71/32 is incorrect because it uses UDP instead of TCP. SSH runs over the TCP protocol, so specifying UDP would not allow the desired access.

Security Group Outbound Rule: Protocol – TCP, Port Range – 22, Destination 110.238.98.71/32 is incorrect because it's an outbound rule, not an inbound rule. Outbound rules control traffic leaving the instance. In the scenario, we need to limit inbound traffic coming from a specific address.

Security Group Outbound Rule: Protocol – UDP, Port Range – 22, Destination 0.0.0.0/0 is incorrect because it is an outbound rule rather than an inbound rule. Moreover, SSH connections require TCP.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html#security-group-rules

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

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

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

53. 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. (Correct)

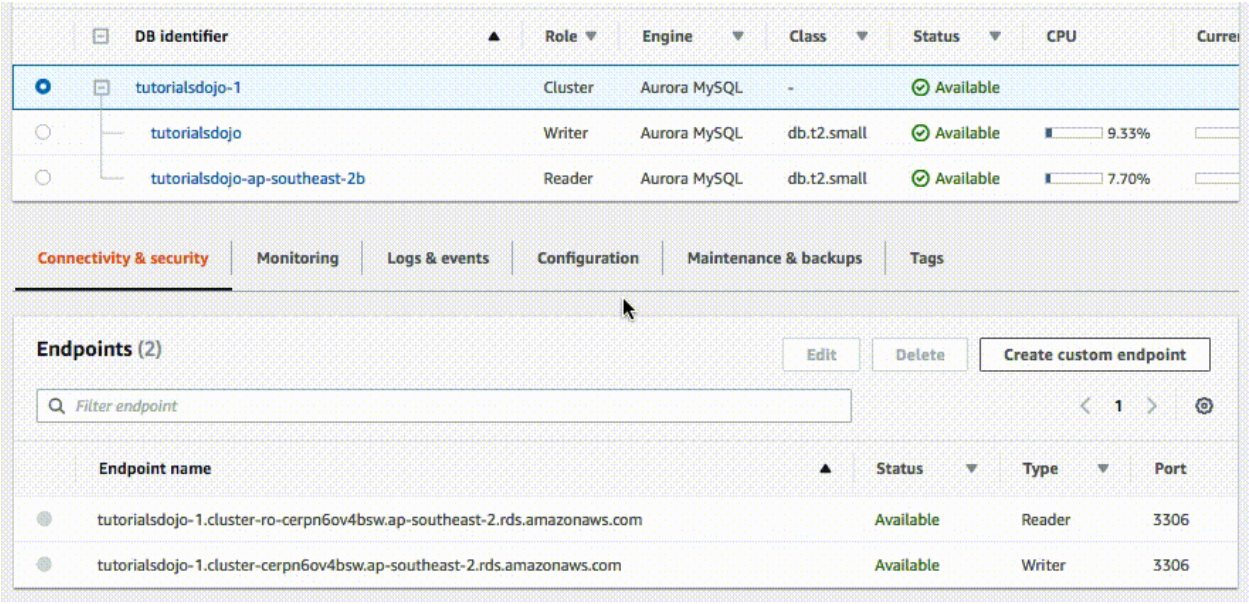
Configure your application to use the reader endpoint for both production traffic and reporting queries, which will enable your Aurora database to automatically perform load-balancing among all the Aurora Replicas.

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.

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.

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.



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. 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

Amazon Aurora Overview: https://youtu.be/iwS1h7rLNBQ

54. QUESTION

Category: CSAA – Design Secure 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 data containing personally identifiable information (PII) in a manner that meets their compliance standards. They should also be alerted if there are potential policy violations with the privacy of their S3 buckets.

Which of the following should the Architect implement to satisfy this requirement?

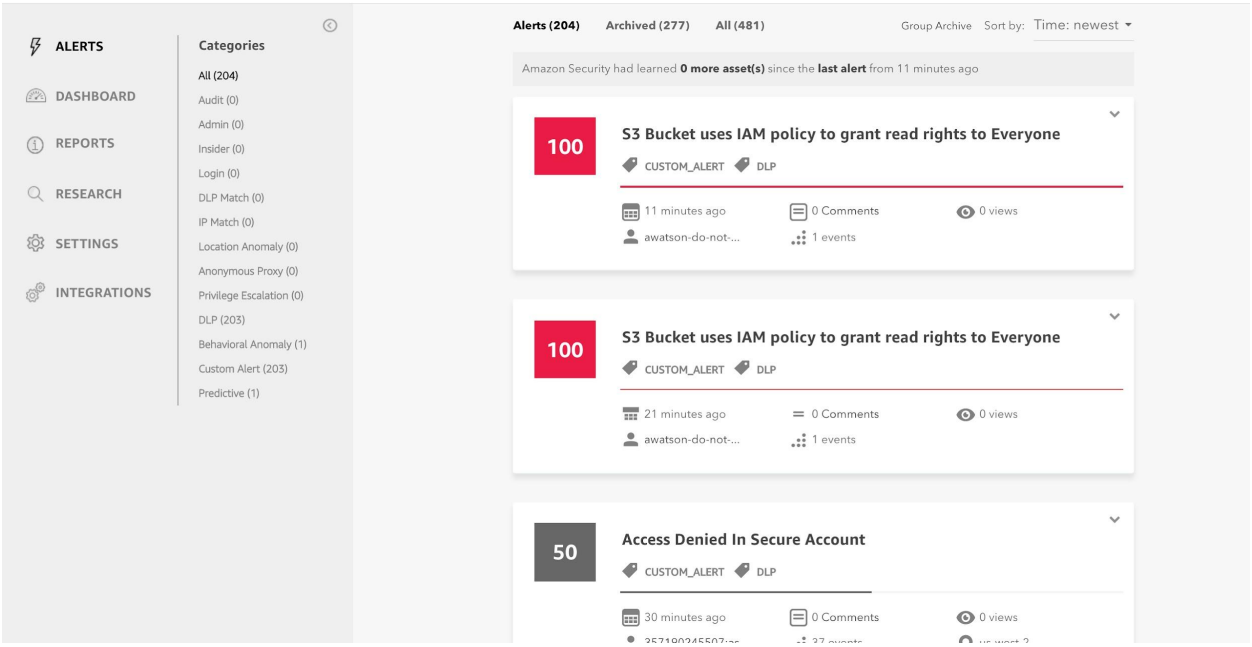
Set up and configure Amazon Macie to monitor their Amazon S3 data. (Correct)

Set up and configure Amazon Kendra to monitor malicious activity on their Amazon S3 data

Set up and configure Amazon Polly to scan for usage patterns on Amazon S3 data

Set up and configure Amazon Fraud Detector to send out alert notifications whenever a security violation is detected on their Amazon S3 data.

