TOPIC BASED - AWS SA PRACTICE QUESTIONS

Video-Based Practise Questions: https://www.youtube.com/watch?v=Wee_F8_k8s4

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

Topic-Based – DynamoDB (SA-Associate)

1. QUESTION

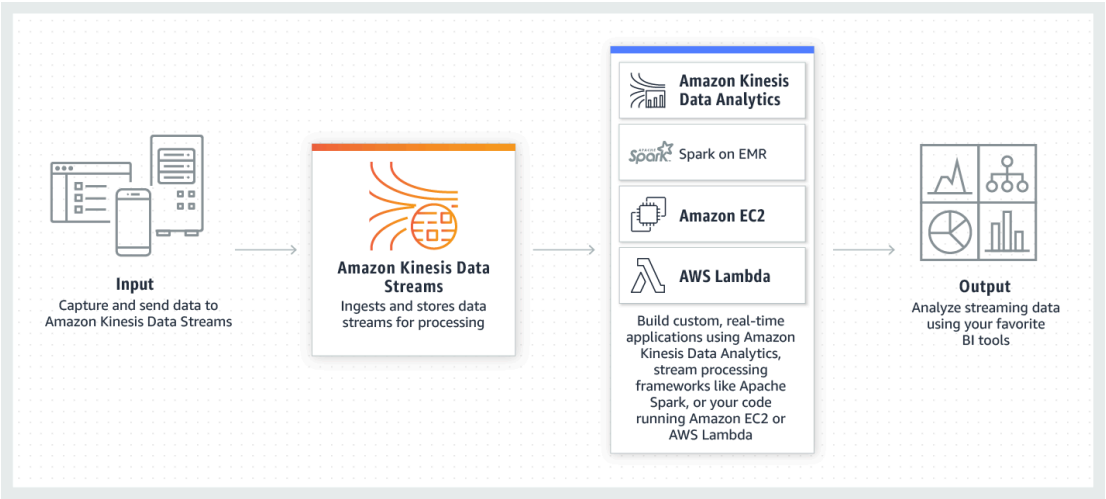
Category: CSAA – Design High-Performing Architectures

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?

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

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/

2. QUESTION

Category: CSAA – Design Secure Architectures

A GraphQL API hosted is hosted in an Amazon EKS cluster with Fargate launch type and deployed using AWS SAM. The API is connected to an Amazon DynamoDB table with an Amazon DynamoDB Accelerator (DAX) as its data store. Both resources are hosted in the us-east-1 region.

The AWS IAM authenticator for Kubernetes is integrated into the EKS cluster for role-based access control (RBAC) and cluster authentication. A solutions architect must improve network security by preventing database calls from traversing the public internet. An automated cross-account backup for the DynamoDB table is also required for long-term retention.

Which of the following should the solutions architect implement to meet the requirement?

Create a DynamoDB interface endpoint. Associate the endpoint to the appropriate route table. Enable Point-in-Time Recovery (PITR) to restore the DynamoDB table to a particular point in time on the same or a different AWS account.
Create a DynamoDB gateway endpoint. Set up a Network Access Control List (NACL) rule that allows outbound traffic to the `dynamodb.us-east-1.amazonaws.com` gateway endpoint. Use the

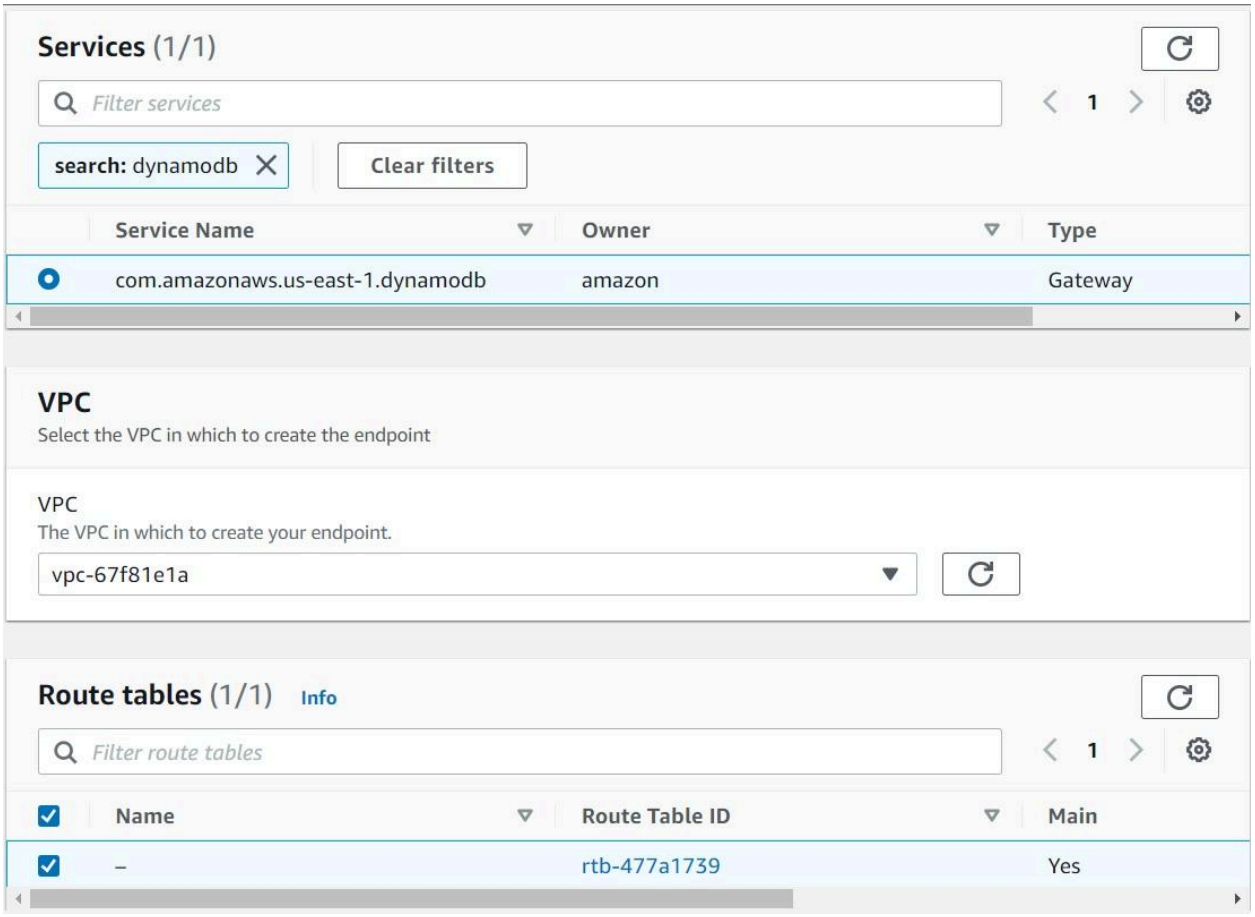
built-in on-demand DynamoDB backups for cross-account backup and recovery.

Create a DynamoDB gateway endpoint. Associate the endpoint to the appropriate route table. Use AWS Backup to automatically copy the on-demand DynamoDB backups to another AWS account for disaster recovery. (Correct)

Create a DynamoDB interface endpoint. Set up a stateless rule using AWS Network Firewall to control all outbound traffic to only use the `dynamodb.us-east-1.amazonaws.com` endpoint. Integrate the DynamoDB table with Amazon Timestream to allow point-in-time recovery from a different AWS account.

Since DynamoDB tables are public resources, applications within a VPC rely on an Internet Gateway to route traffic to/from Amazon DynamoDB. You can use a Gateway endpoint if you want to keep the traffic between your VPC and Amazon DynamoDB within the Amazon network. This way, resources residing in your VPC can use their private IP addresses to access DynamoDB with no exposure to the public internet.

When you create a DynamoDB Gateway endpoint, you specify the VPC where it will be deployed as well as the route table that will be associated with the endpoint. The route table will be updated with an Amazon DynamoDB prefix list (list of CIDR blocks) as the destination and the endpoint's ID as the target.



DynamoDB on-demand backups are available at no additional cost beyond the normal pricing that's associated with backup storage size. DynamoDB on-demand backups cannot be copied to a different account or Region. To create backup copies across AWS accounts and Regions and for other advanced features, you should use AWS Backup.

With AWS Backup, you can configure backup policies and monitor activity for your AWS resources and on-premises workloads in one place. Using DynamoDB with AWS Backup, you can copy your on-demand backups across AWS accounts and Regions, add cost allocation tags to on-demand backups, and transition on-demand backups to cold storage for lower costs. To use these advanced features, you must opt into AWS Backup. Opt-in choices apply to the specific account and AWS Region, so you might have to opt into multiple Regions using the same account.

Hence, the correct answer is: **Create a DynamoDB gateway endpoint. Associate the endpoint to the appropriate route table. Use AWS Backup to automatically copy the on-demand DynamoDB backups to another AWS account for disaster recovery.**

The option that says: **Create a DynamoDB interface endpoint. Associate the endpoint to the appropriate route table. Enable Point-in-Time Recovery (PITR) to restore the**

DynamoDB table to a particular point in time on the same or a different AWS account is incorrect because Amazon DynamoDB does not support interface endpoint. You have to create a DynamoDB Gateway endpoint instead. In addition, the Point-in-Time Recovery (PITR) feature is not capable of restoring a DynamoDB table to a particular point in time in a different AWS account. If this functionality is needed, you have to use the AWS Backup service instead.

The option that says: **Create a DynamoDB gateway endpoint. Set up a Network Access Control List (NACL) rule that allows outbound traffic to the `dynamodb.us-east-1.amazonaws.com` gateway endpoint. Use the built-in on-demand DynamoDB backups for cross-account backup and recovery** is incorrect because using a Network Access Control List alone is not enough to prevent traffic traversing to the public Internet. Moreover, you cannot copy DynamoDB on-demand backups to a different account or Region.

The option that says: **Create a DynamoDB interface endpoint. Set up a stateless rule using AWS Network Firewall to control all outbound traffic to only use the `dynamodb.us-east-1.amazonaws.com` endpoint. Integrate the DynamoDB table with Amazon Timestream to allow point-in-time recovery from a different AWS account** is incorrect. Keep in mind that the `dynamodb.us-east-1.amazonaws.com` is a public service endpoint for Amazon DynamoDB. Since the application is able to communicate with Amazon DynamoDB prior to the required architectural change, it's implied that no firewalls (security group, NACL, etc.) are blocking traffic to/from Amazon DynamoDB, hence, adding an NACL rule to allow outbound traffic to DynamoDB is unnecessary. Furthermore, the use of the AWS Network Firewall in this solution is simply incorrect as you have to integrate this with your Amazon VPC. The use of Amazon Timestream is also wrong since this is a time series database service in AWS for IoT and operational applications. You cannot directly integrate DynamoDB and Amazon Timestream for the purpose of point-in-time data recovery.

References:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/vpc-endpoints-dynamodb.html

https://aws.amazon.com/blogs/database/how-to-configure-a-private-network-environment-for-amazon-dynamodb-using-vpc-endpoints/

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html

Check out this Amazon DynamoDB Cheat sheet:

https://tutorialsdojo.com/amazon-dynamodb

3. QUESTION

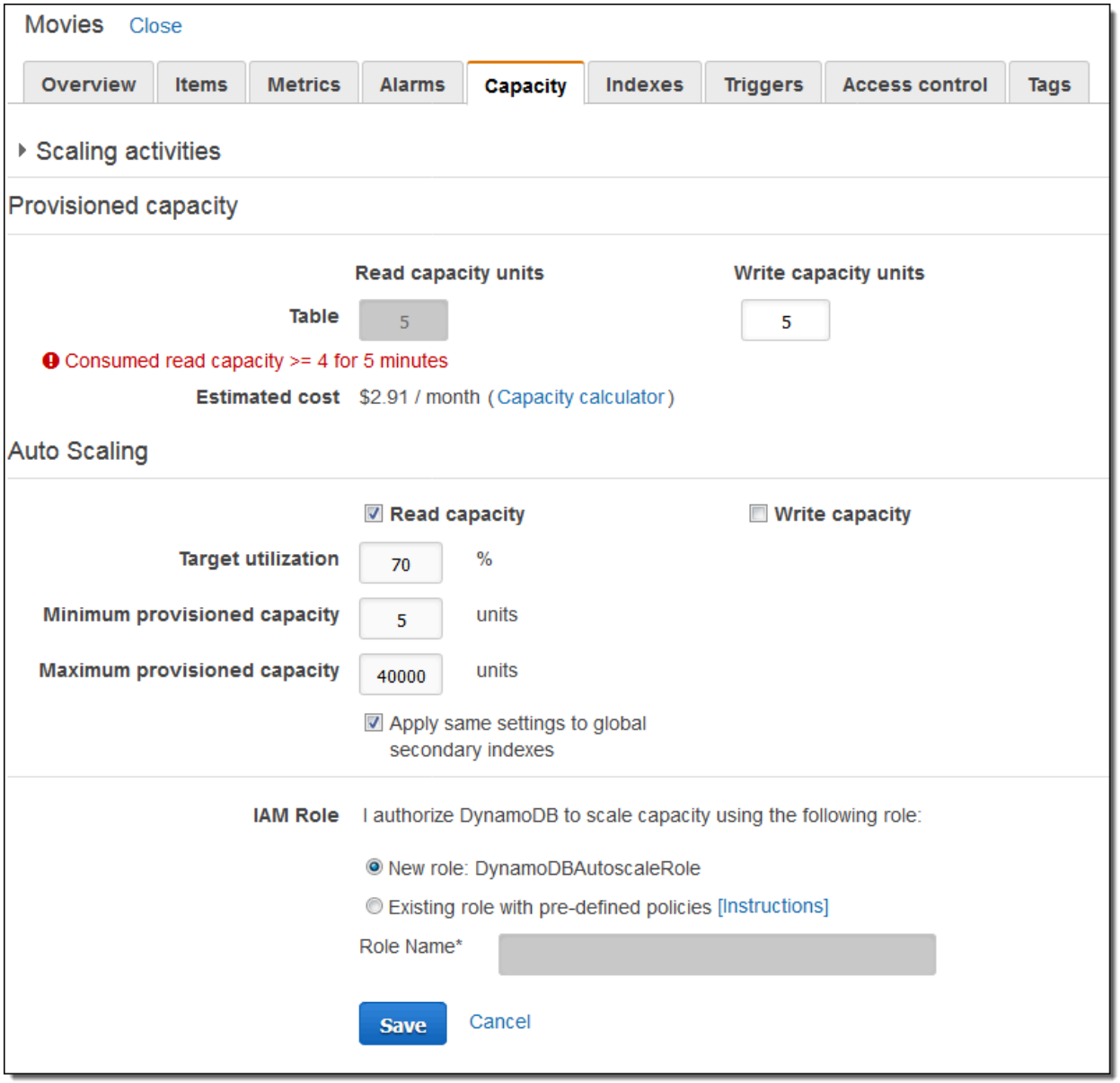
Category: CSAA – Design High-Performing Architectures

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

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

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

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



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

AWS Lambda scales your functions automatically on your behalf. Every time an event notification is received for your function, AWS Lambda quickly locates free capacity within its compute fleet and runs your code. Since your code is stateless,

AWS Lambda can start as many copies of your function as needed without lengthy deployment and configuration delays.

The correct answers are the options that say:

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

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

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

The option that says: **Use AWS IAM Identity Center to authenticate users and have them directly access DynamoDB using single sign-on. Manually set the provisioned read and write capacity to a higher RCU and WCU** is incorrect because AWS IAM Identity Center is a service that just makes it easy to centrally manage access to multiple AWS accounts and business applications. This will not be of much help to the scalability and performance of the application. It is costly to manually set the provisioned read and write capacity to a higher RCU and WCU because this capacity will run round the clock and will still be the same even if the incoming traffic is stable and there is no need to scale.

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

References:

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

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

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

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

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A popular augmented reality (AR) mobile game is heavily using a RESTful API which is hosted in AWS. The API uses Amazon API Gateway and a DynamoDB table with a preconfigured read and write capacity. Based on your systems monitoring, the DynamoDB table begins to throttle requests during high peak loads which causes the slow performance of the game.

Which of the following can you do to improve the performance of your app?

- Create an SQS queue in front of the DynamoDB table.
- Use DynamoDB Auto Scaling (Correct)
- Add the DynamoDB table to an Auto Scaling Group.
- Integrate an Application Load Balancer with your DynamoDB table.

DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without throttling. When the workload decreases, Application Auto Scaling decreases the throughput so that you don't pay for unused provisioned capacity.

Using DynamoDB Auto Scaling is the best answer. DynamoDB Auto Scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf.

Integrating an Application Load Balancer with your DynamoDB table is incorrect because an Application Load Balancer is not suitable to be used with DynamoDB and in addition, this will not increase the throughput of your DynamoDB table.

Adding the DynamoDB table to an Auto Scaling Group is incorrect because you usually put EC2 instances on an Auto Scaling Group, and not a DynamoDB table.

Creating an SQS queue in front of the DynamoDB table is incorrect because this is not a design principle for high throughput DynamoDB table. Using SQS is for

handling queuing and polling the request. This will not increase the throughput of DynamoDB which is required in this situation.

Reference:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html

Check out this Amazon DynamoDB Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Resilient Architectures

A company is running a web application on AWS. The application is made up of an Auto-Scaling group that sits behind an Application Load Balancer and an Amazon DynamoDB table where user data is stored. The solutions architect must design the application to remain available in the event of a regional failure. A solution to automatically monitor the status of your workloads across your AWS account, conduct architectural reviews and check for AWS best practices.

Which configuration meets the requirement with the least amount of downtime possible?

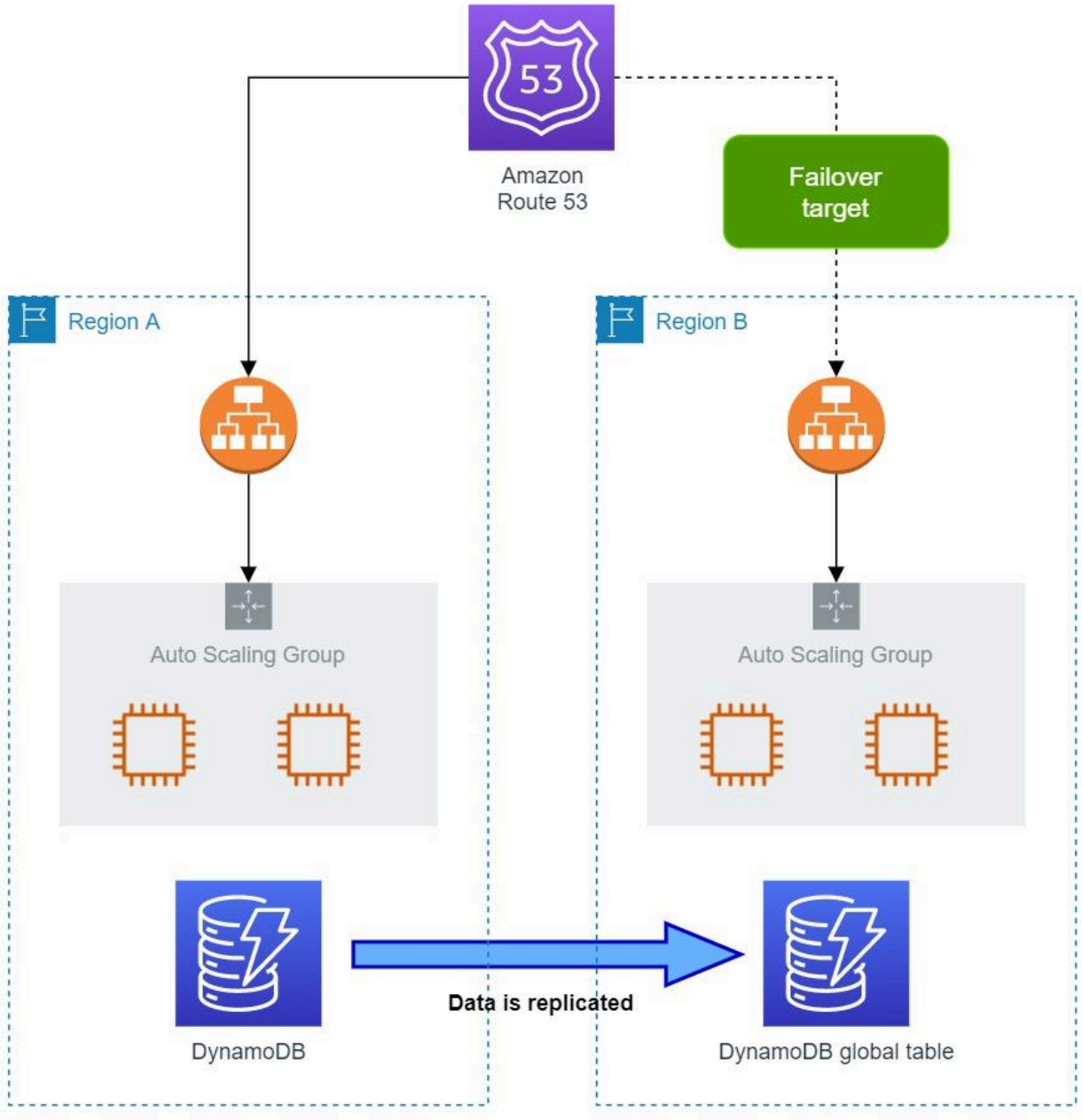
- In a secondary region, create a global table of the DynamoDB table and replicate the auto-scaling group and application load balancer. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up the AWS Well-Architected Tool to easily get recommendations for improving your workloads based on the AWS best practices (Correct)
- In a secondary region, create a global secondary index of the DynamoDB table and replicate the auto-scaling group and application load balancer. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up the AWS Compute

Optimizer to automatically get recommendations for improving your workloads based on the AWS best practices

Write a CloudFormation template that includes the auto-scaling group, application load balancer, and DynamoDB table. In the event of a failure, deploy the template in a secondary region. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up and configure the Amazon Managed Service for Prometheus service to receive insights for improving your workloads based on the AWS best practices.

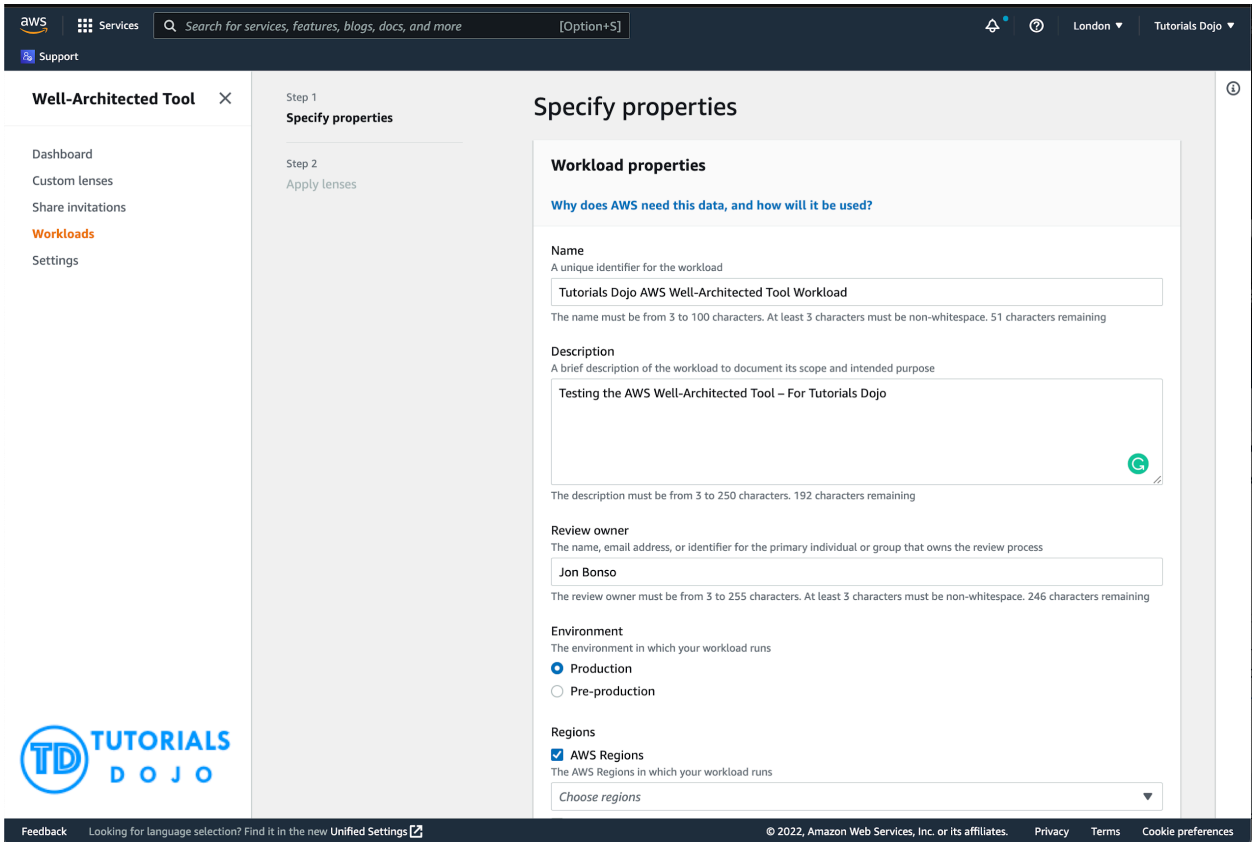
Write a CloudFormation template that includes the auto-scaling group, application load balancer, and DynamoDB table. In the event of a failure, deploy the template in a secondary region. Configure Amazon EventBridge (Amazon CloudWatch Events) to trigger a Lambda function that updates the application's Route 53 DNS record. Launch an Amazon Managed Grafana workspace to automatically receive tips and action items for improving your workloads based on the AWS best practices

When you have more than one resource performing the same function—for example, more than one HTTP serve—you can configure Amazon Route 53 to check the health of your resources and respond to DNS queries using only the healthy resources. For example, suppose your website, example.com, is hosted on six servers, two each in three data centers around the world. You can configure Route 53 to check the health of those servers and to respond to DNS queries for example.com using only the servers that are currently healthy.



In this scenario, you can replicate the process layer (EC2 instances, Application Load Balancer) to a different region and create a global table based on the existing DynamoDB table (data layer). Amazon DynamoDB will handle data synchronization between the tables in different regions. This way, the state of the application is preserved even in the event of an outage. Lastly, configure Route 53 DNS failover and set the DNS name of the backup application load balancer as a target.

You can also use the Well-Architected Tool to automatically monitor the status of your workloads across your AWS account, conduct architectural reviews and check for AWS best practices.



This tool is based on the AWS Well-Architected Framework, which was developed to help cloud architects build secure, high-performing, resilient, and efficient application infrastructures. The Framework has been used in tens of thousands of workload reviews by AWS solutions architects, and it provides a consistent approach for evaluating your cloud architecture and implementing designs that will scale with your application needs over time.

Hence, the correct answer is: **In a secondary region, create a global table of the DynamoDB table and replicate the auto-scaling group and application load balancer. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up the AWS Well-Architected Tool to easily get recommendations for improving your workloads based on the AWS best practices**

The option that says: **In a secondary region, create a global secondary index of the DynamoDB table and replicate the auto-scaling group and application load balancer. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up the AWS Compute Optimizer to automatically get recommendations for improving your workloads based on the AWS best practices** is incorrect because this configuration is impossible to implement. A global secondary index can only be created in the region where its parent table resides. Moreover, the AWS Compute Optimizer simply helps you to identify the optimal AWS resource configurations, such as Amazon Elastic Compute Cloud (EC2) instance types,

Amazon Elastic Block Store (EBS) volume configurations, and AWS Lambda function memory sizes. It is not capable of providing recommendations to improve your workloads based on AWS best practices.

The option that says: **Write a CloudFormation template that includes the auto-scaling group, application load balancer, and DynamoDB table. In the event of a failure, deploy the template in a secondary region. Use Route 53 DNS failover to automatically route traffic to the resources in the secondary region. Set up and configure the Amazon Managed Service for Prometheus service to receive insights for improving your workloads based on the AWS best practices** is incorrect. This solution describes a situation in which the environment is provisioned only after a regional failure occurs. It won't work because to enable Route 53 DNS failover, you'd need to target an existing environment. The use of the Amazon Managed Service for Prometheus service is irrelevant as well. This is just a serverless, Prometheus-compatible monitoring service for container metrics that makes it easier to securely monitor container environments at scale.

The option that says: **Write a CloudFormation template that includes the auto-scaling group, application load balancer, and DynamoDB table. In the event of a failure, deploy the template in a secondary region. Configure Amazon EventBridge (Amazon CloudWatch Events) to trigger a Lambda function that updates the application's Route 53 DNS record. Launch an Amazon Managed Grafana workspace to automatically receive tips and action items for improving your workloads based on the AWS best practices** is incorrect. This could work, but it won't deliver the shortest downtime possible since resource provisioning takes minutes to complete. Switching traffic to a standby environment is a faster method, albeit more expensive. Amazon Managed Grafana is a fully managed service with rich, interactive data visualizations to help customers analyze, monitor, and alarm on metrics, logs, and traces across multiple data sources. This service does not provide recommendations based on AWS best practices. You have to use the AWS Well-Architected Tool instead.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-configuring.html

https://aws.amazon.com/blogs/networking-and-content-delivery/creating-disaster-recovery-mechanisms-using-amazon-route-53/

https://aws.amazon.com/well-architected-tool

Check out this Amazon Route 53 Cheat Sheet:

https://tutorialsdojo.com/amazon-route-53/

6. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

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/

7. 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)

Set up a DAX cluster to access the source DynamoDB table. Create a new DynamoDB trigger and a Lambda function. For every update made in the user data, the trigger will send data to the Lambda function which will then notify the subscribers via email using SNS.

Create a Lambda function that uses DynamoDB Streams Kinesis Adapter which will fetch data from the DynamoDB Streams endpoint. Set up an SNS Topic that will notify the subscribers via email when there is an update made by a particular user.

Using the Kinesis Client Library (KCL), write an application that leverages on DynamoDB Streams Kinesis Adapter that will fetch data from the DynamoDB Streams endpoint. When there are updates made by a particular user, notify the subscribers via email using SNS.

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.

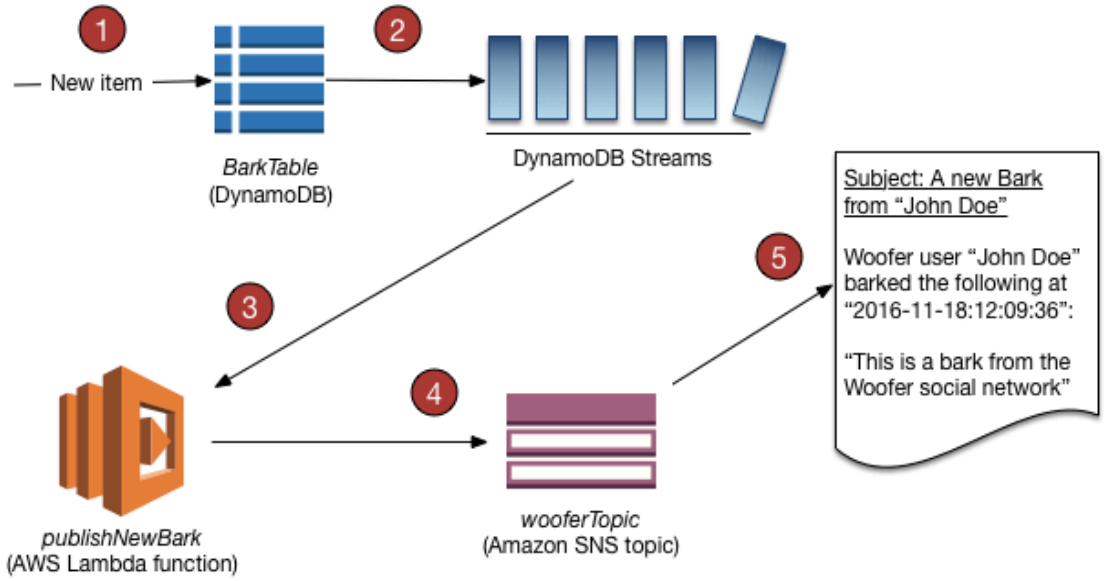
Whenever an application creates, updates, or deletes items in the table, DynamoDB Streams writes a stream record with the primary key attribute(s) of the items that were modified. A *stream record* contains information about a data modification to a single item in a DynamoDB table. You can configure 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/

8. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

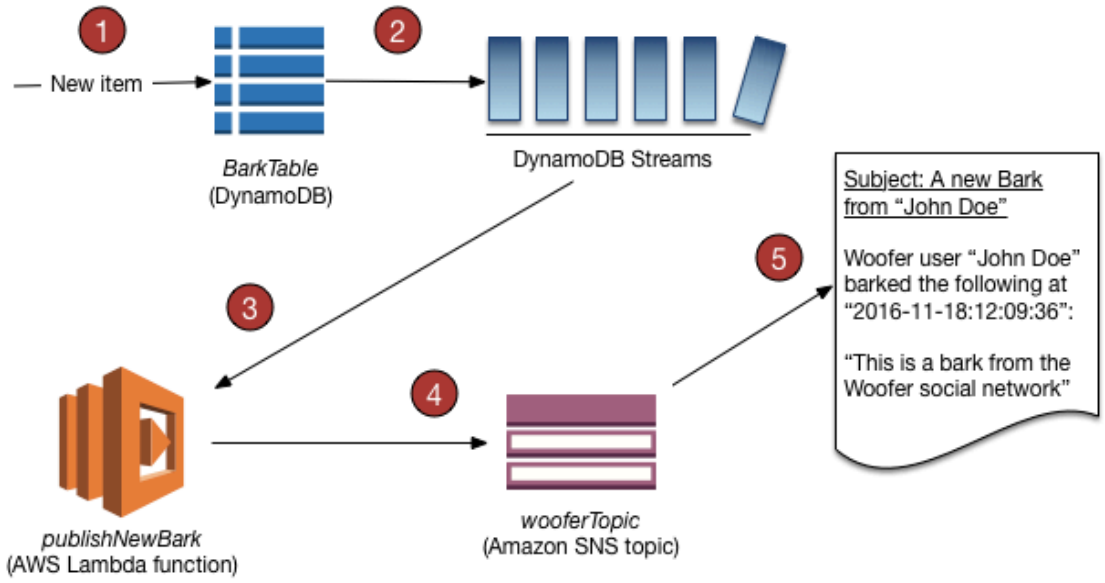
Use Systems Manager Automation to detect new entries in the DynamoDB table then automatically invoke the Lambda function for processing.
Enable DynamoDB Streams to capture table activity and automatically trigger the Lambda function. (Correct)

Invoke the Lambda functions using SNS each time that the ECS Cluster successfully processed financial data.

Use CloudWatch Alarms to trigger the Lambda function whenever a new entry is created in the DynamoDB table.

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

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



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

Suppose you have a mobile gaming app that writes to a `TutorialsDojoCourses` table. Whenever the `TopCourse` attribute of the `TutorialsDojoScores` table is updated, a corresponding stream record is written to the table's stream. This event could then trigger a Lambda function that posts a congratulatory message on a

social media network. (The function would simply ignore any stream records that are not updated to `TutorialsDojoCourses` or that do not modify the `TopCourse` attribute.)

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

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

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

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

References:

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

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

Check out this Amazon DynamoDB cheat sheet:

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

Topic-Based – Auto Scaling (SA-Associate)

1. QUESTION

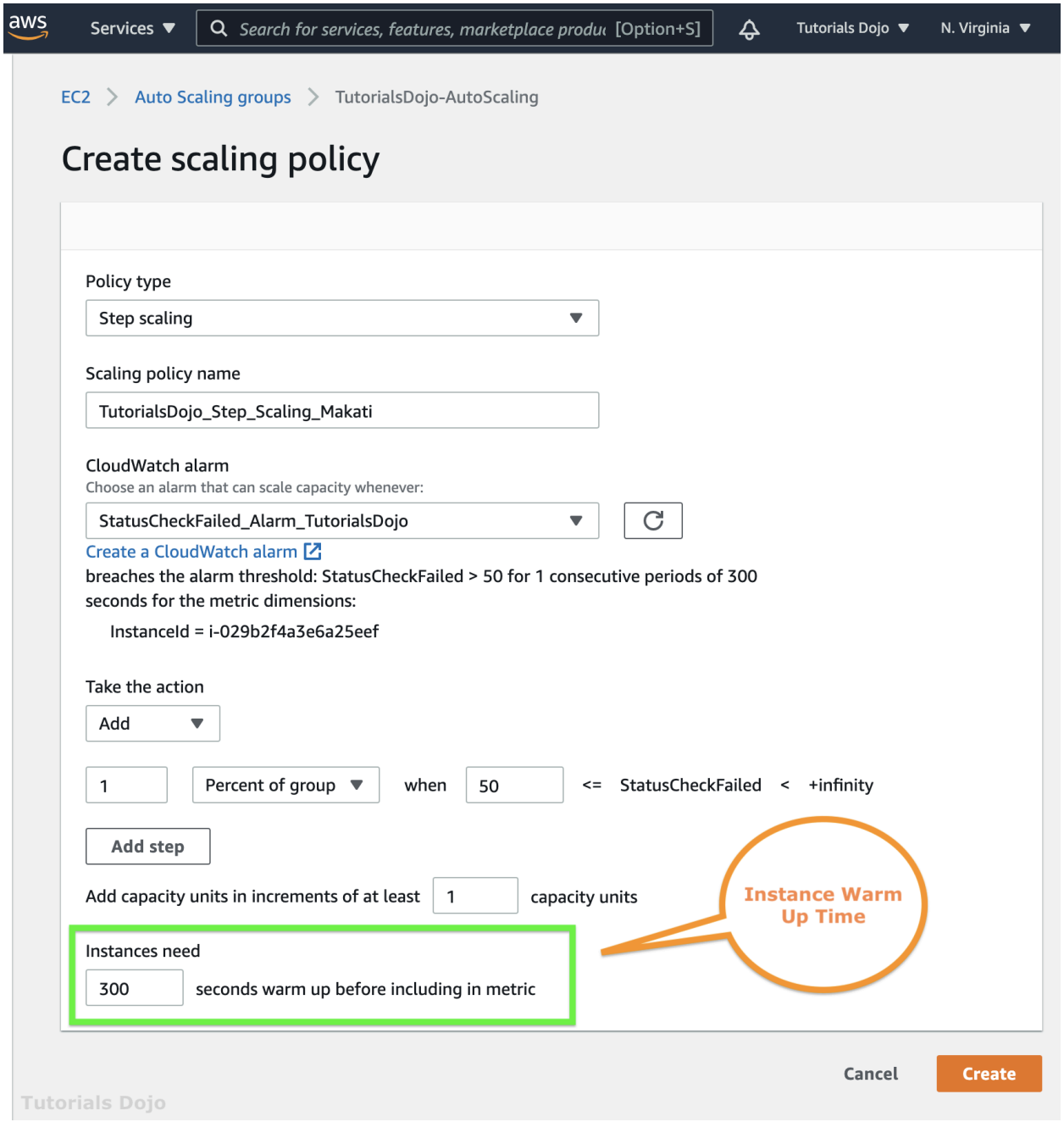
Category: CSAA – Design Resilient Architectures

A web application hosted in an Auto Scaling group of EC2 instances in AWS. The application receives a burst of traffic every morning, and a lot of users are complaining about request timeouts. The EC2 instance takes 1 minute to boot up before it can respond to user requests. The cloud architecture must be redesigned to better respond to the changing traffic of the application.