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 generates two categories of findings: policy findings and sensitive data findings. A policy finding is a detailed report of a potential policy violation or issue with the security or privacy of an Amazon S3 bucket. Macie generates these

findings as part of its ongoing monitoring activities for your Amazon S3 data. A sensitive data finding is a detailed report of sensitive data in an S3 object. Macie generates these findings when it discovers sensitive data in S3 objects that you configure a sensitive data discovery job to analyze.

Hence, the correct answer is: **Set up and configure Amazon Macie to monitor their Amazon S3 data.**

The option that says: **Set up and configure Amazon Polly to scan for usage patterns on Amazon S3 data** is incorrect because Amazon Polly is simply a service that turns text into lifelike speech, allowing you to create applications that talk, and build entirely new categories of speech-enabled products. Polly can't be used to scane usage patterns on your S3 data.

The option that says: **Set up and configure Amazon Kendra to monitor malicious activity on their Amazon S3 data** is incorrect Amazon Kendra is just an enterprise search service that allows developers to add search capabilities to their applications. This enables their end users to discover information stored within the vast amount of content spread across their company, but not monitor malcious activity on their S3 buckets.

The option that says: **Set up and configure Amazon Fraud Detector to send out alert notifications whenever a security violation is detected on their Amazon S3 data** is incorrect because the Amazon Fraud Detector is only a fully managed service for identifying potentially fraudulent activities and for catching more online fraud faster. It does not check any S3 data containing personally identifiable information (PII), unlike Amazon Macie.

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:
https://youtu.be/ogVamzF2Dzk

55. 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 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.**
- **The web application is not deleting the messages in the SQS queue after it has processed them.** (Correct)
- **The web application is set for long polling so the messages are being sent twice.**

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.

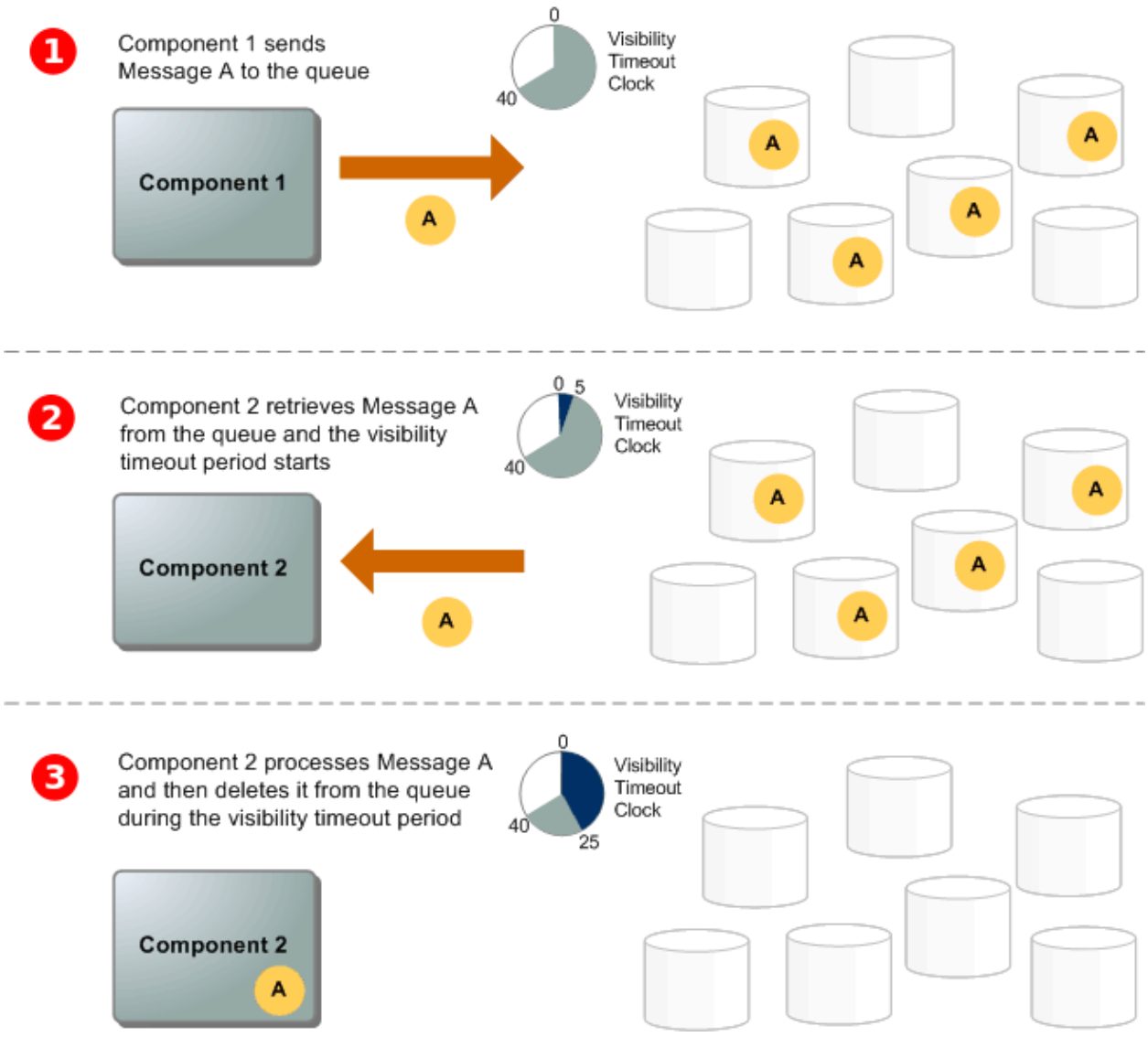
There are three main parts in a distributed messaging system:

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/

56. 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. (Correct)

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.

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. (Correct)

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.

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.