How should the Solutions Architect redesign the architecture?

- Create a Network Load Balancer with slow-start mode.
- Create a step scaling policy and configure an instance warm-up time condition. (Correct)
- Create a new launch template and upgrade the size of the instance.
- Create a CloudFront distribution and set the EC2 instance as the origin.

Amazon EC2 Auto Scaling helps you maintain application availability and allows you to automatically add or remove EC2 instances according to conditions you define. You can use the fleet management features of EC2 Auto Scaling to maintain the health and availability of your fleet. You can also use the dynamic and predictive scaling features of EC2 Auto Scaling to add or remove EC2 instances. Dynamic scaling responds to changing demand and predictive scaling automatically schedules the right number of EC2 instances based on predicted demand. Dynamic scaling and predictive scaling can be used together to scale faster.



Step scaling applies "step adjustments" which means you can set multiple actions to vary the scaling depending on the size of the alarm breach. When you create a step scaling policy, you can also specify the number of seconds that it takes for a newly launched instance to warm up.

Hence, the correct answer is: Create a step scaling policy and configure an instance warm-up time condition.

The option that says: **Create a Network Load Balancer with slow start mode** is incorrect because Network Load Balancer does not support slow start mode. If you need to enable slow start mode, you should use Application Load Balancer.

The option that says: **Create a new launch template and upgrade the size of the instance** is incorrect because a larger instance does not always improve the boot time. Instead of upgrading the instance, you should create a step scaling policy and add a warm-up time.

The option that says: **Create a CloudFront distribution and set the EC2 instance as the origin** is incorrect because this approach only resolves the traffic latency. Take note that the requirement in the scenario is to resolve the timeout issue and not the traffic latency.

References:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-simple-step.html

https://aws.amazon.com/ec2/autoscaling/faqs/

Check out these AWS Cheat Sheets:

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

https://tutorialsdojo.com/step-scaling-vs-simple-scaling-policies-in-amazon-ec2/

2. 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?

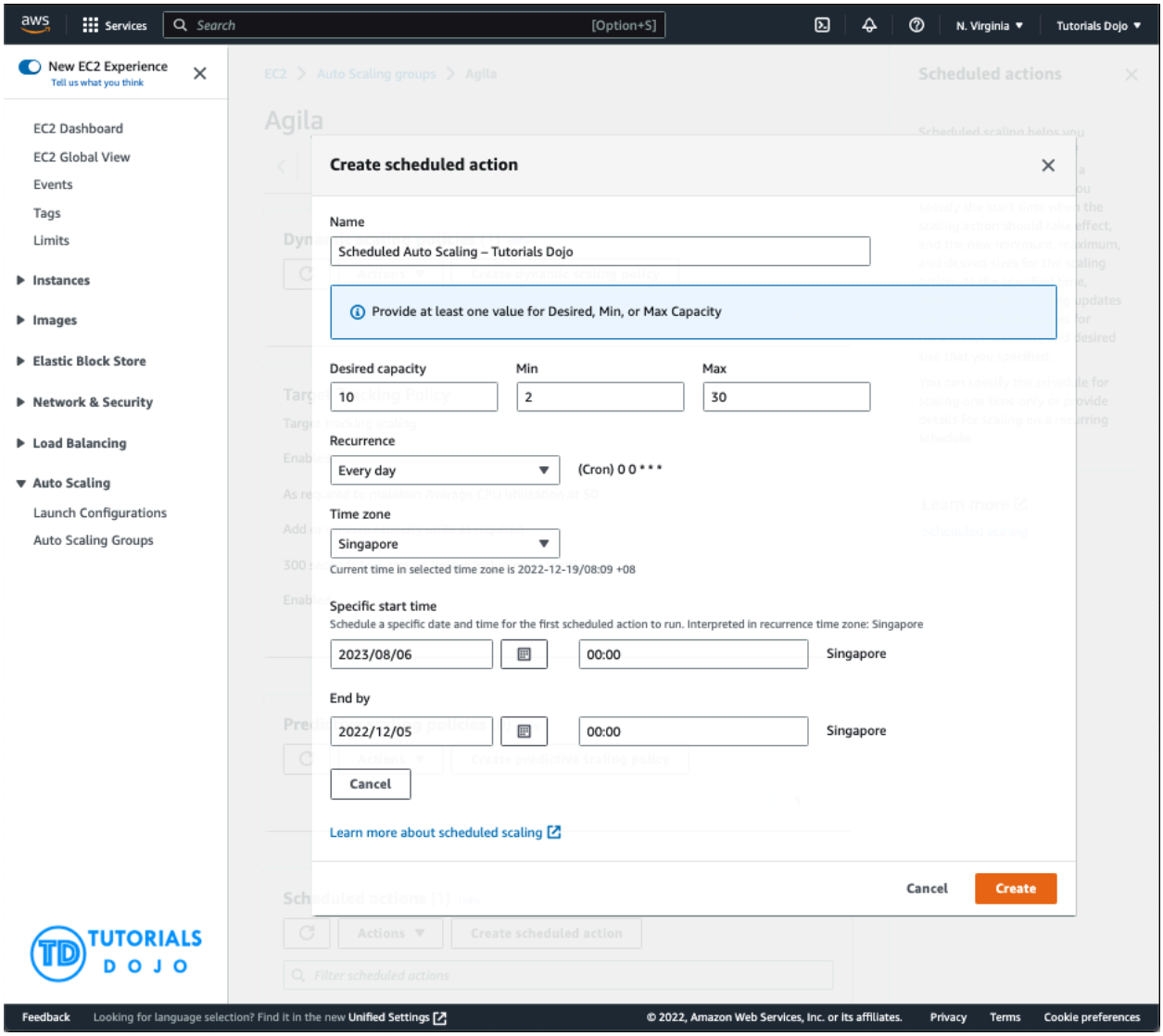
Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the CPU utilization.

Configure a Dynamic scaling policy for the Auto Scaling group to launch new instances based on the Memory utilization.

Configure a Scheduled scaling policy for the Auto Scaling group to launch new instances before the start of the day. (Correct)

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/

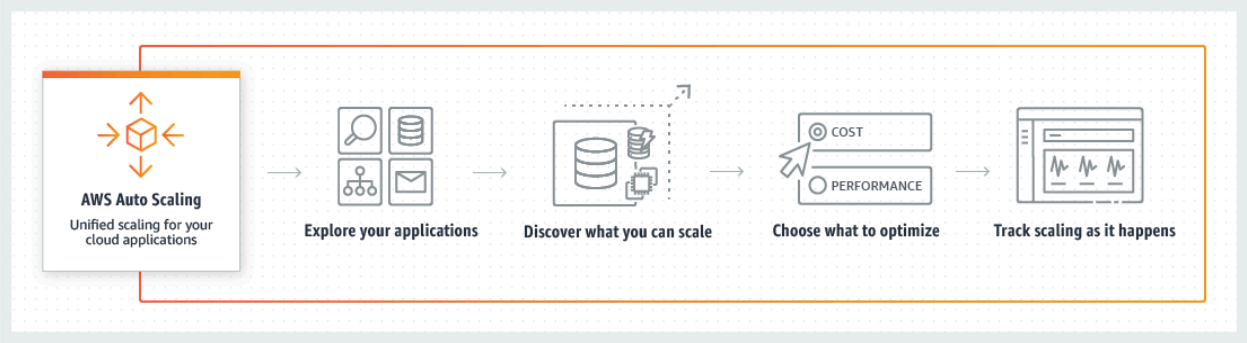
3. QUESTION

Category: CSAA – Design Resilient Architectures

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

Do nothing. You can start directly launching EC2 instances in the Auto Scaling group with the same launch template.
Create a new launch template. (Correct)
Create a new target group and launch template.
Create a new target group.

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



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

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

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

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

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

References:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/launch-templates.html

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

Check out this AWS Auto Scaling Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

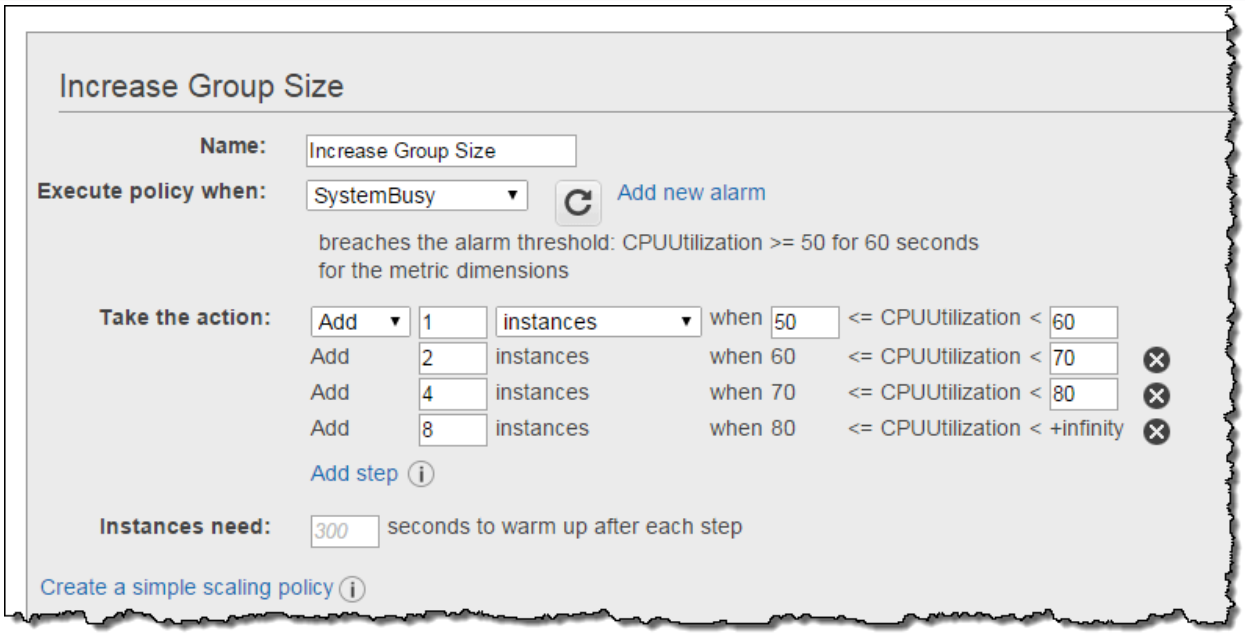
An application is hosted in an Auto Scaling group of EC2 instances. To improve the monitoring process, you have to configure the current capacity to increase or decrease based on a set of scaling adjustments. This should be done by specifying the scaling metrics and threshold values for the CloudWatch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

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

Scheduled Scaling
Simple scaling
Step scaling (Correct)
Target tracking scaling

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

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



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

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

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

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

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

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

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

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

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

References:

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

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

5. QUESTION

Category: CSAA – Design Resilient Architectures

A major TV network has a web application running on eight Amazon T3 EC2 instances behind an application load balancer. The number of requests that the application processes are consistent and do not experience spikes. A Solutions Architect must configure an Auto Scaling group for the instances to ensure that the application is running at all times.

Which of the following options can satisfy the given requirements?

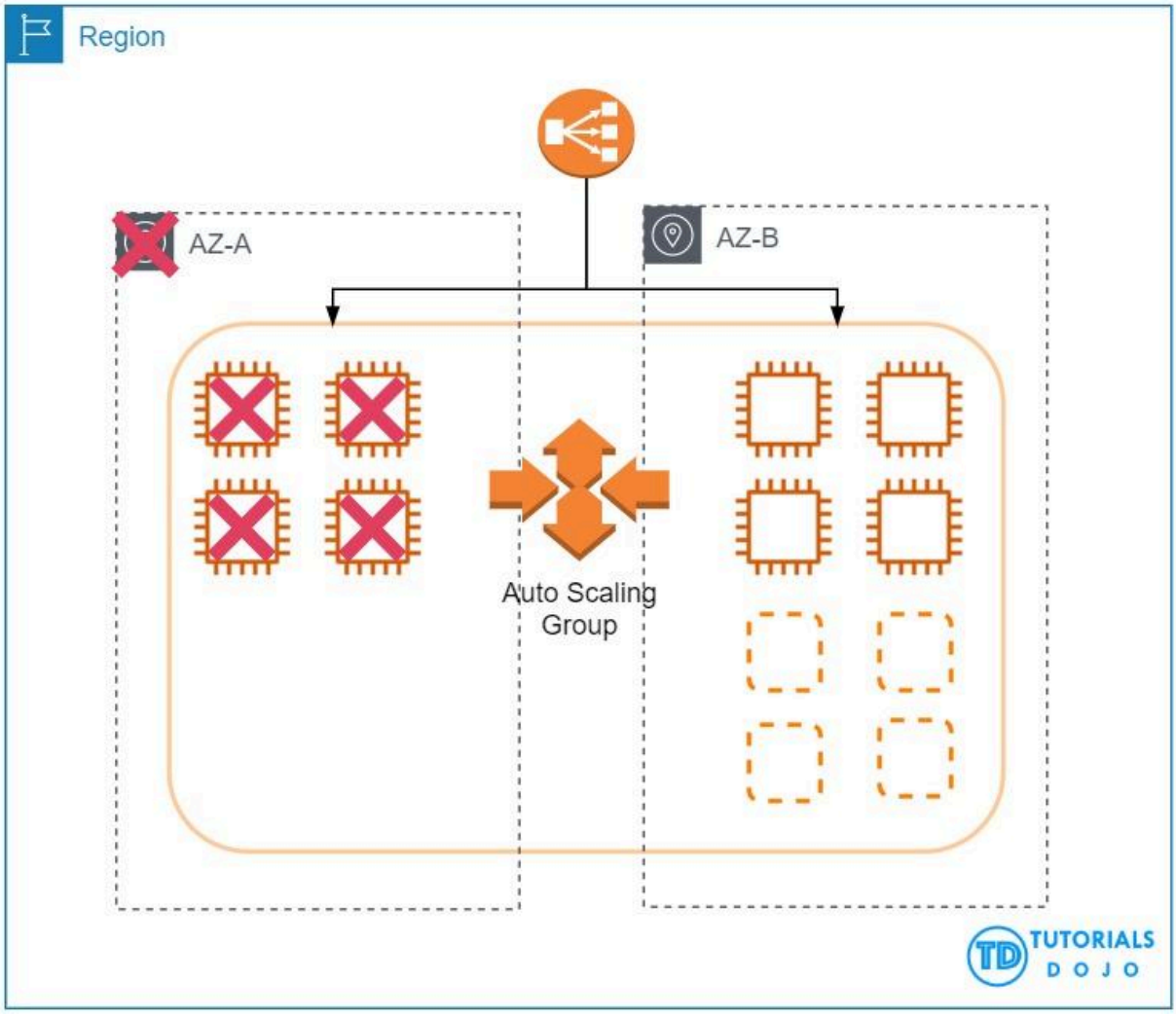
Deploy four EC2 instances with Auto Scaling in one region and four in another region behind an Amazon Elastic Load Balancer.

Deploy eight EC2 instances with Auto Scaling in one Availability Zone behind an Amazon Elastic Load Balancer.

Deploy four EC2 instances with Auto Scaling in one Availability Zone and four in another availability zone in the same region behind an Amazon Elastic Load Balancer. (Correct)

Deploy two EC2 instances with Auto Scaling in four regions behind an Amazon Elastic Load Balancer.

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



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

The 110% compute capacity for the 4 servers might cause some degradation of the service but not a total outage since there are still some instances that handle the requests. Depending on your scale-up configuration in your Auto Scaling group, the additional 4 EC2 instances can be launched in a matter of minutes.

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

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

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

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

References:

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

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

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

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

6. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

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

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

The following options are incorrect:

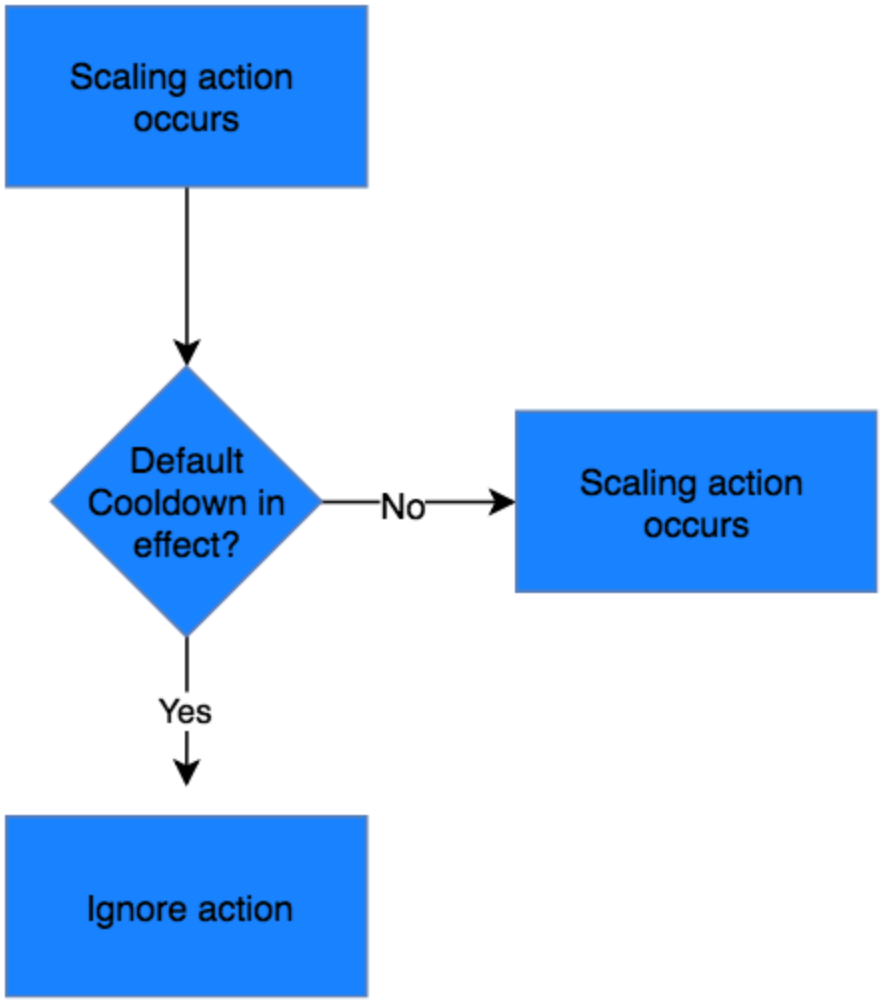
– It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.