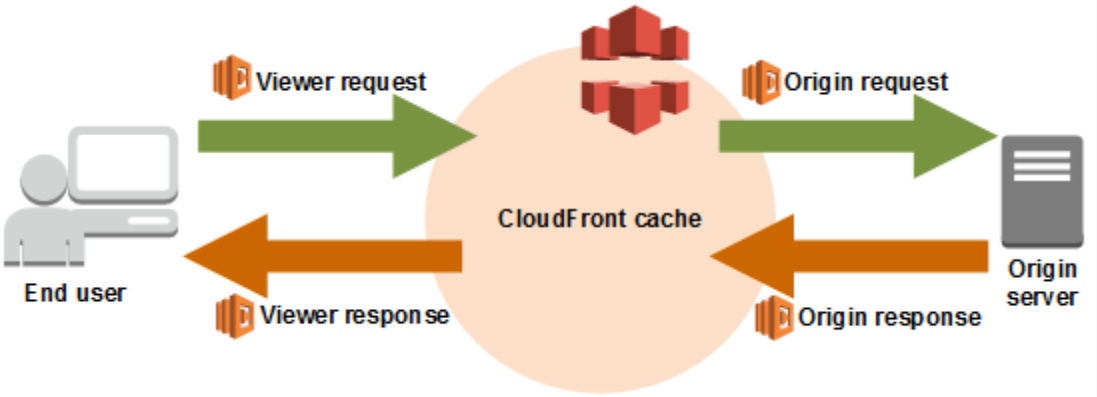
Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests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

57. 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 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.

Use partition keys with high-cardinality attributes, which have a large number of distinct values for each item. (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/3ZOyUNIeorU

58. 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 an Auto Scaling group of Amazon EC2 instances to handle the bursts of traffic in seconds.

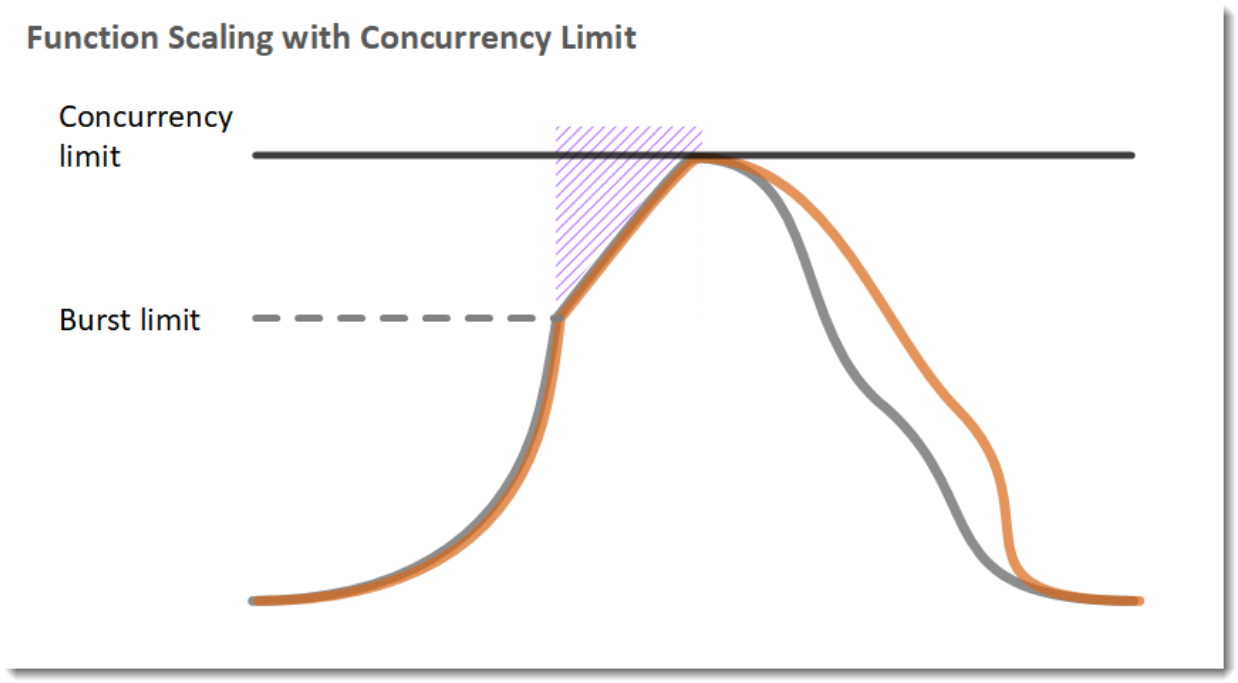
Create an API using Amazon API Gateway and use the Amazon ECS cluster with Service Auto Scaling to handle the bursts of traffic in seconds.

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 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.



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/

59. 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?

Switch from using AWS Lambda and API Gateway to a more scalable and highly available architecture using EC2 instances, ELB, and Auto Scaling.
Enable throttling limits and result caching in API Gateway.
Use CloudFront in front of the API Gateway to act as a cache.
Move the Lambda function in a VPC.

59. 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.

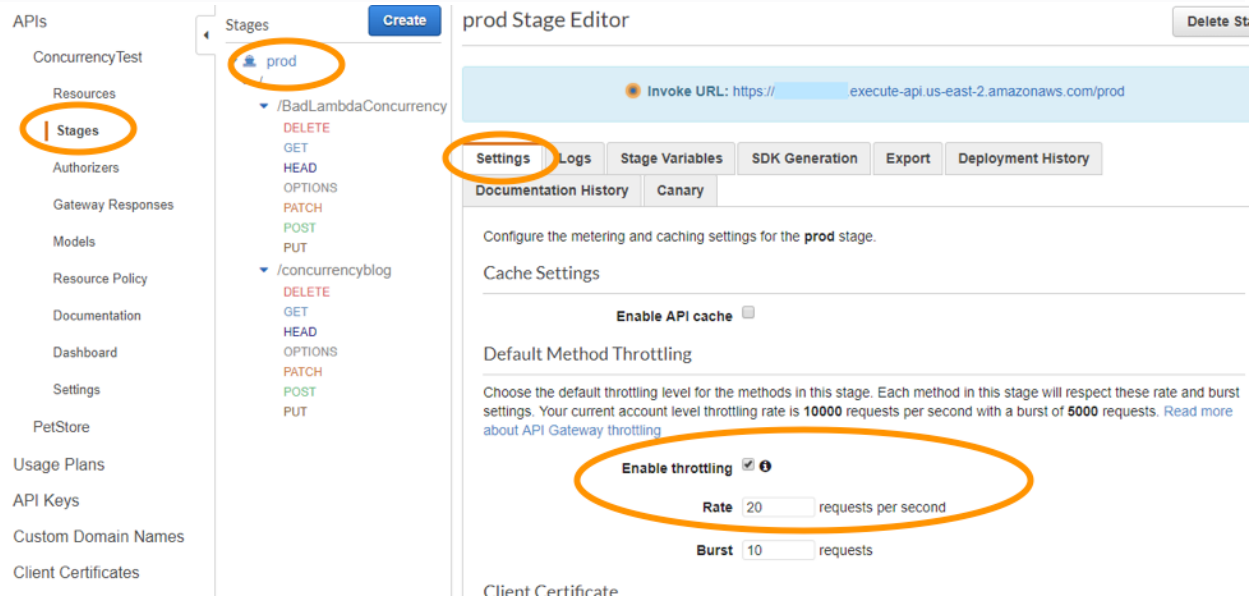
In this scenario, how can you protect the backend systems of the platform from traffic spikes?