– It ensures that the Auto Scaling group launches or terminates additional EC2 instances without any downtime.

– Its default value is 600 seconds.

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

The figure below demonstrates the scaling cooldown:



Reference:

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

Check out this AWS Auto Scaling Cheat Sheet:

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

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

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

7. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

The instance will be randomly selected by the Auto Scaling group
The EC2 instance launched from the oldest launch template. (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:

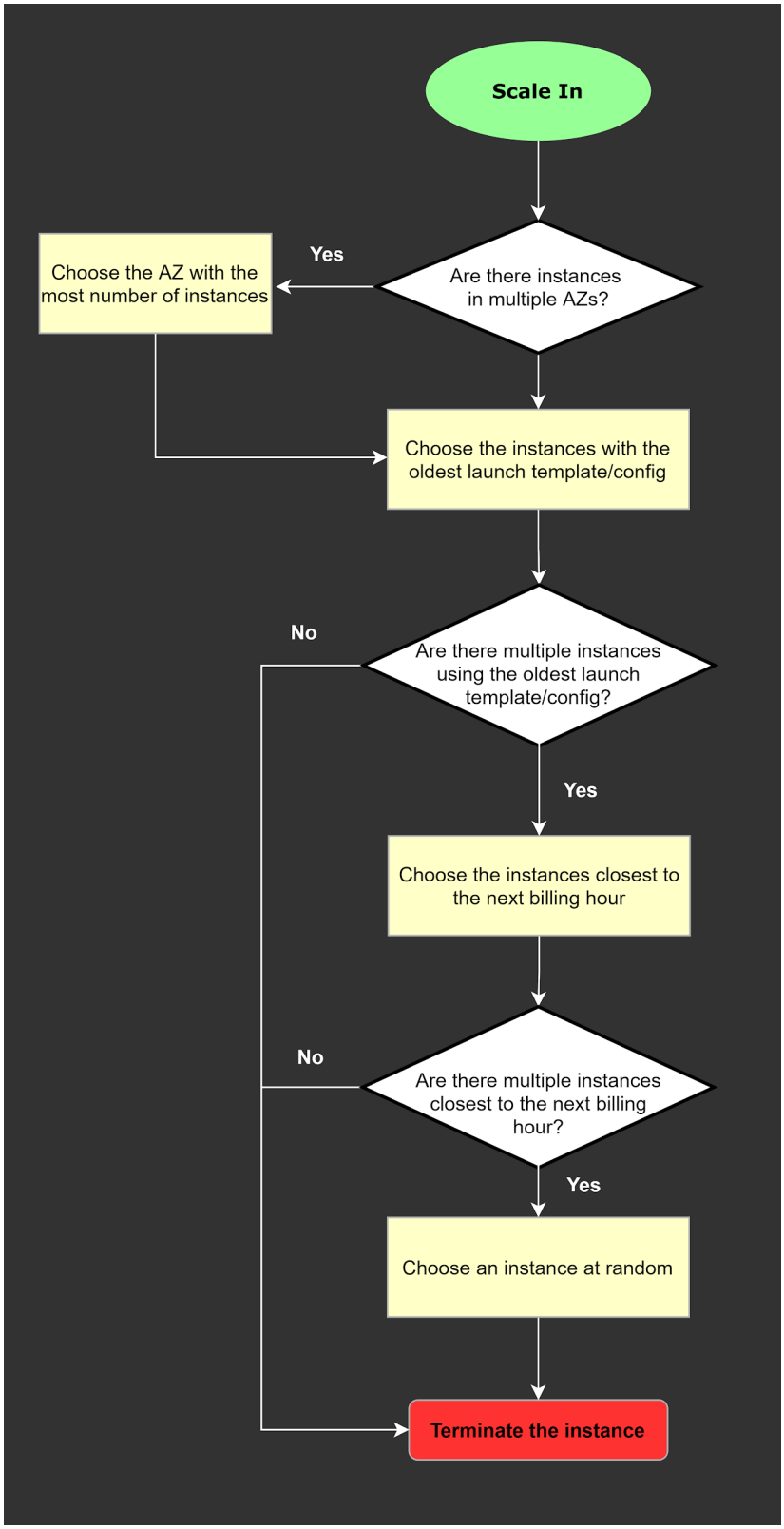
1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that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

2. Determine which unprotected instances in the selected Availability Zone use the oldest launch template. 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:



Hence, the correct answer is: **The EC2 instance launched from the oldest launch template.**

The option that says: **The EC2 instance which has the least number of user sessions** is incorrect because the number of user sessions is not a factor considered by Amazon EC2 Auto Scaling groups when deciding which instances to terminate during a scale-in event.

The option that says: **The EC2 instance which has been running for the longest time** is incorrect because the duration for which an EC2 instance has been running is not a factor considered by Amazon EC2 Auto Scaling groups when deciding which instances to terminate during a scale-in event.

The option that says: **The instance will be randomly selected by the Auto Scaling group** is incorrect because Amazon EC2 Auto Scaling groups do not randomly select instances for termination during a scale-in event.

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

Launch multiple EC2 instances behind an Application Load Balancer to host your application services, and SWF which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.

Launch multiple On-Demand EC2 instances to host your application services and an SQS queue which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.

Launch an Auto-Scaling group of EC2 instances to host your application services and an SQS queue. Include an Auto Scaling trigger to watch the SQS queue size which will either scale in or scale out the number of EC2 instances based on the queue. (Correct)

Launch multiple EC2 instances behind an Application Load Balancer to host your application services and SNS which will act as a highly-scalable buffer that stores messages as they travel between distributed applications.

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

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

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

Topic-Based – EC2 (SA-Associate)

1. 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.

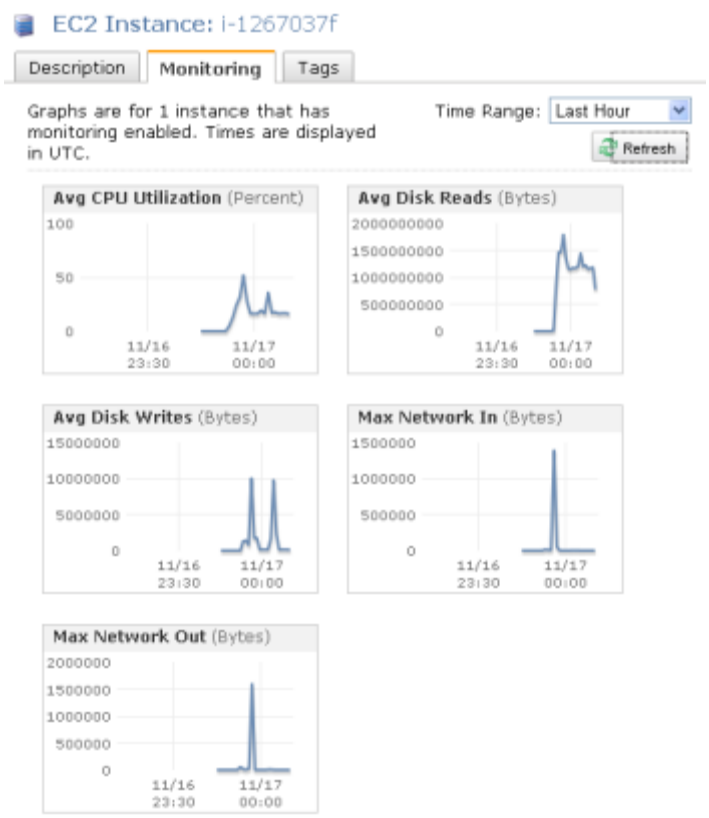
Which of the following is a custom metric in CloudWatch which you have to manually set up?

CPU Utilization of an EC2 instance
Network packets out of an EC2 instance
Disk Reads activity 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/

2. QUESTION

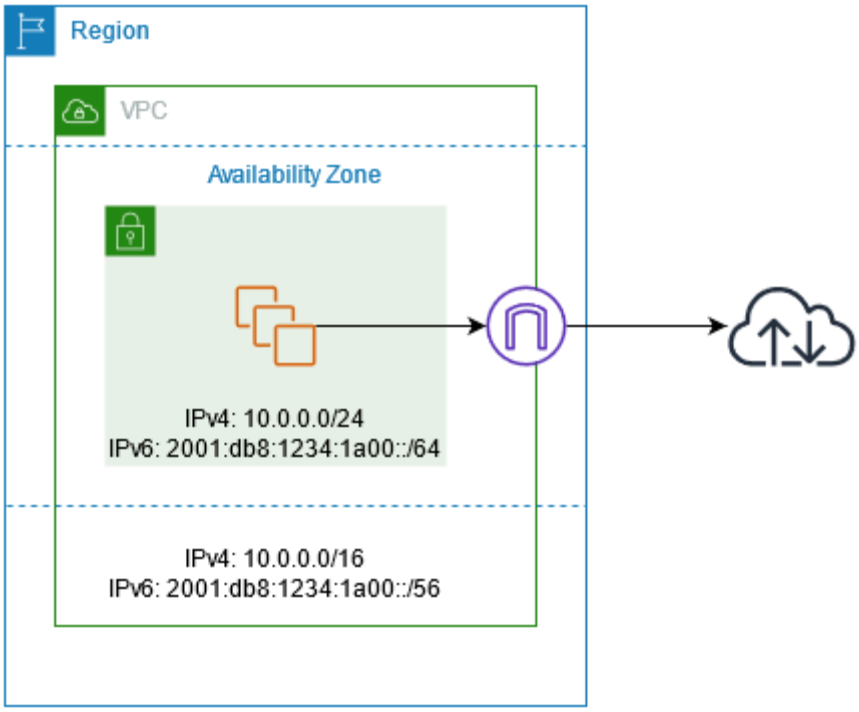
Category: CSAA – Design Secure Architectures

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 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.**
- **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 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 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)

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.

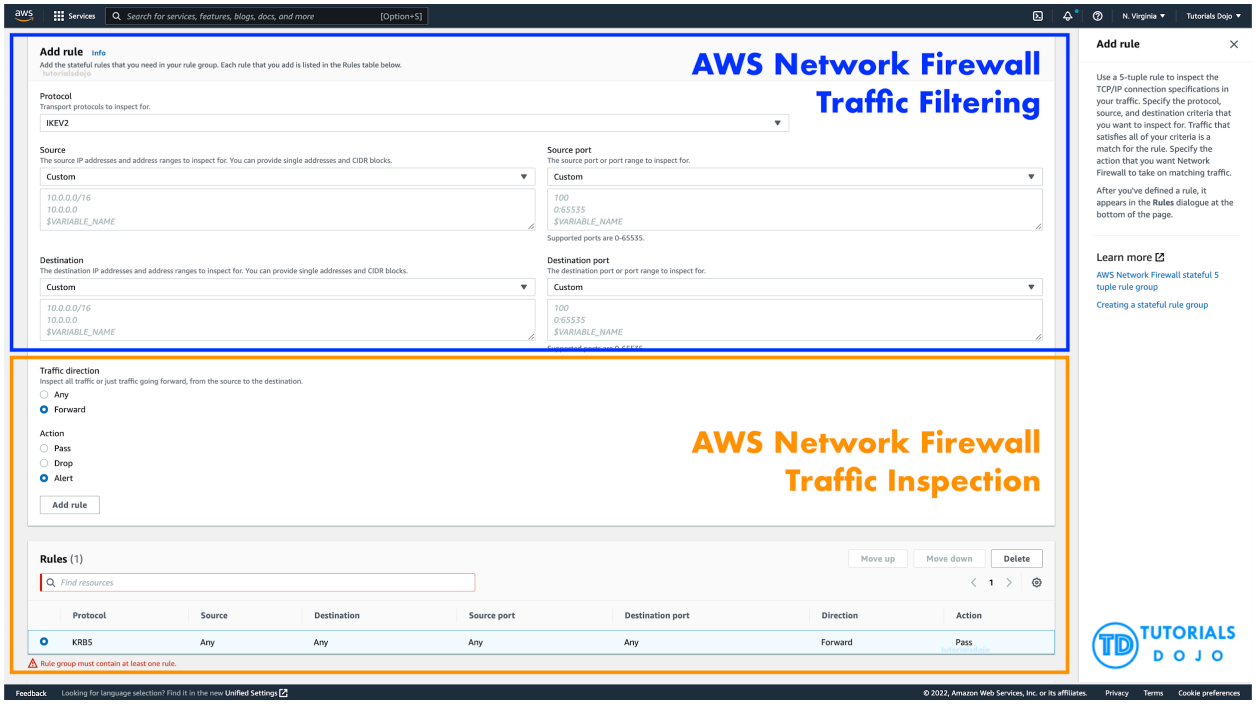


Destination	Target
10.0.0.0/16	Local
2001:db8:1234:1a00:/64	Local
::/0	*eigw-id*

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/

3. QUESTION

Category: CSAA – Design Resilient Architectures

You are automating the creation of EC2 instances in your VPC. Hence, you wrote a python script to trigger the Amazon EC2 API to request 50 EC2 instances in a single Availability Zone. However, you noticed that after 20 successful requests, subsequent requests failed.

What could be a reason for this issue and how would you resolve it?

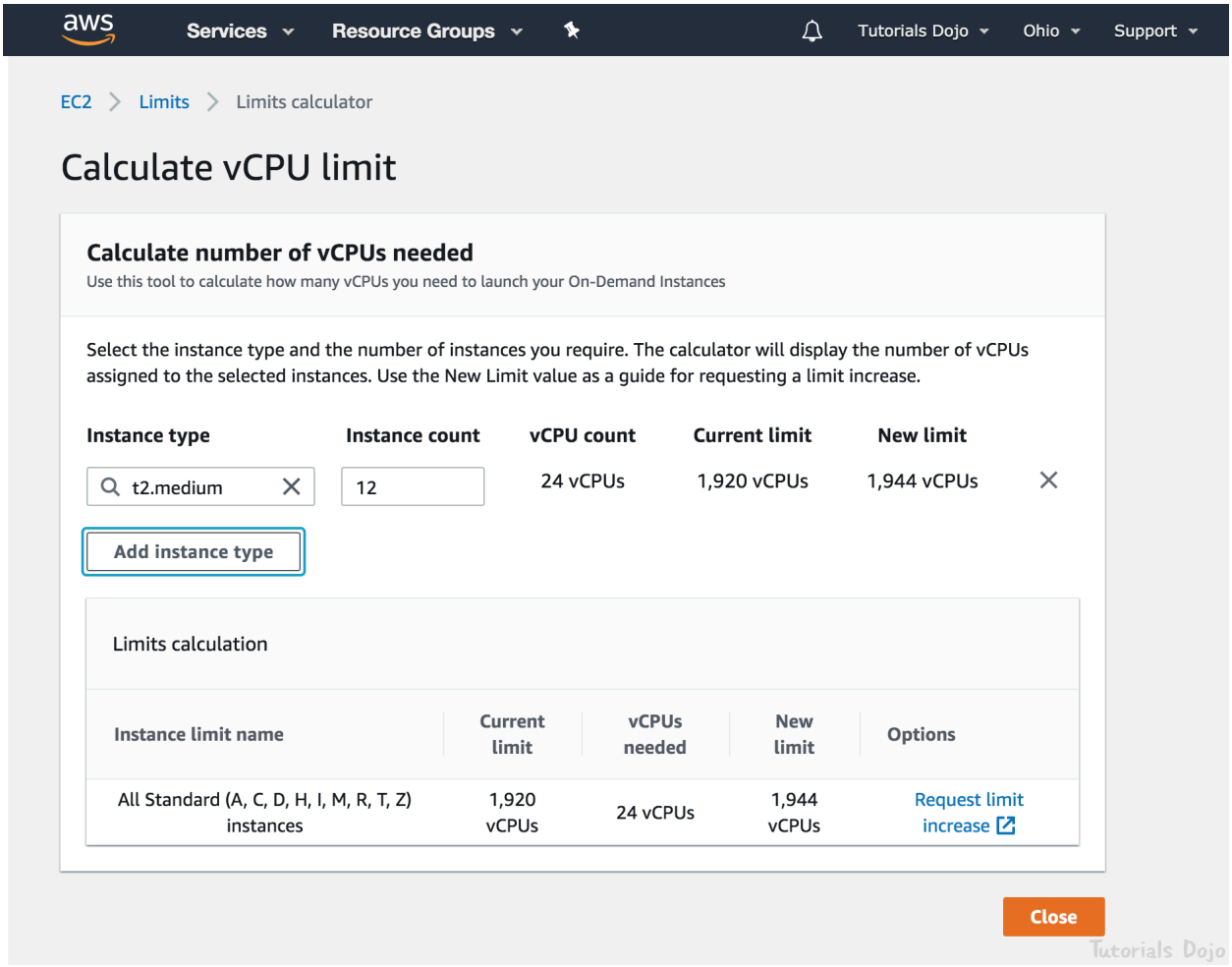
There was an issue with the Amazon EC2 API. Just resend the requests and these will be provisioned successfully.

By default, AWS allows you to provision a maximum of 20 instances per Availability Zone. Select a different Availability Zone and retry the failed request.

There is a vCPU-based On-Demand Instance limit per region which is why subsequent requests failed. Just submit the limit increase form to AWS and retry the failed requests once approved. (Correct)

By default, AWS allows you to provision a maximum of 20 instances per region. Select a different region and retry the failed request.

You are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region. New AWS accounts may start with limits that are lower than the limits described here.



If you need more instances, complete the Amazon EC2 limit increase request form with your use case, and your limit increase will be considered. Limit increases are tied to the region they were requested for.

Hence, the correct answer is: **There is a vCPU-based On-Demand Instance limit per region which is why subsequent requests failed. Just submit the limit increase form to AWS and retry the failed requests once approved.**

The option that says: **There was an issue with the Amazon EC2 API. Just resend the requests and these will be provisioned successfully** is incorrect because you are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit. There is also a limit of purchasing 20 Reserved Instances and requesting Spot Instances per your dynamic Spot limit per region hence, there is no problem with the EC2 API.

The option that says: **By default, AWS allows you to provision a maximum of 20 instances per region. Select a different region and retry the failed request** is incorrect. There is no need to select a different region since this limit can be increased after submitting a request form to AWS.

The option that says: **By default, AWS allows you to provision a maximum of 20 instances per Availability Zone. Select a different Availability Zone and retry the failed request** is incorrect because the vCPU-based On-Demand Instance limit is set per region and not per Availability Zone. This can be increased after submitting a request form to AWS.

References:

https://docs.aws.amazon.com/general/latest/gr/aws_service_limits.html#limits_ec2

https://aws.amazon.com/ec2/faqs/#How_many_instances_can_I_run_in_Amazon_EC2

Check out this Amazon EC2 Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

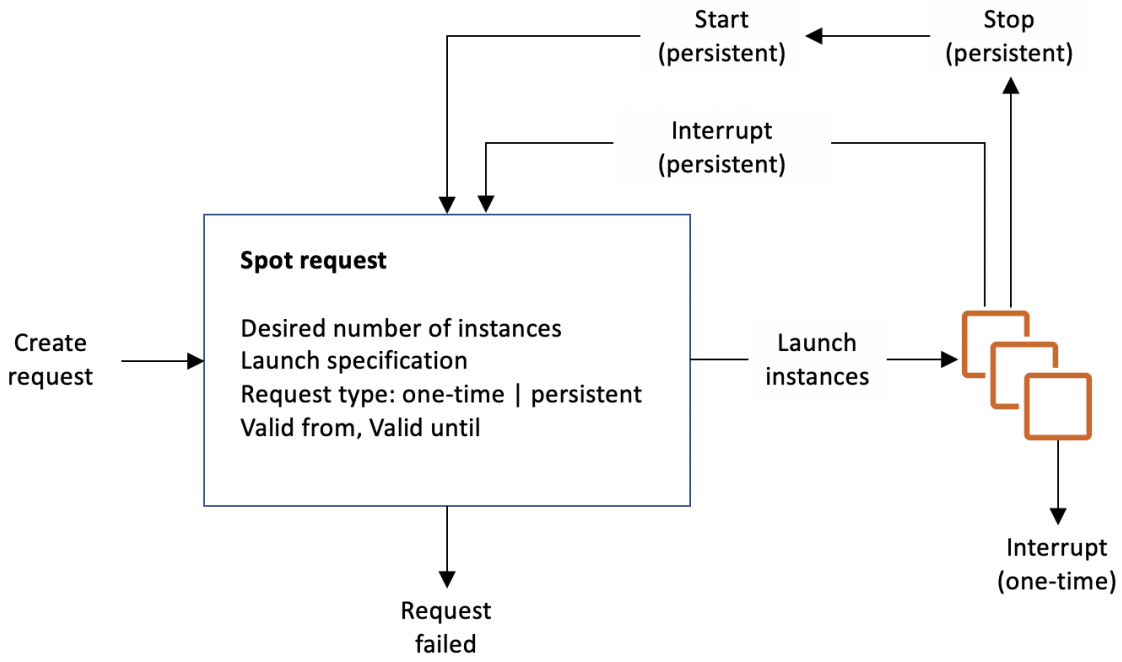
The media company that you are working for has a video transcoding application running on Amazon EC2. Each EC2 instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based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

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

Spot instances (Correct)
Reserved instances
On-demand instances

Dedicated instances

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



Amazon EC2 Spot instances are spare compute capacity in the AWS cloud available to you at steep discounts compared to On-Demand prices. EC2 Spot enables you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget. By simply selecting Spot when launching EC2 instances, you can save up to 90% on On-Demand prices. The only difference between On-Demand instances and Spot Instances is that Spot instances can be interrupted by EC2 with two minutes of notification when the EC2 needs the capacity back.

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

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

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

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

References:

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

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

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

Check out this Amazon EC2 Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Secure Architectures

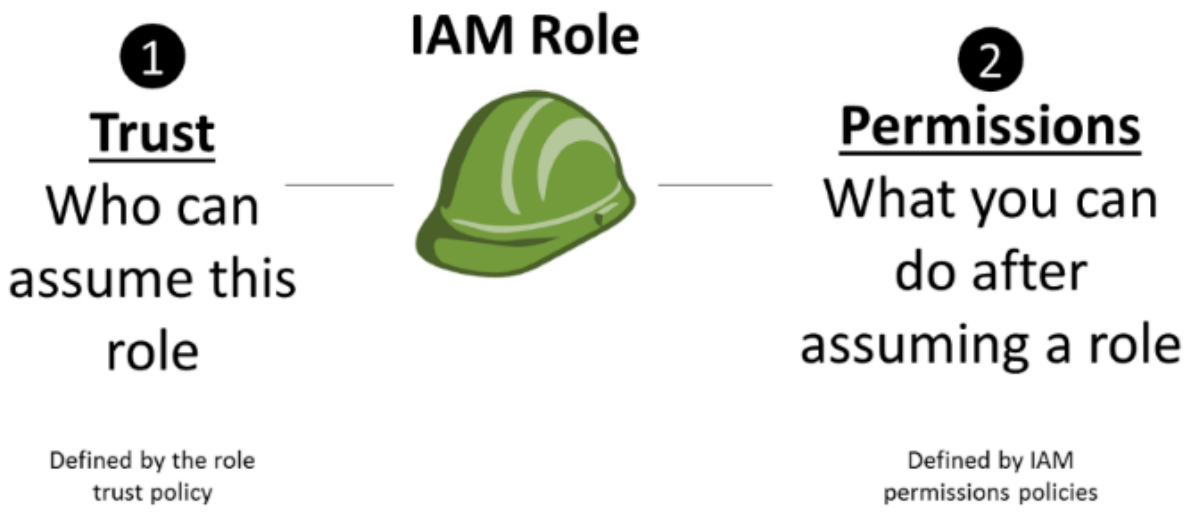
A company developed a meal planning application that provides meal recommendations for the week as well as the food consumption of the users. The application resides on an EC2 instance which requires access to various AWS services for its day-to-day operations.

Which of the following is the best way to allow the EC2 instance to access the S3 bucket and other AWS services?

Store the API credentials in the EC2 instance.
Add the API Credentials in the Security Group and assign it to the EC2 instance.
Store the API credentials in a bastion host.

Create a role in IAM and assign it to the EC2 instance. (Correct)

The best practice in handling API Credentials is to create a new role in the Identity Access Management (IAM) service and then assign it to a specific EC2 instance. In this way, you have a secure and centralized way of storing and managing your credentials.



Storing the API credentials in the EC2 instance, **adding the API Credentials in the Security Group and assigning it to the EC2 instance**, and **storing the API credentials in a bastion host** are incorrect because it is not secure to store nor use the API credentials from an EC2 instance. You should use IAM service instead.

Reference:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

Check out this AWS IAM Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Resilient Architectures

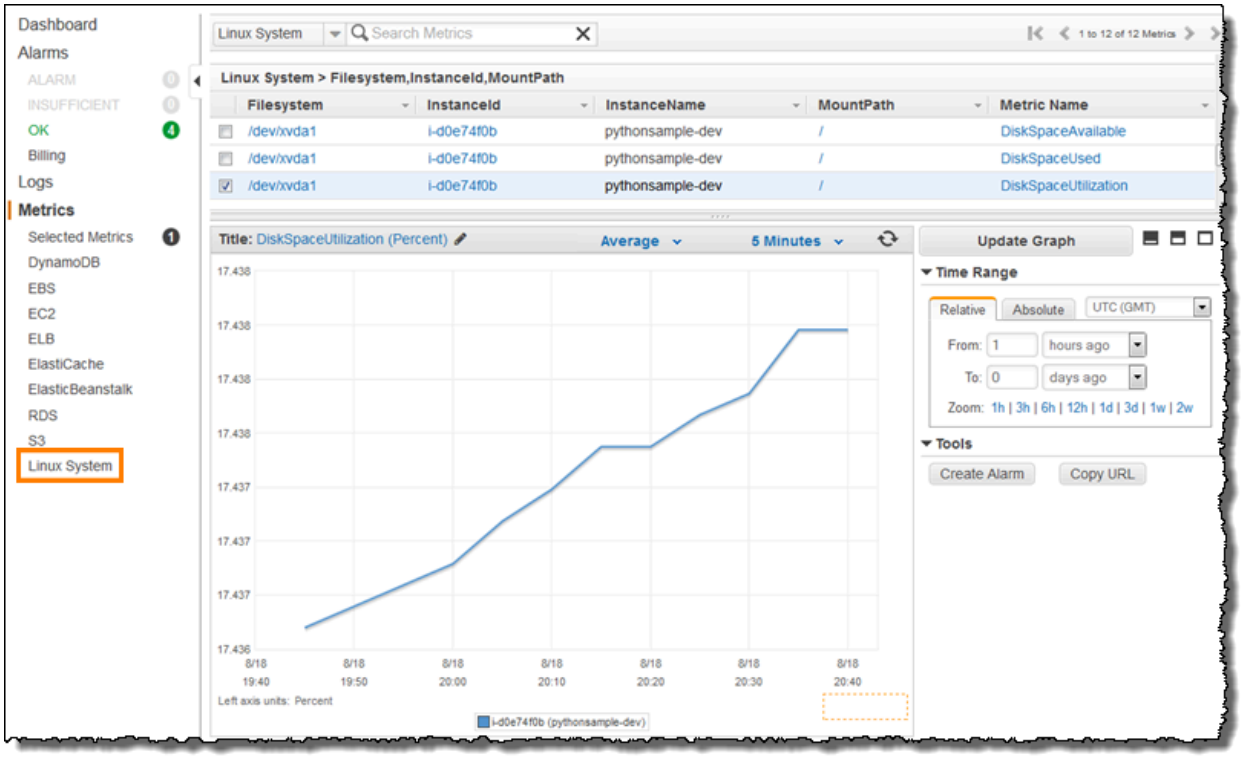
A company has a cloud architecture that is composed of Linux and Windows EC2 instances that process high volumes of financial data 24 hours a day, 7 days a week. To ensure high availability of the systems, the Solutions Architect

needs to create a solution that allows them to monitor the memory and disk utilization metrics of all the instances.
Which of the following is the most suitable monitoring solution to implement?

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

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/

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A multinational corporate and investment bank regularly processes steady workloads of accruals, loan interests, and other critical financial calculations every night from 10 PM to 3 AM on their on-premises data center for their corporate clients. Once the process is done, the results are then uploaded to the Oracle General Ledger which means that the processing should not be delayed or interrupted. The CTO has decided to move its IT infrastructure to AWS to save costs. The company needs to reserve compute capacity in a specific Availability Zone to properly run their workloads.

As the Senior Solutions Architect, how can you implement a cost-effective architecture in AWS for their financial system?

Use On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second. Reserve compute capacity in a specific Availability Zone to avoid any interruption.

Use Regional Reserved Instances to reserve capacity on a specific Availability Zone and lower the operating cost through its billing discounts.

Use On-Demand Capacity Reservations, which provide compute capacity that is always available on the specified recurring schedule. (Correct)

Use Dedicated Hosts, which provide a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.

On-Demand Capacity Reservations enable you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. This gives you the ability to create and manage Capacity Reservations independently from the billing discounts offered by Savings Plans or Regional Reserved Instances.

By creating Capacity Reservations, you ensure that you always have access to EC2 capacity when you need it, for as long as you need it. You can create Capacity Reservations at any time, without entering into a one-year or three-year term commitment, and the capacity is available immediately. Billing starts as soon as the capacity is provisioned and the Capacity Reservation enters the active state. When you no longer need it, cancel the Capacity Reservation to stop incurring charges.

	Capacity Reservations	Zonal Reserved Instances	Regional Reserved Instances	Savings Plans
Term	No commitment required. Can be created and canceled as needed.	Requires a fixed one-year or three-year commitment		
Capacity benefit	Capacity reserved in a specific Availability Zone.		No capacity reserved.	
Billing discount	No billing discount. †	Provides a billing discount.		
Instance Limits	Your On-Demand Instance limits per Region apply.	Default is 20 per Availability Zone. You can request a limit increase.	Default is 20 per Region. You can request a limit increase.	No limit.

When you create a Capacity Reservation, you specify:

– The Availability Zone in which to reserve the capacity

– The number of instances for which to reserve capacity

– The instance attributes, including the instance type, tenancy, and platform/OS

Capacity Reservations can only be used by instances that match their attributes. By default, they are automatically used by running instances that match the attributes. If you don't have any running instances that match the attributes of the Capacity Reservation, it remains unused until you launch an instance with matching attributes.

In addition, you can use Savings Plans and Regional Reserved Instances with your Capacity Reservations to benefit from billing discounts. AWS automatically applies your discount when the attributes of a Capacity Reservation match the attributes of a Savings Plan or Regional Reserved Instance.

In this scenario, the company only runs the process for 5 hours (from 10 PM to 3 AM) every night. By usinng Capacity Reservations, they not only ensure availability but can also implement automation to procure and cancel capacity, as well as terminate instances once they are no longer needed. This approach prevents them from incurring unnecessary charges, ensuring they are billed only for the resources they actually use.

Hence, the correct answer is to **use On-Demand Capacity Reservations, which provide compute capacity that is always available on the specified recurring schedule.**

The option that says: **Use On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second. Reserve compute capacity in a specific Availability Zone to avoid any interruption** is incorrect because although an On-Demand instance is stable and suitable for processing critical data, it costs more than any other option. Moreover, the critical financial calculations are only done every night from 10 PM to 3 AM and not 24/7. This means that your computing capacity will not be utilized for a total of 19 hours every single day. On-Demand instances cannot reserve compute capacity at all. So this option is incorrect.

The option that says: **Use Regional Reserved Instances to reserve capacity on a specific Availability Zone and lower the operating cost through its billing discounts.** is incorrect because this feature is available in Zonal Reserved Instances only and not on Regional Reserved Instances.

The option that says: **Use Dedicated Hosts, which provide a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs** is incorrect because the use of a fully dedicated physical host is not warranted in this scenario. Moreover, this will be underutilized since you only run the process for 5 hours (from 10 PM to 3 AM only), wasting 19 hours of compute capacity every single day.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-capacity-reservations.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html

Check out this Amazon EC2 Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design Resilient Architectures

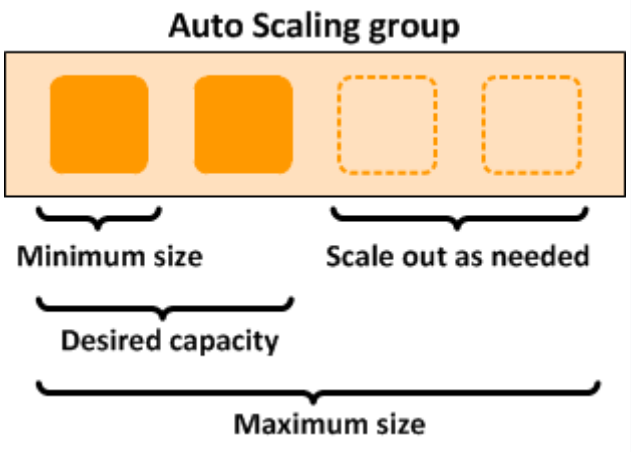
A company needs to deploy at least 2 EC2 instances to support the normal workloads of its application and automatically scale up to 6 EC2 instances to handle the peak load. The architecture must be highly available and fault-tolerant as it is processing mission-critical workloads.

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.
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/

Topic-Based – EBS (SA-Associate)

1. QUESTION

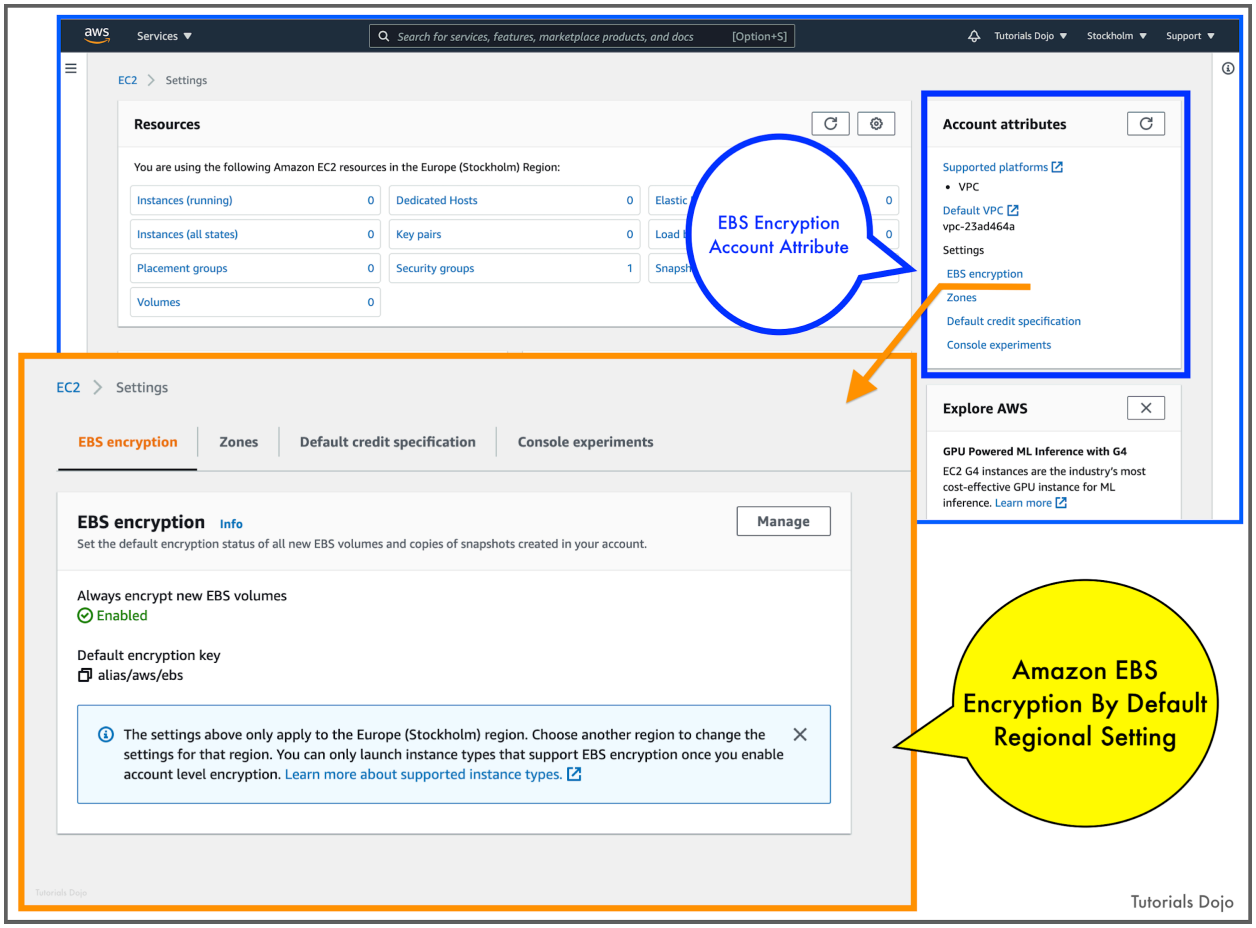
Category: CSAA – Design Secure Architectures

A company has several unencrypted EBS snapshots in their VPC. The Solutions Architect must ensure that all of the new EBS volumes restored from the unencrypted snapshots are automatically encrypted.

What should be done to accomplish this requirement?

- **Launch new EBS volumes and encrypt them using an asymmetric customer master key (CMK).**
- **Enable the EBS Encryption By Default feature for specific EBS volumes.**
- **Launch new EBS volumes and specify the symmetric customer master key (CMK) for encryption.**
- **Enable the EBS Encryption By Default feature for the AWS Region.** (Correct)

You can configure your AWS account to enforce the encryption of the new EBS volumes and snapshot copies that you create. For example, Amazon EBS encrypts the EBS volumes created when you launch an instance and the snapshots that you copy from an unencrypted snapshot.



Encryption by default has no effect on existing EBS volumes or snapshots. The following are important considerations in EBS encryption:

– Encryption by default is a Region-specific setting. If you enable it for a Region, you cannot disable it for individual volumes or snapshots in that Region.

– When you enable encryption by default, you can launch an instance only if the instance type supports EBS encryption.

– Amazon EBS does not support asymmetric CMKs.

You cannot change the CMK that is associated with an existing snapshot or encrypted volume. However, you can associate a different CMK during a snapshot copy operation so that the resulting copied snapshot is encrypted by the new CMK.