Switch from using AWS Lambda and API Gateway to a more scalable and highly available architecture using EC2 instances, ELB, and Auto Scaling.
Enable throttling limits and result caching in API Gateway. (Correct)
Use CloudFront in front of the API Gateway to act as a cache.
Move the Lambda function in a VPC.

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 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

60. QUESTION

Category: CSAA – Design Secure Architectures

A government agency plans to store confidential tax documents on AWS. Due to the sensitive information in the files, the Solutions Architect must restrict the data access requests made to the storage solution to a specific Amazon VPC only. The solution should also prevent the files from being deleted or overwritten to meet the regulatory requirement of having a write-once-read-many (WORM) storage model.

Which combination of the following options should the Architect implement? (Select TWO.)

Set up a new Amazon S3 bucket to store the tax documents and integrate it with AWS Network Firewall. Configure the Network Firewall to only accept data access requests from a specific Amazon VPC.

Store the tax documents in the Amazon S3 Glacier Instant Retrieval storage class to restrict fast data retrieval to a particular Amazon VPC of your choice.

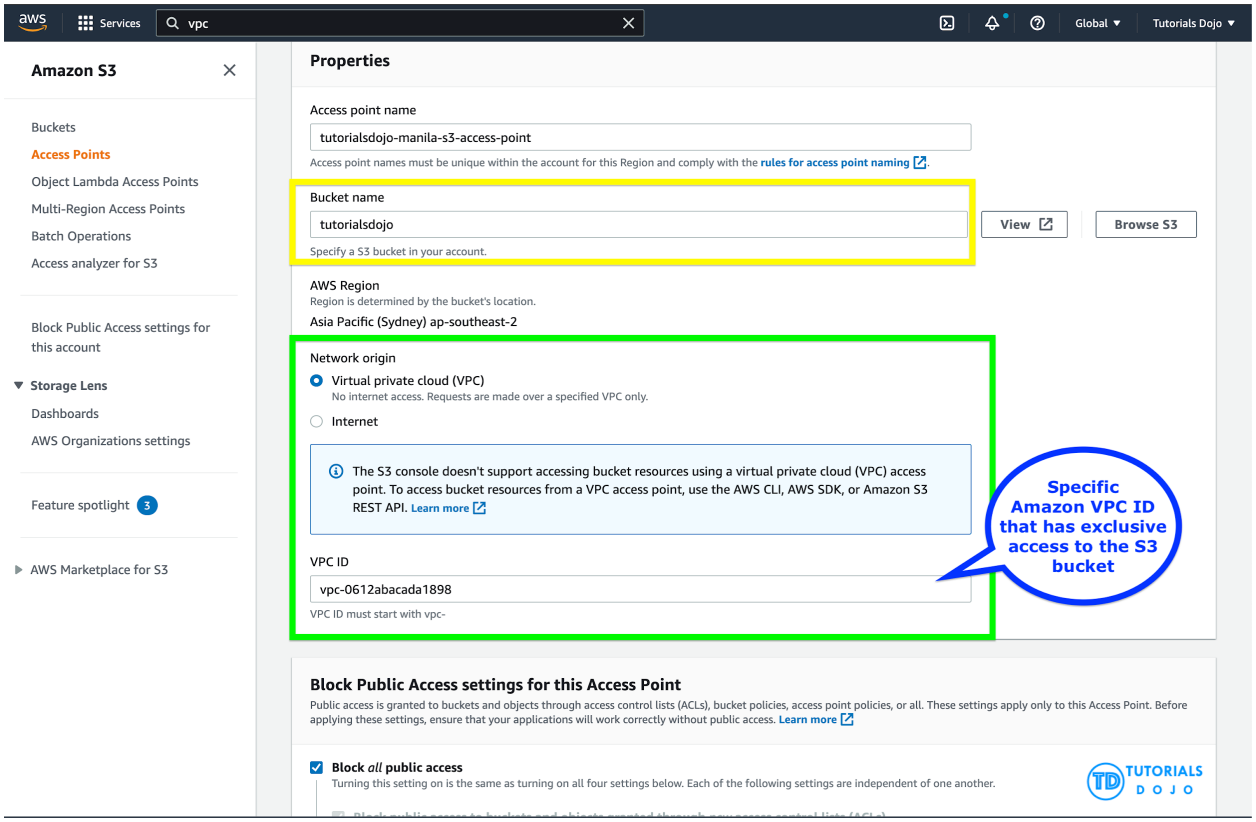
Configure an Amazon S3 Access Point for the S3 bucket to restrict data access to a particular Amazon VPC only. (Correct)

Enable Object Lock but disable Object Versioning on the new Amazon S3 bucket to comply with the write-once-read-many (WORM) storage model requirement.