Although there is no direct way to encrypt an existing unencrypted volume or snapshot, you can encrypt them by creating either a volume or a snapshot. If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default key for EBS encryption. Even if you have not enabled encryption by default, you can enable encryption when you create an individual volume or snapshot. Whether you enable encryption by default or in individual

creation operations, you can override the default key for EBS encryption and use symmetric customer-managed CMK.

Hence, the correct answer is: **Enable the EBS Encryption By Default feature for the AWS Region.**

The option that says: **Launch new EBS volumes and encrypt them using an asymmetric customer master key (CMK)** is incorrect because Amazon EBS does not support asymmetric CMKs. To encrypt an EBS snapshot, you need to use symmetric CMK.

The option that says: **Launch new EBS volumes and specify the symmetric customer master key (CMK) for encryption** is incorrect. Although this solution will enable data encryption, this process is manual and can potentially cause some unencrypted EBS volumes to be launched. A better solution is to enable the EBS Encryption By Default feature. It is stated in the scenario that all of the new EBS volumes restored from the unencrypted snapshots must be automatically encrypted.

The option that says: **Enable the EBS Encryption By Default feature for specific EBS volumes** is incorrect because the Encryption By Default feature is a Region-specific setting and thus, you can't enable it to selected EBS volumes only.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html#encryption-by-default

https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

Check out this Amazon EBS Cheat Sheet:

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

Comparison of Amazon S3 vs Amazon EBS vs Amazon EFS:

https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/

2. QUESTION

Category: CSAA – Design High-Performing Architectures

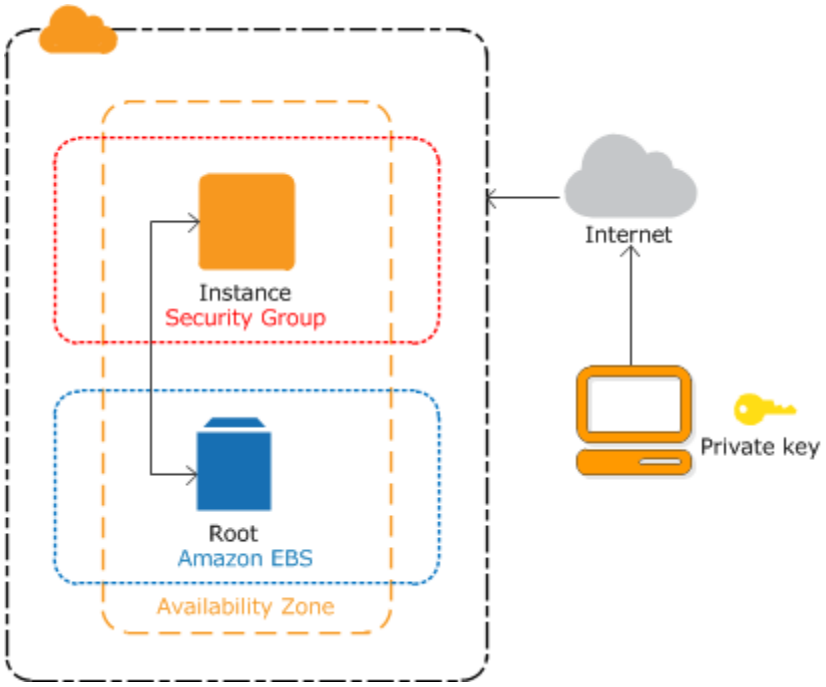
A technical lead of the Cloud Infrastructure team was consulted by a software developer regarding the required AWS resources of the web application that he is building. The developer knows that an Instance Store only provides ephemeral

storage where the data is automatically deleted when the instance is terminated. To ensure that the data of the web application persists, the app should be launched in an EC2 instance that has a durable, block-level storage volume attached. The developer knows that they need to use an EBS volume, but they are not sure what type they need to use.

In this scenario, which of the following is true about Amazon EBS volume types and their respective usage? (Select TWO.)

- General Purpose SSD (gp3) volumes with multi-attach enabled offer consistent and low-latency performance, and are designed for applications requiring multi-az resiliency.
- Spot volumes provide the lowest cost per gigabyte of all EBS volume types and are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important.
- Provisioned IOPS volumes offer storage with consistent and low-latency performance, and are designed for I/O intensive applications such as large relational or NoSQL databases. (Correct)
- Single root I/O virtualization (SR-IOV) volumes are suitable for a broad range of workloads, including small to medium-sized databases, development and test environments, and boot volumes.
- Magnetic volumes provide the lowest cost per gigabyte of all EBS volume types and are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important. (Correct)

Amazon EBS provides three volume types to best meet the needs of your workloads: General Purpose (SSD), Provisioned IOPS (SSD), and Magnetic.



General Purpose (SSD) is the new, SSD-backed, general purpose EBS volume type that is recommended as the default choice for customers. General Purpose (SSD) volumes are suitable for a broad range of workloads, including small to medium-sized databases, development and test environments, and boot volumes.

Provisioned IOPS (SSD) volumes offer storage with consistent and low-latency performance and are designed for I/O intensive applications such as large relational or NoSQL databases. Magnetic volumes provide the lowest cost per gigabyte of all EBS volume types.

Magnetic volumes are ideal for workloads where data are accessed infrequently, and applications where the lowest storage cost is important. Take note that this is a Previous Generation Volume. The latest low-cost magnetic storage types are Cold HDD (sc1) and Throughput Optimized HDD (st1) volumes.

Hence, the correct answers are:

– Provisioned IOPS volumes offer storage with consistent and low-latency performance, and are designed for I/O intensive applications such as large relational or NoSQL databases.

– Magnetic volumes provide the lowest cost per gigabyte of all EBS volume types and are ideal for workloads where data is accessed infrequently, and applications where the lowest storage cost is important.

The option that says: **Spot volumes provide the lowest cost per gigabyte of all EBS volume types and are ideal for workloads where data is accessed infrequently, and**

applications where the lowest storage cost is important is incorrect because there is no EBS type called a "Spot volume" however, there is an Instance purchasing option for Spot Instances.

The option that says: **General Purpose SSD (gp3) volumes with multi-attach enabled offer consistent and low-latency performance, and are designed for applications requiring multi-az resiliency** is incorrect because the multi-attach feature can only be enabled on EBS Provisioned IOPS io2 or io1 volumes. In addition, multi-attach won't offer multi-az resiliency because this feature only allows an EBS volume to be attached on multiple instances within an availability zone.

The option that says: **Single root I/O virtualization (SR-IOV) volumes are suitable for a broad range of workloads, including small to medium-sized databases, development and test environments, and boot volumes** is incorrect because SR-IOV is related with Enhanced Networking on Linux and not in EBS.

References:

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html

Check out this Amazon EBS Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design Resilient Architectures

An organization needs a persistent block storage volume that will be used for mission-critical workloads. The backup data will be stored in an object storage service and after 30 days, the data will be stored in a data archiving storage service.

What should you do to meet the above requirement?

Attach an instance store volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA.

Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 One Zone-IA.

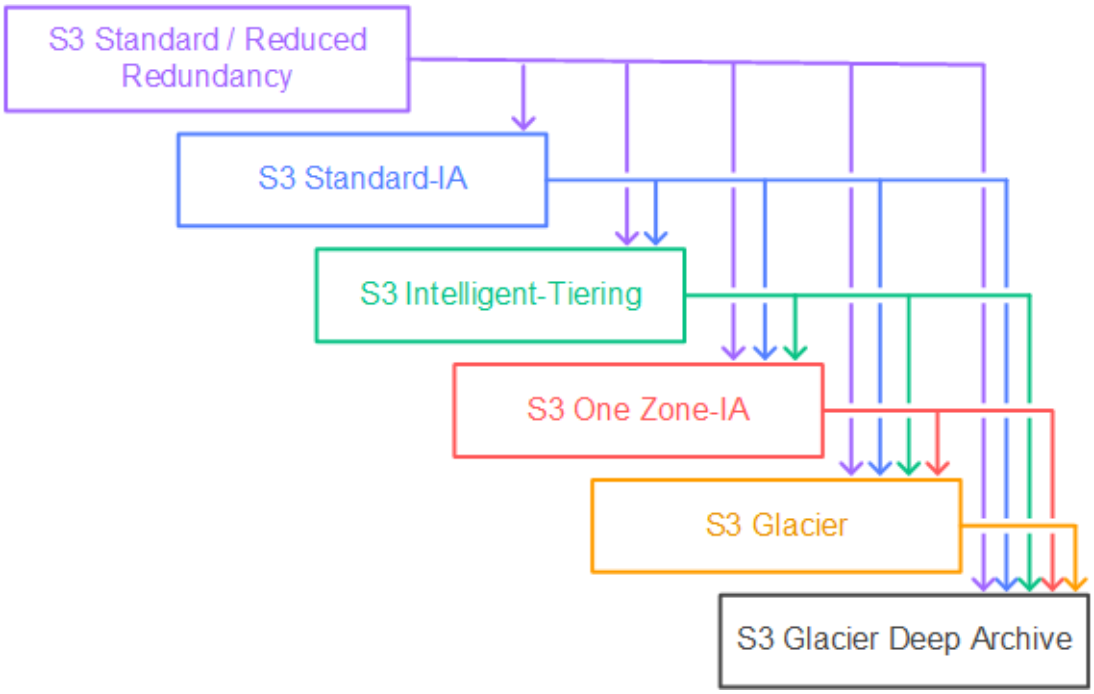
Attach an EBS volume in your EC2 instance. Use Amazon S3 to store your backup data and configure a lifecycle policy to transition your objects to Amazon S3 Glacier. (Correct)

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.

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/

4. QUESTION

Category: CSAA – Design Resilient Architectures

As part of the Business Continuity Plan of your company, your IT Director instructed you to set up an automated backup of all of the EBS Volumes for your EC2 instances as soon as possible.

What is the fastest and most cost-effective solution to automatically back up all of your EBS Volumes?

Use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots. (Correct)
Use an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes.

Set your Amazon Storage Gateway with EBS volumes as the data source and store the backups in your on-premises servers through the storage gateway.

For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically.

You can use Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation, retention, and deletion of snapshots taken to back up your Amazon EBS volumes. Automating snapshot management helps you to:

– Protect valuable data by enforcing a regular backup schedule.

– Retain backups as required by auditors or internal compliance.

– Reduce storage costs by deleting outdated backups.

Combined with the monitoring features of Amazon CloudWatch Events and AWS CloudTrail, Amazon DLM provides a complete backup solution for EBS volumes at no additional cost.

Hence, **using Amazon Data Lifecycle Manager (Amazon DLM) to automate the creation of EBS snapshots** is the correct answer as it is the fastest and most cost-effective solution that provides an automated way of backing up your EBS volumes.

The option that says: **For an automated solution, create a scheduled job that calls the "create-snapshot" command via the AWS CLI to take a snapshot of production EBS volumes periodically** is incorrect because even though this is a valid solution, you would still need additional time to create a scheduled job that calls the "create-snapshot" command. It would be better to use Amazon Data Lifecycle Manager (Amazon DLM) instead as this provides you the fastest solution which enables you to automate the creation, retention, and deletion of the EBS snapshots without having to write custom shell scripts or creating scheduled jobs.

Setting your Amazon Storage Gateway with EBS volumes as the data source and storing the backups in your on-premises servers through the storage gateway is incorrect as the Amazon Storage Gateway is used only for creating a backup of data from your on-premises server and not from the Amazon Virtual Private Cloud.

Using an EBS-cycle policy in Amazon S3 to automatically back up the EBS volumes is incorrect as there is no such thing as EBS-cycle policy in Amazon S3.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/snapshot-lifecycle.html

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html

Check out this Amazon EBS Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Resilient Architectures

A company plans to migrate all of their applications to AWS. The Solutions Architect suggested to store all the data to EBS volumes. The Chief Technical Officer is worried that EBS volumes are not appropriate for the existing workloads due to compliance requirements, downtime scenarios, and IOPS performance.

Which of the following are valid points in proving that EBS is the best service to use for migration? (Select TWO.)

- EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions. (Correct)
- An EBS volume is off-instance storage that can persist independently from the life of an instance. (Correct)
- EBS volumes can be attached to any EC2 Instance in any Availability Zone.
- When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component.
- Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to

Amazon RDS, where it is stored redundantly in multiple Availability Zones

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Here is a list of important information about EBS Volumes:

– When you create an EBS volume in an Availability Zone, it is automatically replicated within that zone to prevent data loss due to a failure of any single hardware component.

– After you create a volume, you can attach it to any EC2 instance in the same Availability Zone

– Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1) volume to multiple Nitro-based instances that are in the same Availability Zone. However, other EBS types are not supported.

– An EBS volume is off-instance storage that can persist independently from the life of an instance. You can specify not to terminate the EBS volume when you terminate the EC2 instance during instance creation.

– EBS volumes support live configuration changes while in production which means that you can modify the volume type, volume size, and IOPS capacity without service interruptions.

– Amazon EBS encryption uses 256-bit Advanced Encryption Standard algorithms (AES-256)

– EBS Volumes offer 99.999% SLA.

The option that says: **When you create an EBS volume in an Availability Zone, it is automatically replicated on a separate AWS region to prevent data loss due to a failure of any single hardware component** is incorrect because when you create an EBS volume in an Availability Zone, it is automatically replicated within that zone only, and not on a separate AWS region, to prevent data loss due to a failure of any single hardware component.

The option that says: **EBS volumes can be attached to any EC2 Instance in any Availability Zone** is incorrect as EBS volumes can only be attached to an EC2 instance in the same Availability Zone.

The option that says: **Amazon EBS provides the ability to create snapshots (backups) of any EBS volume and write a copy of the data in the volume to Amazon RDS, where it is stored redundantly in multiple Availability Zones** is almost correct. But instead of storing the volume to Amazon RDS, the EBS Volume snapshots are actually sent to Amazon S3.

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumes.html

https://aws.amazon.com/ebs/features/

Check out this Amazon EBS Cheat Sheet:

https://tutorialsdojo.com/aws-cheat-sheet-amazon-ebs/

6. QUESTION

Category: CSAA – Design Secure Architectures

A health organization is using a large Dedicated EC2 instance with multiple EBS volumes to host its health records web application. The EBS volumes must be encrypted due to the confidentiality of the data that they are handling and also to comply with the HIPAA (Health Insurance Portability and Accountability Act) standard.

In EBS encryption, what service does AWS use to secure the volume's data at rest? (Select TWO.)

By using Amazon-managed keys in AWS Key Management Service (KMS). (Correct)
By using your own keys in AWS Key Management Service (KMS). (Correct)
By using S3 Client-Side Encryption.
By using S3 Server-Side Encryption.

By using a password stored in CloudHSM.

By using the SSL certificates provided by the AWS Certificate Manager (ACM).

Amazon EBS encryption offers seamless encryption of EBS data volumes, boot volumes, and snapshots, eliminating the need to build and maintain a secure key management infrastructure. EBS encryption enables data at rest security by encrypting your data using Amazon-managed keys, or keys you create and manage using the AWS Key Management Service (KMS). The encryption occurs on the servers that host EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.

Hence, the correct answers are: **using your own keys in AWS Key Management Service (KMS)** and **using Amazon-managed keys in AWS Key Management Service (KMS).**

Using S3 Server-Side Encryption and **using S3 Client-Side Encryption** are both incorrect as these relate only to S3.

Using a password stored in CloudHSM is incorrect as you only store keys in CloudHSM and not passwords.

Using the SSL certificates provided by the AWS Certificate Manager (ACM) is incorrect as ACM only provides SSL certificates and not data encryption of EBS Volumes.

Reference:

https://aws.amazon.com/ebs/faqs/

Check out this Amazon EBS Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Secure Architectures

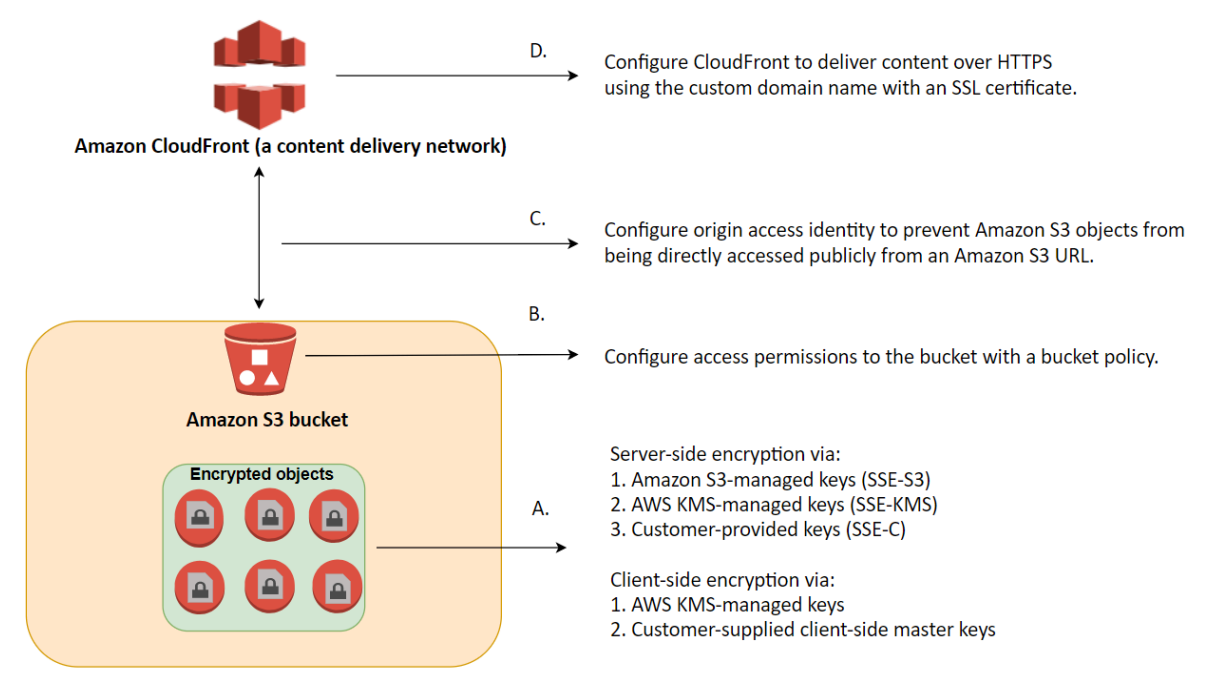
An investment bank is working with an IT team to handle the launch of the new digital wallet system. The applications will run on multiple EBS-backed EC2 instances which will store the logs, transactions, and billing statements of the user in an S3 bucket. Due to tight security and compliance requirements, the IT

team is exploring options on how to safely store sensitive data on the EBS volumes and S3.

Which of the below options should be carried out when storing sensitive data on AWS? (Select TWO.)

- Use AWS Shield and WAF
- Create an EBS Snapshot
- Enable EBS Encryption (Correct)
- Enable Amazon S3 Server-Side or use Client-Side Encryption (Correct)
- Migrate the EC2 instances from the public to private subnet.

Enabling EBS Encryption and **enabling Amazon S3 Server-Side or use Client-Side Encryption** are correct. Amazon EBS encryption offers a simple encryption solution for your EBS volumes without the need to build, maintain, and secure your own key management infrastructure.



In Amazon S3, data protection refers to protecting data while in-transit (as it travels to and from Amazon S3) and at rest (while it is stored on disks in Amazon S3 data centers). You can protect data in transit by using SSL or by using client-side encryption. You have the following options to protect data at rest in Amazon S3.