Create a new Amazon S3 bucket with the S3 Object Lock feature enabled. Store the documents in the bucket and set the Legal Hold option for object retention. (Correct)

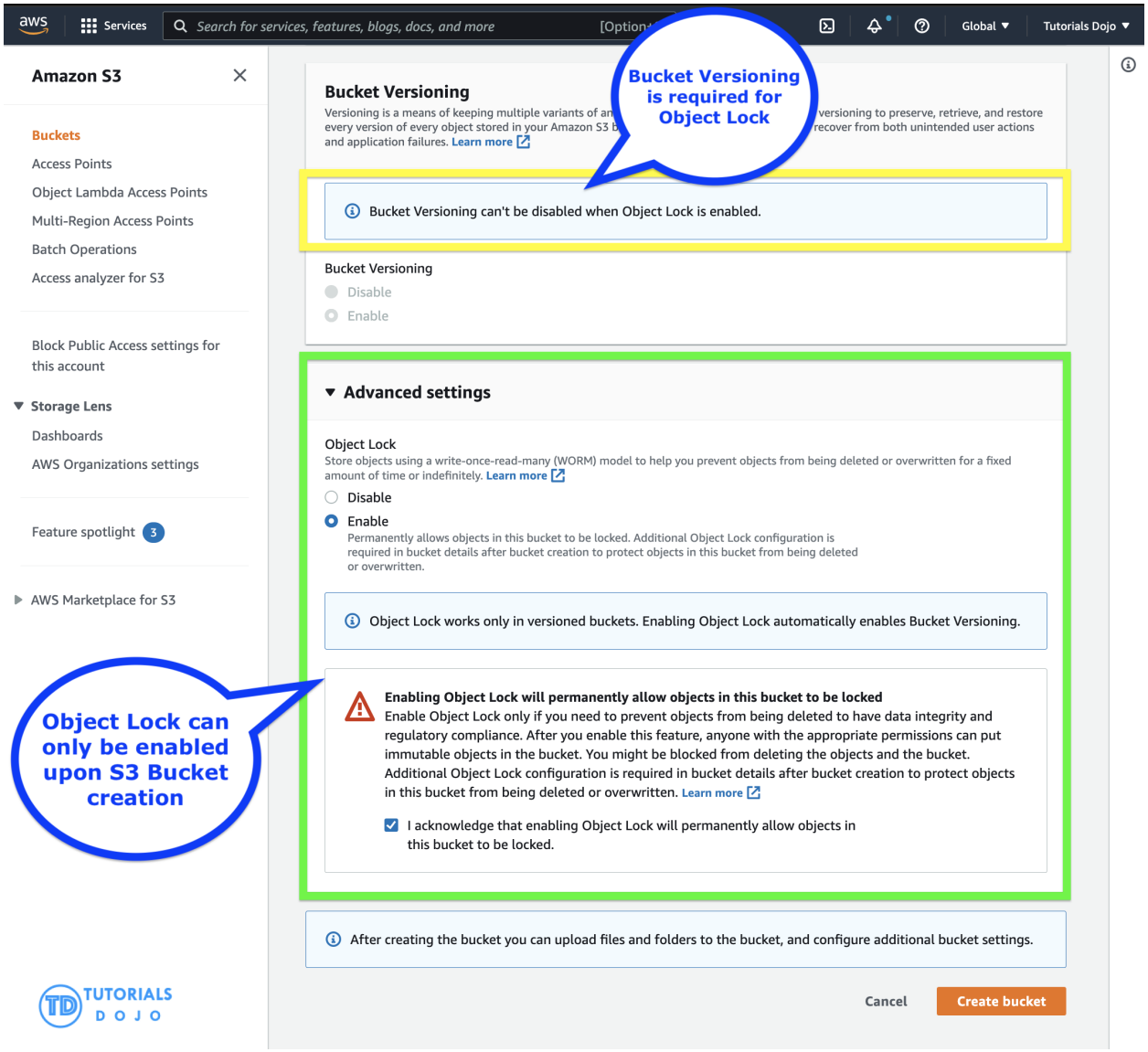
Amazon S3 access points simplify data access for any AWS service or customer application that stores data in S3. Access points are named network endpoints that are attached to buckets that you can use to perform S3 object operations, such as `GetObject` and `PutObject`.

Each access point has distinct permissions and network controls that S3 applies for any request that is made through that access point. Each access point enforces a customized access point policy that works in conjunction with the bucket policy that is attached to the underlying bucket. You can configure any access point to accept requests only from a virtual private cloud (VPC) to restrict Amazon S3 data access to a private network. You can also configure custom block public access settings for each access point.



You can also use Amazon S3 Multi-Region Access Points to provide a global endpoint that applications can use to fulfill requests from S3 buckets located in multiple AWS Regions. You can use Multi-Region Access Points to build multi-Region applications with the same simple architecture used in a single Region, and then run those applications anywhere in the world. Instead of sending requests over the congested public internet, Multi-Region Access Points provide built-in network resilience with acceleration of internet-based requests to Amazon S3. Application requests made to a Multi-Region Access Point global endpoint use AWS Global Accelerator to automatically route over the AWS global network to the S3 bucket with the lowest network latency.

With S3 Object Lock, you can store objects using a write-once-read-many (WORM) model. Object Lock can help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. You can use Object Lock to help meet regulatory requirements that require WORM storage, or to simply add another layer of protection against object changes and deletion.



Before you lock any objects, you have to enable a bucket to use S3 Object Lock. You enable Object Lock when you create a bucket. After you enable Object Lock on a bucket, you can lock objects in that bucket. When you create a bucket with Object Lock enabled, you can't disable Object Lock or suspend versioning for that bucket.

Hence, the correct answers are:

– Configure an Amazon S3 Access Point for the S3 bucket to restrict data access to a particular Amazon VPC only.

– Create a new Amazon S3 bucket with the S3 Object Lock feature enabled. Store the documents in the bucket and set the Legal Hold option for object retention.

The option that says: **Set up a new Amazon S3 bucket to store the tax documents and integrate it with AWS Network Firewall. Configure the Network Firewall to only**

accept data access requests from a specific Amazon VPC is incorrect because you cannot directly use an AWS Network Firewall to restrict S3 bucket data access requests to a specific Amazon VPC only. You have to use an Amazon S3 Access Point instead for this particular use case. An AWS Network Firewall is commonly integrated to your Amazon VPC and not to an S3 bucket.

The option that says: **Store the tax documents in the Amazon S3 Glacier Instant Retrieval storage class to restrict fast data retrieval to a particular Amazon VPC of your choice** is incorrect because Amazon S3 Glacier Instant Retrieval is just an archive storage class that delivers the lowest-cost storage for long-lived data that is rarely accessed and requires retrieval in milliseconds. It neither provides write-once-read-many (WORM) storage nor a fine-grained network control that restricts S3 bucket access to a specific Amazon VPC.

The option that says: **Enable Object Lock but disable Object Versioning on the new Amazon S3 bucket to comply with the write-once-read-many (WORM) storage model requirement** is incorrect. Although the Object Lock feature does provide write-once-read-many (WORM) storage, the Object Versioning feature must also be enabled too in order for this to work. In fact, you cannot manually disable the Object Versioning feature if you have already selected the Object Lock option.

References:

https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-points.html

https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lock.html

Check out this Amazon S3 Cheat Sheet:

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

61. QUESTION

Category: CSAA – Design Secure 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 the storage of their personal documents.

Which of the following will you need to consider so you can set up a solution that incorporates a 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.** (Correct)
- **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.**
- **Set up a Federation proxy or an Identity provider, and use AWS Security Token Service to generate temporary tokens.** (Correct)

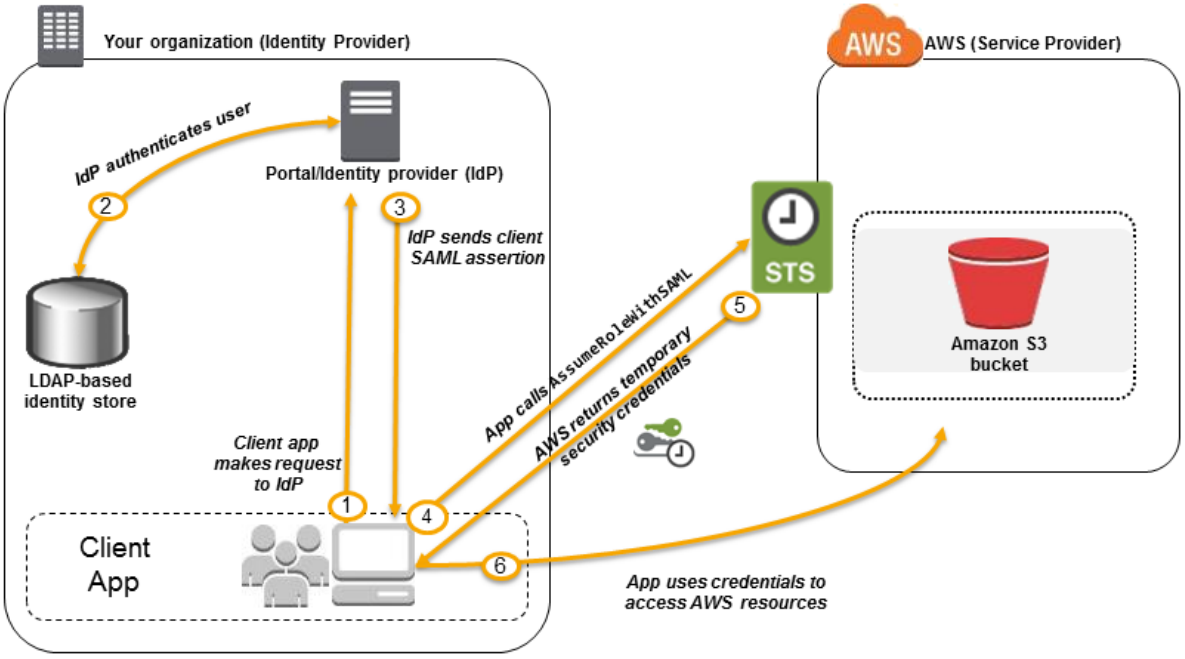
The question refers to one of the common scenarios for temporary credentials in AWS. Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles. In this example, it is called enterprise identity federation considering that you also 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/premiumsupport/knowledge-center/iam-s3-user-specific-folder/

62. QUESTION

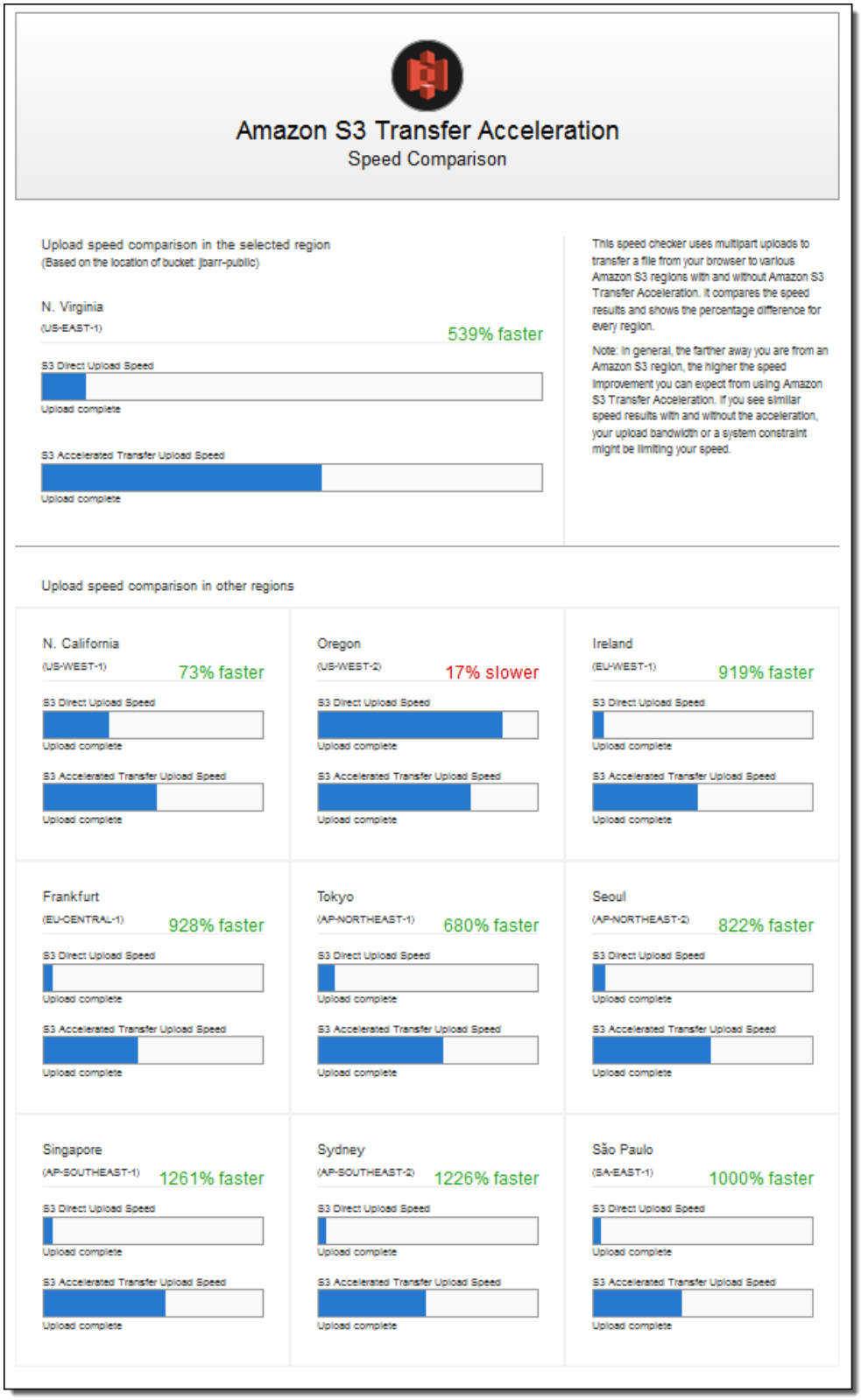
Category: CSAA – Design High-Performing Architectures