- Use Server-Side Encryption – You request Amazon S3 to encrypt your object before saving it on disks in its data centers and decrypt it when you download the objects.
- Use Client-Side Encryption – You can encrypt data client-side and upload the encrypted data to Amazon S3. In this case, you manage the encryption process, the encryption keys, and related tools.

Creating an EBS Snapshot is incorrect because this is a backup solution of EBS. It does not provide security of data inside EBS volumes when executed.

Migrating the EC2 instances from the public to private subnet is incorrect because the data you want to secure are those in EBS volumes and S3 buckets. Moving your EC2 instance to a private subnet involves a different matter of security practice, which does not achieve what you want in this scenario.

Using AWS Shield and WAF is incorrect because these protect you from common security threats for your web applications. However, what you are trying to achieve is securing and encrypting your data inside EBS and S3.

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

Check out this Amazon EBS Cheat Sheet:

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

8. QUESTION

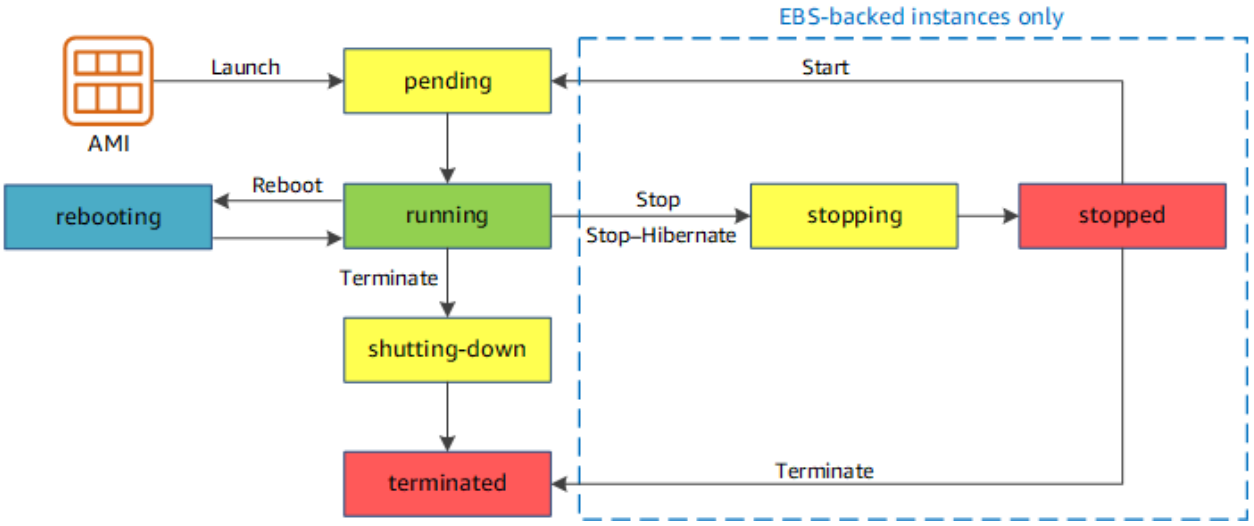
Category: CSAA – Design Secure Architectures

A company is using an On-Demand EC2 instance to host a legacy web application that uses an Amazon Instance Store-Backed AMI. The web application should be decommissioned as soon as possible and hence, you need to terminate the EC2 instance.

When the instance is terminated, what happens to the data on the root volume?

Data is unavailable until the instance is restarted.
Data is automatically saved as an EBS volume.
Data is automatically deleted. (Correct)
Data is automatically saved as an EBS snapshot.

AMIs are categorized as either *backed by Amazon EBS* or *backed by instance store*. The former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. The latter means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.



The data on instance store volumes persist only during the life of the instance which means that if the instance is terminated, the data will be automatically deleted.

Hence, the correct answer is: **Data is automatically deleted.**

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ComponentsAMIs.html

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

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

Topic-Based – EFS (SA-Associate)

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A content management system (CMS) is hosted on a fleet of auto-scaled, On-Demand EC2 instances that use Amazon Aurora as its database. Currently, the system stores the file documents that the users upload in one of the attached EBS Volumes. Your manager noticed that the system performance is quite slow and he has instructed you to improve the architecture of the system.

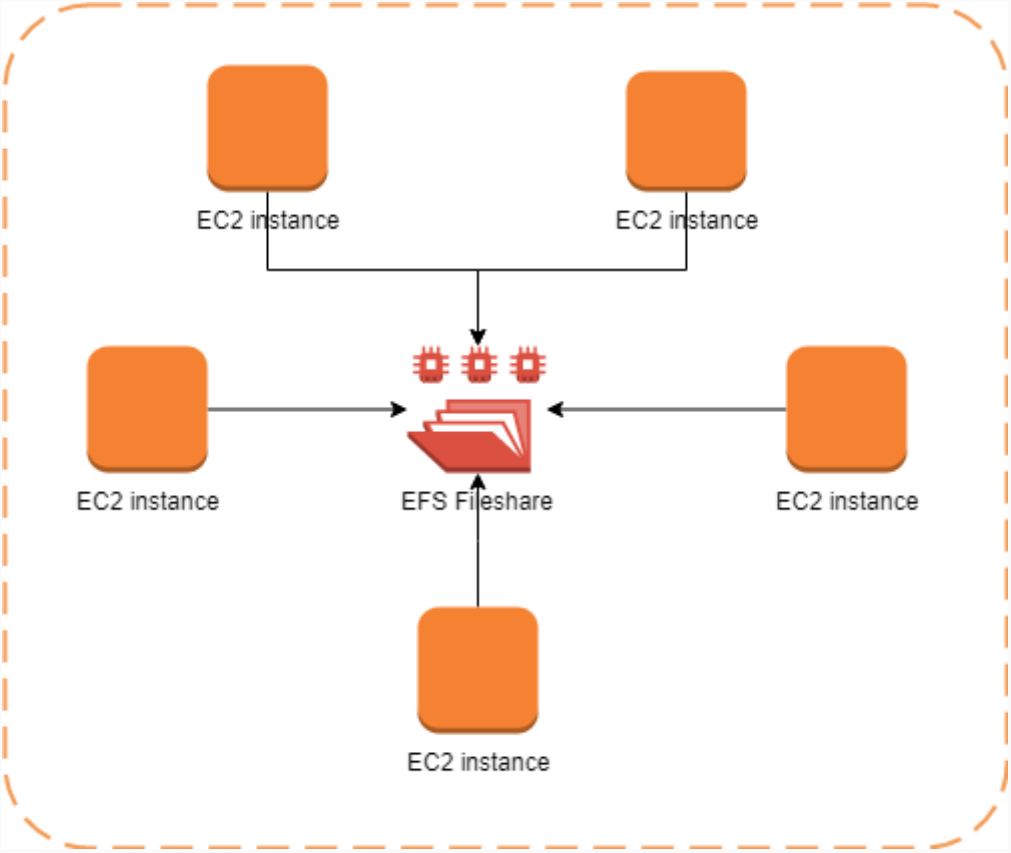
In this scenario, what will you do to implement a scalable, high-available POSIX-compliant shared file system?

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

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. When mounted on Amazon EC2 instances, an Amazon EFS file system provides a standard file system interface and file system access semantics, allowing you to seamlessly integrate Amazon EFS with your existing applications and tools. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, allowing Amazon EFS to provide a common data source for workloads and applications running on more than one Amazon EC2 instance.

This particular scenario tests your understanding of EBS, EFS, and S3. In this scenario, there is a fleet of On-Demand EC2 instances that store file documents from the users to one of the attached EBS Volumes. The system performance is quite slow because the architecture doesn't provide the EC2 instances parallel shared access to the file documents.

Although an EBS Volume can be attached to multiple EC2 instances, you can only do so on instances within an availability zone. What we need is high-available storage that can span multiple availability zones. Take note as well that the type of storage needed here is “file storage” which means that **S3** is not the best service to use because it is mainly used for “object storage”, and S3 does not provide the notion of “folders” too. This is why **using EFS** is the correct answer.



Upgrading your existing EBS volumes to Provisioned IOPS SSD Volumes is incorrect because an EBS volume is a storage area network (SAN) storage and not a POSIX-compliant shared file system. You have to use EFS instead.

Using ElastiCache is incorrect because this is an in-memory data store that improves the performance of your applications, which is not what you need since it is not a file storage.

Reference:

https://aws.amazon.com/efs/

Check out this Amazon EFS Cheat Sheet:

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

Check out this Amazon S3 vs EBS vs EFS Cheat Sheet:

https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/

2. QUESTION

Category: CSAA – Design High-Performing Architectures

A leading e-commerce company is in need of a storage solution that can be simultaneously accessed by 1000 Linux servers in multiple availability zones. The servers are hosted in EC2 instances that use a hierarchical directory structure via the NFSv4 protocol. The service should be able to handle the rapidly changing data at scale while still maintaining high performance. It should also be highly durable and highly available whenever the servers will pull data from it, with little need for management.

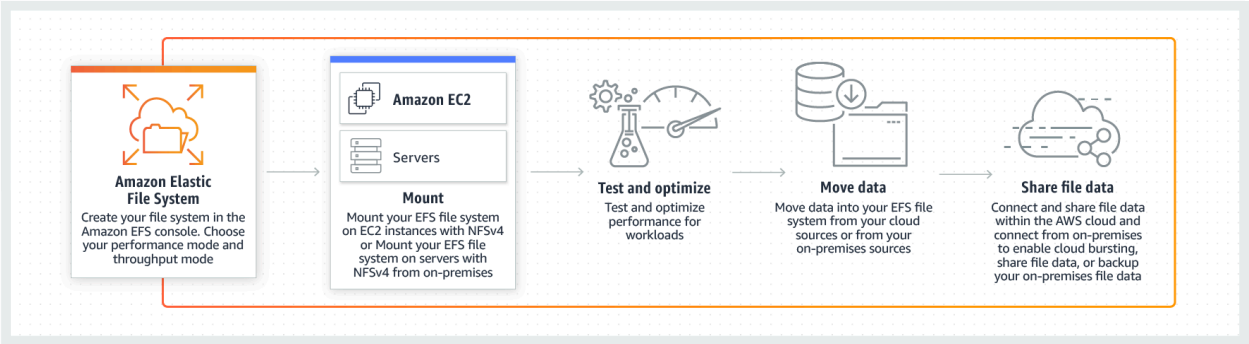
As the Solutions Architect, which of the following services is the most cost-effective choice that you should use to meet the above requirement?

Amazon S3
Amazon EFS (Correct)
Amazon EBS
Amazon FSx for Windows File Server

Amazon Web Services (AWS) offers cloud storage services to support a wide range of storage workloads such as EFS, S3, and EBS. You have to understand when you should use Amazon EFS, Amazon S3, and Amazon Elastic Block Store (EBS) based on the specific workloads. In this scenario, the keywords are *rapidly changing data* and 1000 Linux servers.

Amazon EFS is a file storage service for use with Amazon EC2. Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file

locking), and concurrently-accessible storage for up to thousands of Amazon EC2 instances. EFS provides the same level of high availability and high scalability like S3 however, this service is more suitable for scenarios where it is required to have a POSIX-compatible file system or if you are storing rapidly changing data.



Data that must be updated very frequently might be better served by storage solutions that take into account read and write latencies, such as Amazon EBS volumes, Amazon RDS, Amazon DynamoDB, Amazon EFS, or relational databases running on Amazon EC2.

Amazon EBS is a block-level storage service for use with Amazon EC2. Amazon EBS can deliver performance for workloads that require the lowest-latency access to data from a single EC2 instance.

Amazon S3 is an object storage service. Amazon S3 makes data available through an Internet API that can be accessed anywhere.

In this scenario, Amazon EFS is the best answer. As stated above, Amazon EFS provides a file system interface, file system access semantics (such as strong consistency and file locking), and concurrently-accessible storage for up to thousands of Amazon EC2 instances. EFS provides the performance, durability, high availability, and storage capacity needed by the 1000 Linux servers in the scenario.

Amazon S3 is incorrect. Although this provides the same level of high availability and high scalability like EFS, this service is not suitable for storing data that is rapidly changing, just as mentioned in the above explanation. It is still more effective to use EFS as it offers strong consistency and file locking, which the S3 service lacks.

Amazon EBS is incorrect because an EBS Volume cannot be shared by multiple instances.

Amazon FSx for Windows File Server is incorrect. Although this storage service can be connected to multiple EC2 instances, it is still constrained to Windows OS only. Take note that the scenario mentions that the Amazon EC2 instances are Linux servers and not Windows machines.

References:

https://docs.aws.amazon.com/efs/latest/ug/how-it-works.html

https://aws.amazon.com/efs/features/

https://d1.awsstatic.com/whitepapers/AWS%20Storage%20Services%20Whitepaper-v9.pdf#page=9

Check out this Amazon EFS Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design High-Performing Architectures

A multinational company has been building its new data analytics platform with high-performance computing workloads (HPC) which requires a scalable, POSIX-compliant storage service. The data need to be stored redundantly across multiple AZs and allows concurrent connections from thousands of EC2 instances hosted on multiple Availability Zones.

Which of the following AWS storage service is the most suitable one to use in this scenario?

Amazon ElastiCache
Amazon S3
Amazon EBS Volumes
Amazon Elastic File System (Correct)

In this question, you should take note of this phrase: “allows concurrent connections from multiple EC2 instances”. There are various AWS storage options that you can choose but whenever these criteria show up, always consider using EFS instead of using EBS Volumes which is mainly used as a “block” storage and can only have one connection to one EC2 instance at a time.

Amazon EFS is a fully-managed service that makes it easy to set up and scale file storage in the Amazon Cloud. With a few clicks in the AWS Management Console, you can create file systems that are accessible to Amazon EC2 instances via a file system interface (using

standard operating system file I/O APIs) and supports full file system access semantics (such as strong consistency and file locking).

Amazon EFS file systems can automatically scale from gigabytes to petabytes of data without needing to provision storage. Tens, hundreds, or even thousands of Amazon EC2 instances can access an Amazon EFS file system at the same time, and Amazon EFS provides consistent performance to each Amazon EC2 instance. Amazon EFS is designed to be highly durable and highly available.

References:

https://docs.aws.amazon.com/efs/latest/ug/performance.html

https://aws.amazon.com/efs/faq/

Check out this Amazon EFS Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A Solutions Architect is implementing a new High-Performance Computing (HPC) system in AWS that involves orchestrating several Amazon Elastic Container Service (Amazon ECS) tasks with an EC2 launch type that is part of an Amazon ECS cluster. The system will be frequently accessed by users around the globe and it is expected that there would be hundreds of ECS tasks running most of the time. The Architect must ensure that its storage system is optimized for high-frequency read and write operations. The output data of each ECS task is around 10 MB but the obsolete data will eventually be archived and deleted so the total storage size won't exceed 10 TB.

Which of the following is the MOST suitable solution that the Architect should recommend?

Launch an Amazon Elastic File System (Amazon EFS) with Provisioned Throughput mode and set the performance mode to `Max I/O`. Configure the EFS file system as the container mount point in the ECS task definition of the Amazon ECS cluster. (Correct)

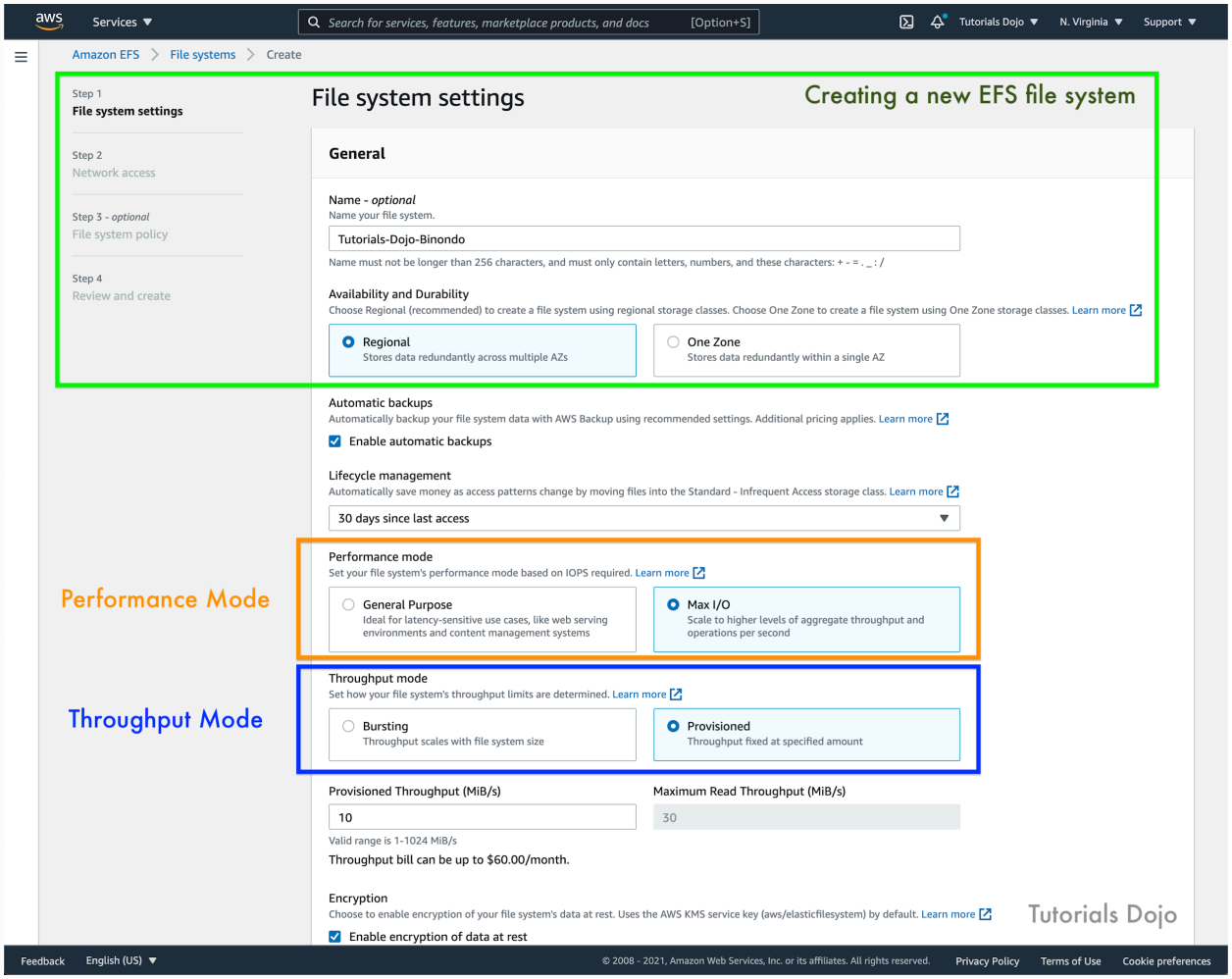
Set up an SMB file share by creating an Amazon FSx File Gateway in Storage Gateway. Set the file share as the container mount point in the ECS task definition of the Amazon ECS cluster.

Launch an Amazon Elastic File System (Amazon EFS) file system with Bursting Throughput mode and set the performance mode to `General Purpose`. Configure the EFS file system as the container mount point in the ECS task definition of the Amazon ECS cluster.