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.** (Correct)
- **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.**
- **Set up a Site-to-Site VPN connection.**

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.



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/

63. QUESTION

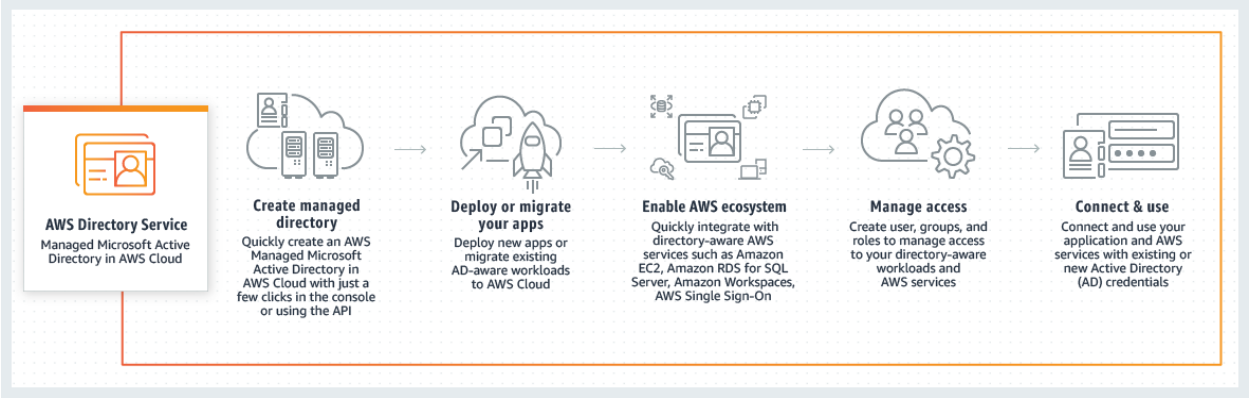
Category: CSAA – Design Resilient Architectures

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** (Correct)
- **Lambda**
- **AWS Directory Service Simple AD**
- **IAM Groups**
- **IAM Roles** (Correct)

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-ad-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

64. QUESTION

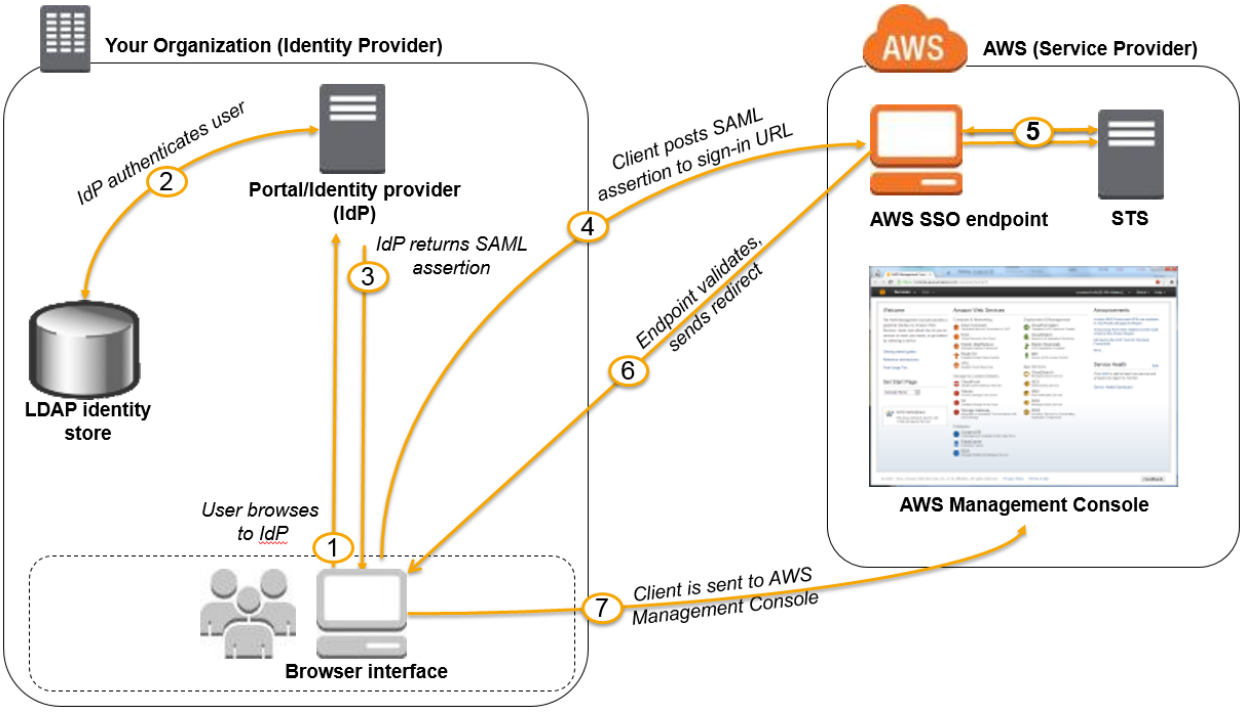
Category: CSAA – Design Secure Architectures

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 Microsoft Active Directory Federation Service (AD FS).** (Correct)
- **Use IAM users**
- **Set up SAML 2.0-Based Federation by using a Web Identity Federation.**
- **Use Amazon VPC**

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/

65. QUESTION

Category: CSAA – Design Resilient Architectures

An e-commerce company uses a regional Amazon API Gateway to host its public REST APIs. The API Gateway endpoint is accessed through a custom domain name configured using an Amazon Route 53 alias record. As part of its continuous improvement efforts, the company wants to release a new version of its APIs which includes enhanced features and performance optimizations.

How can the company minimize customer impact, and ensure MINIMAL data loss during the update process in the MOST cost-effective manner?

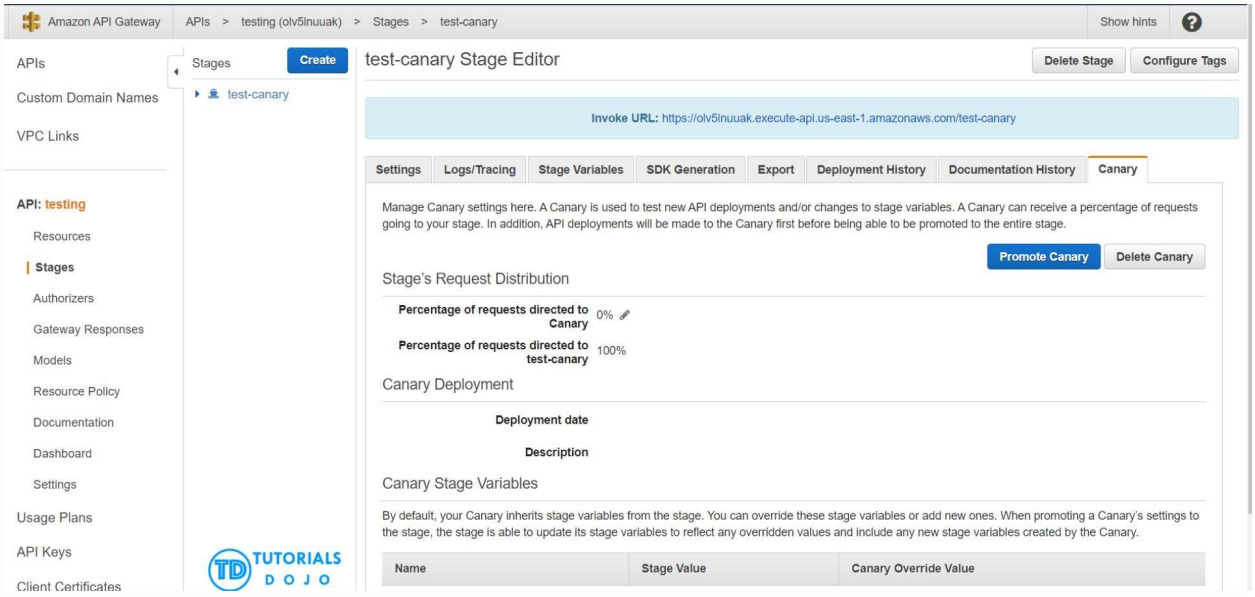
Implement a canary release deployment strategy for the API Gateway. Deploy the latest version of the APIs to a canary stage and direct a portion of the user traffic to this stage. Verify the new APIs. Gradually increase the traffic percentage, monitor for any issues, and, if successful, promote the canary stage to production. (Correct)

Create a new API Gateway with the updated version of the APIs in OpenAPI JSON or YAML file format, but keep the same custom domain name for the new API Gateway.

Implement a blue-green deployment strategy for the API Gateway. Deploy the latest version of the APIs to the green environment and direct some of the user traffic to it. Verify the new APIs. If it is thoroughly verified, deploy the green environment to production.

Modify the existing API Gateway with the updated version of the APIs, but keep the same custom domain name for the new API Gateway by using the import-to-update operation in either overwrite or merge mode.

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. It is a front door for your APIs, enabling you to design and implement scalable, highly available, and secure APIs. With Amazon API Gateway, you can create RESTful APIs that any HTTP client, such as web browsers and mobile devices, can consume.



Implementing a canary release deployment strategy for the API Gateway is a great way to ensure your APIs remain stable and reliable. This strategy involves releasing a new version

of your API to a small subset of users, allowing you to test the latest version in a controlled environment.

If the new version performs well, you can gradually roll out the update to the rest of your users. This approach lets you catch any issues before they affect your entire user base, minimizing the impact on your customers. By using Amazon API Gateway, you can quickly implement a canary release deployment strategy, ensuring that your APIs are always up-to-date and performing at their best.

Hence the correct answer is: **Implement a canary release deployment strategy for the API Gateway. Deploy the latest version of the APIs to a canary stage and direct a portion of the user traffic to this stage. Verify the new APIs. Gradually increase the traffic percentage, monitor for any issues, and, if successful, promote the canary stage to production.**

The option that says: **Create a new API Gateway with the updated version of the APIs in OpenAPI JSON or YAML file format, but keep the same custom domain name for the new API Gateway** is incorrect. Upgrading to a new API Gateway using an updated version of the APIs in OpenAPI JSON or YAML file format while keeping the same custom domain name can result in downtime and confusion during the switch. This is because of DNS propagation delays, which can negatively affect users and even lead to data loss.

The option that says: **Modify the existing API Gateway with the updated version of the APIs, but keep the same custom domain name for the new API Gateway by using the import-to-update operation in either overwrite or merge mode** is incorrect. Using the import-to-update operation in either overwrite or merge mode may not provide enough isolation and control testing for the new version of the APIs. If something goes wrong during the update process, it could lead to data loss on the existing API Gateway, potentially affecting all customers simultaneously.

The option that says: **Implement a blue-green deployment strategy for the API Gateway. Deploy the latest version of the APIs to the green environment and direct some of the user traffic to it. Verify the new APIs. If it is thoroughly verified, deploy the green environment to production** is incorrect. In a blue-green deployment, the blue (existing) and green (updated) environments must be provisioned and maintained. This adds complexity and cost to the update process, which breaks the cost requirement that's explicitly mentioned in the scenario. Additionally, directing some user traffic to the green environment may lead to issues for those users, especially if there are undiscovered bugs or performance problems in the updated APIs.

References:

https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html

https://docs.aws.amazon.com/apigateway/latest/developerguide/canary-release.html

https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-import-api-update.html

Check out this Amazon API Gateway Cheat Sheet:

https://tutorialsdojo.com/amazon-api-gateway/