Launch an Amazon DynamoDB table with Amazon DynamoDB Accelerator (DAX) and DynamoDB Streams enabled. Configure the table to be accessible by all Amazon ECS cluster instances. Set the DynamoDB table as the container mount point in the ECS task definition of the Amazon ECS cluster.

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with your Amazon ECS tasks. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files. Your applications can have the storage they need when they need it.

You can use Amazon EFS file systems with Amazon ECS to access file system data across your fleet of Amazon ECS tasks. That way, your tasks have access to the same persistent storage, no matter the infrastructure or container instance on which they land. When you reference your Amazon EFS file system and container mount point in your Amazon ECS task definition, Amazon ECS takes care of mounting the file system in your container.



To support a wide variety of cloud storage workloads, Amazon EFS offers two performance modes:

– General Purpose mode

– Max I/O mode.

You choose a file system's performance mode when you create it, and it cannot be changed. The two performance modes have no additional costs, so your Amazon EFS file system is billed and metered the same, regardless of your performance mode.

There are two throughput modes to choose from for your file system:

– Bursting Throughput

– Provisioned Throughput

With Bursting Throughput mode, a file system's throughput scales as the amount of data stored in the EFS Standard or One Zone storage class grows. File-based workloads are typically spiky, driving high levels of throughput for short periods of time, and low levels of

throughput the rest of the time. To accommodate this, Amazon EFS is designed to burst to high throughput levels for periods of time.

Provisioned Throughput mode is available for applications with high throughput to storage (MiB/s per TiB) ratios, or with requirements greater than those allowed by the Bursting Throughput mode. For example, say you're using Amazon EFS for development tools, web serving, or content management applications where the amount of data in your file system is low relative to throughput demands. Your file system can now get the high levels of throughput your applications require without having to pad your file system.

In the scenario, the file system will be frequently accessed by users around the globe so it is expected that there would be hundreds of ECS tasks running most of the time. The Architect must ensure that its storage system is optimized for high-frequency read and write operations.

Hence, the correct answer is: **Launch an Amazon Elastic File System (Amazon EFS) with Provisioned Throughput mode and set the performance mode to `Max I/O`. Configure the EFS file system as the container mount point in the ECS task definition of the Amazon ECS cluster.**

The option that says: **Set up an SMB file share by creating an Amazon FSx File Gateway in Storage Gateway. Set the file share as the container mount point in the ECS task definition of the Amazon ECS cluster** is incorrect. Although you can use an Amazon FSx for Windows File Server in this situation, it is not appropriate to use this since the application is not connected to an on-premises data center. Take note that the AWS Storage Gateway service is primarily used to integrate your existing on-premises storage to AWS.

The option that says: **Launch an Amazon Elastic File System (Amazon EFS) file system with Bursting Throughput mode and set the performance mode to `General Purpose`. Configure the EFS file system as the container mount point in the ECS task definition of the Amazon ECS cluster** is incorrect because using Bursting Throughput mode won't be able to sustain the constant demand of the global application. Remember that the application will be frequently accessed by users around the world and there are hundreds of ECS tasks running most of the time.

The option that says: **Launch an Amazon DynamoDB table with Amazon DynamoDB Accelerator (DAX) and DynamoDB Streams enabled. Configure the table to be accessible by all Amazon ECS cluster instances. Set the DynamoDB table as the container mount point in the ECS task definition of the Amazon ECS cluster** is incorrect because you cannot directly set a DynamoDB table as a container mount point. In the first place, DynamoDB is a database and not a file system which means that it can't be "mounted" to a server.

References:

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/tutorial-efs-volumes.html

https://docs.aws.amazon.com/efs/latest/ug/performance.html

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/tutorial-wfsx-volumes.html

Check out this Amazon EFS Cheat Sheet:

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

Topic-Based – ELB (SA-Associate)

1. QUESTION

Category: CSAA – Design Secure Architectures

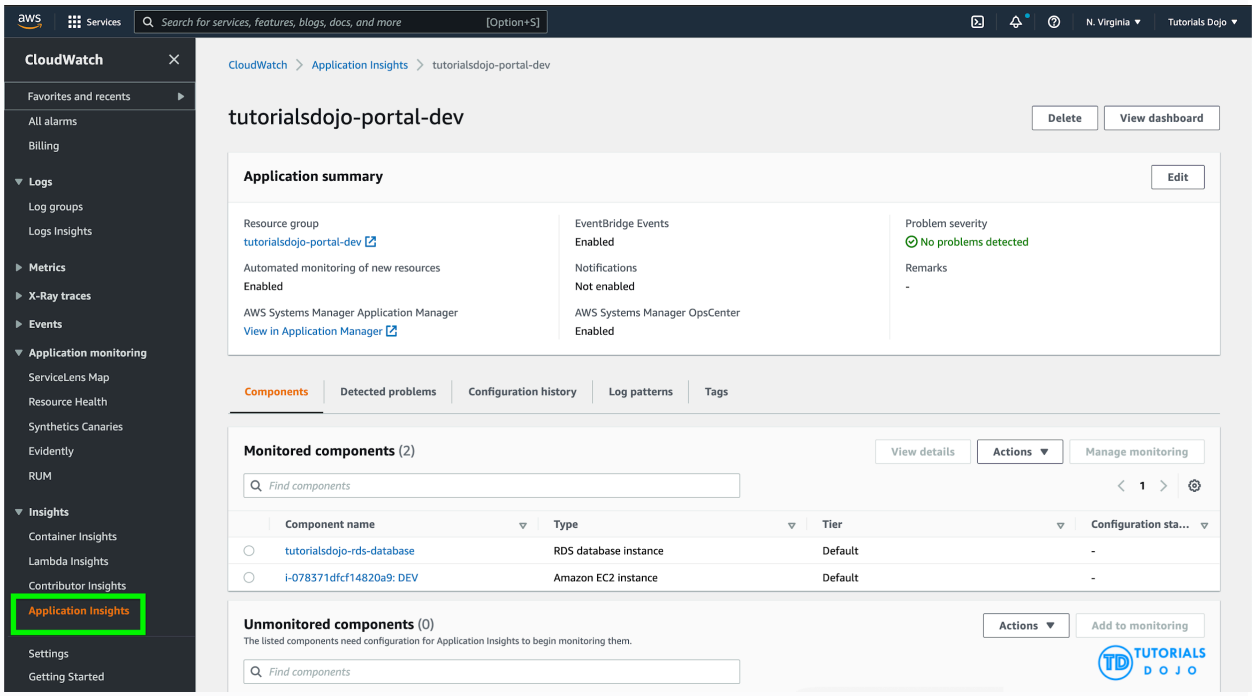
A social media company needs to capture the detailed information of all HTTP requests that went through their public-facing Application Load Balancer every five minutes. The client's IP address and network latencies must also be tracked. They want to use this data for analyzing traffic patterns and for troubleshooting their Docker applications orchestrated by the Amazon ECS Anywhere service.

Which of the following options meets the customer requirements with the LEAST amount of overhead?

Install and run the AWS X-Ray daemon on the Amazon ECS cluster. Use the Amazon CloudWatch ServiceLens to analyze the traffic that goes through the application.
Enable access logs on the Application Load Balancer. Integrate the Amazon ECS cluster with Amazon CloudWatch Application Insights to analyze traffic patterns and simplify troubleshooting. (Correct)
Enable AWS CloudTrail for their Application Load Balancer. Use the AWS CloudTrail Lake to analyze and troubleshoot the application traffic.

Integrate Amazon EventBridge (Amazon CloudWatch Events) metrics on the Application Load Balancer to capture the client IP address. Use Amazon CloudWatch Container Insights to analyze traffic patterns.

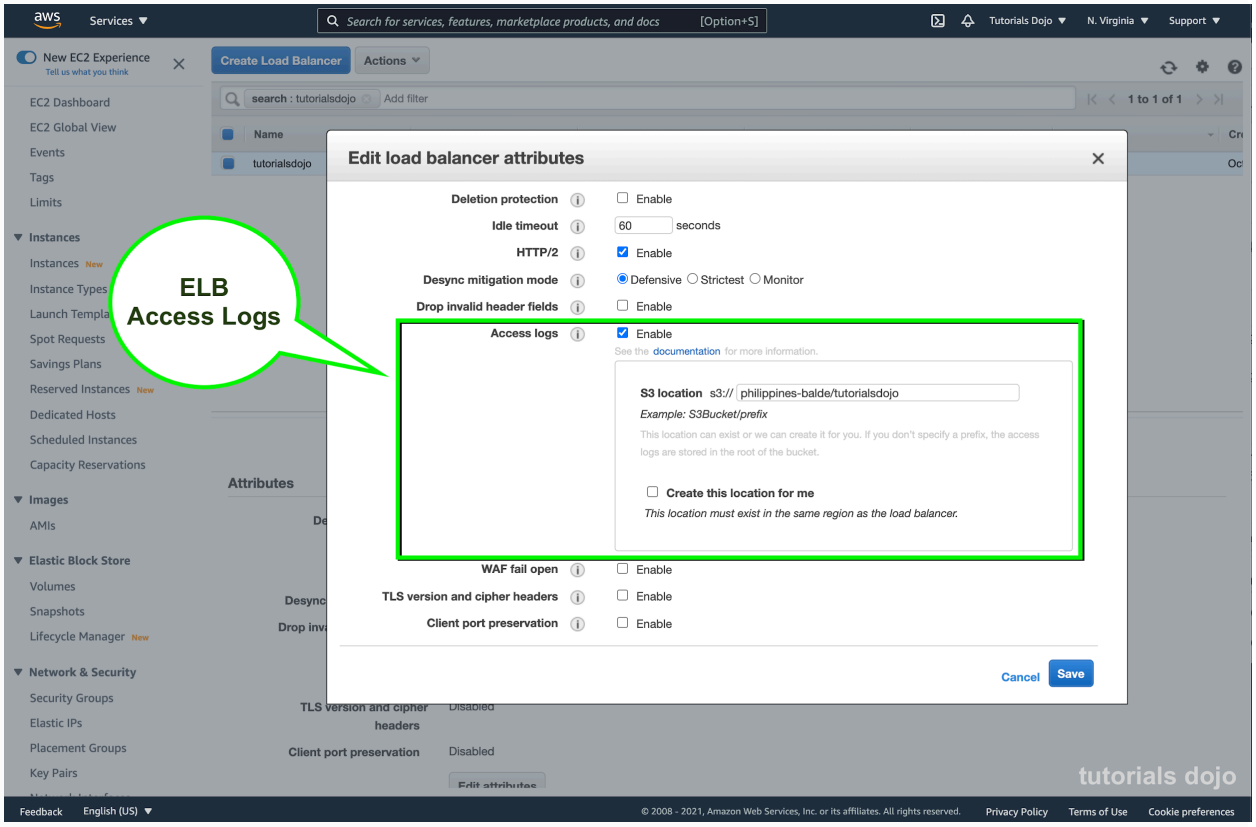
Amazon CloudWatch Application Insights facilitates observability for your applications and underlying AWS resources. It helps you set up the best monitors for your application resources to continuously analyze data for signs of problems with your applications. Application Insights, which is powered by SageMaker and other AWS technologies, provides automated dashboards that show potential problems with monitored applications, which help you to quickly isolate ongoing issues with your applications and infrastructure. The enhanced visibility into the health of your applications that Application Insights provides helps reduce the “mean time to repair” (MTTR) to troubleshoot your application issues.



When you add your applications to Amazon CloudWatch Application Insights, it scans the resources in the applications and recommends and configures metrics and logs on CloudWatch for application components. Example application components include SQL Server backend databases and Microsoft IIS/Web tiers. Application Insights analyzes metric patterns using historical data to detect anomalies and continuously detects errors and exceptions from your application, operating system, and infrastructure logs. It correlates these observations using a combination of classification algorithms and built-in rules. Then, it automatically creates dashboards that show the relevant observations and problem severity information to help you prioritize your actions.

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the

request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues.



Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Load Balancing captures the logs and stores them in the Amazon S3 bucket that you specify as compressed files. You can disable access logging at any time.

Hence, the correct answer is: **Enable access logs on the Application Load Balancer. Integrate the Amazon ECS cluster with Amazon CloudWatch Application Insights to analyze traffic patterns and simplify troubleshooting.**

The option that says: **Enable AWS CloudTrail for their Application Load Balancer. Use the AWS CloudTrail Lake to analyze and troubleshoot the application traffic** is incorrect because AWS CloudTrail is primarily used to monitor and record the account activity across your AWS resources and not your web applications. You cannot use CloudTrail to capture the detailed information of all HTTP requests that go through your public-facing Application Load Balancer (ALB). CloudTrail can only track the resource changes made to your ALB, but not the actual IP traffic that goes through it. For this use case, you have to enable the access logs feature instead. In addition, the AWS CloudTrail Lake feature is more suitable for running SQL-based queries on your API events and not for analyzing application traffic.

The option that says: **Install and run the AWS X-Ray daemon on the Amazon ECS cluster. Use the Amazon CloudWatch ServiceLens to analyze the traffic that goes through the application** is incorrect. Although this solution is possible, this won't track the client's IP address since the access log feature in the ALB is not enabled. Take note that the scenario explicitly mentioned that the client's IP address and network latencies must also be tracked.

The option that says: **Integrate Amazon EventBridge (Amazon CloudWatch Events) metrics on the Application Load Balancer to capture the client IP address. Use Amazon CloudWatch Container Insights to analyze traffic patterns** is incorrect because Amazon EventBridge doesn't track the actual traffic to your ALB. It is the Amazon CloudWatch service that monitors the changes to your ALB itself and the actual IP traffic that it distributes to the target groups. The primary function of CloudWatch Container Insights is to collect, aggregate, and summarize metrics and logs from your containerized applications and microservices.

References:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html

http://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-monitoring.html

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

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

Application Load Balancer vs Network Load Balancer vs Gateway Load Balancer:

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

2. QUESTION

Category: CSAA – Design High-Performing Architectures

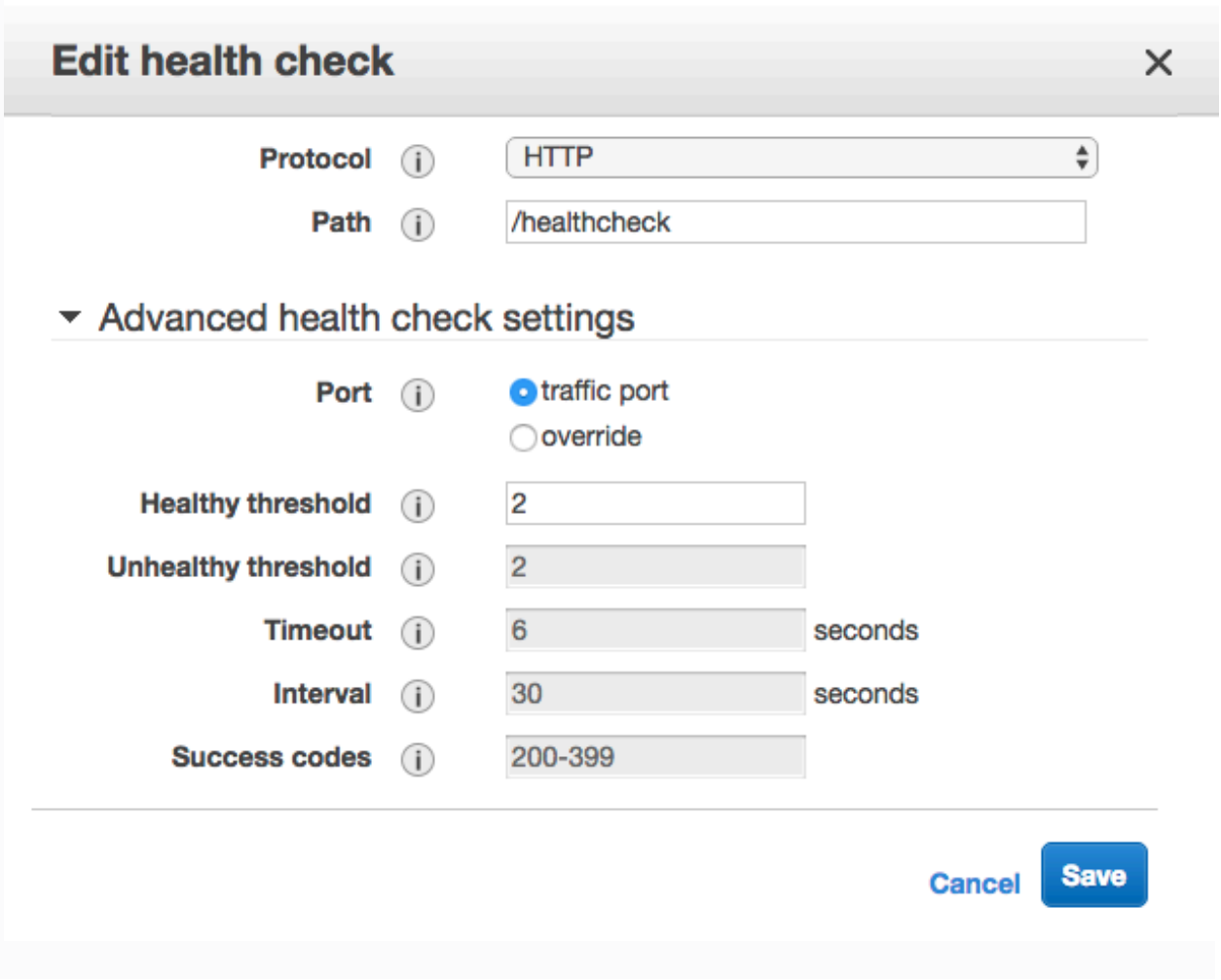
A company plans to design a highly available architecture in AWS. They have two target groups with three EC2 instances each, which are added to an Application Load Balancer. In the security group of the EC2 instance, you have verified that port 80 for

HTTP is allowed. However, the instances are still showing out of service from the load balancer.

What could be the root cause of this issue?

The wrong instance type was used for the EC2 instance.
The wrong subnet was used in your VPC
The health check configuration is not properly defined. (Correct)
The instances are using the wrong AMI.

Since the security group is properly configured, the issue may be caused by a wrong **health check configuration** in the Target Group.



Your Application Load Balancer periodically sends requests to its registered targets to test their status. These tests are called *health checks*. Each load balancer node routes requests only to the healthy targets in the enabled Availability Zones for the load balancer. Each load balancer node checks the health of each target, using the health check settings for the target group with which the target is registered. After your target is registered, it must pass one health check to be considered healthy. After each health check is completed, the load balancer node closes the connection that was established for the health check.

Hence, the options such as **using the wrong AMI**, instance type, or subnet, are less likely to be the cause of the problem, as they wouldn't directly affect the health check status reported by the load balancer.

Reference:

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html

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

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

ELB Health Checks vs Route 53 Health Checks For Target Health Monitoring:

https://tutorialsdojo.com/elb-health-checks-vs-route-53-health-checks-for-target-health-monitoring/

3. QUESTION

Category: CSAA – Design Resilient Architectures

A company plans to host a movie streaming app in AWS. The chief information officer (CIO) wants to ensure that the application is highly available and scalable. The application is deployed to an Auto Scaling group of EC2 instances on multiple AZs. A load balancer must be configured to distribute incoming requests evenly to all EC2 instances across multiple Availability Zones.

Which of the following features should the Solutions Architect use to satisfy these criteria?

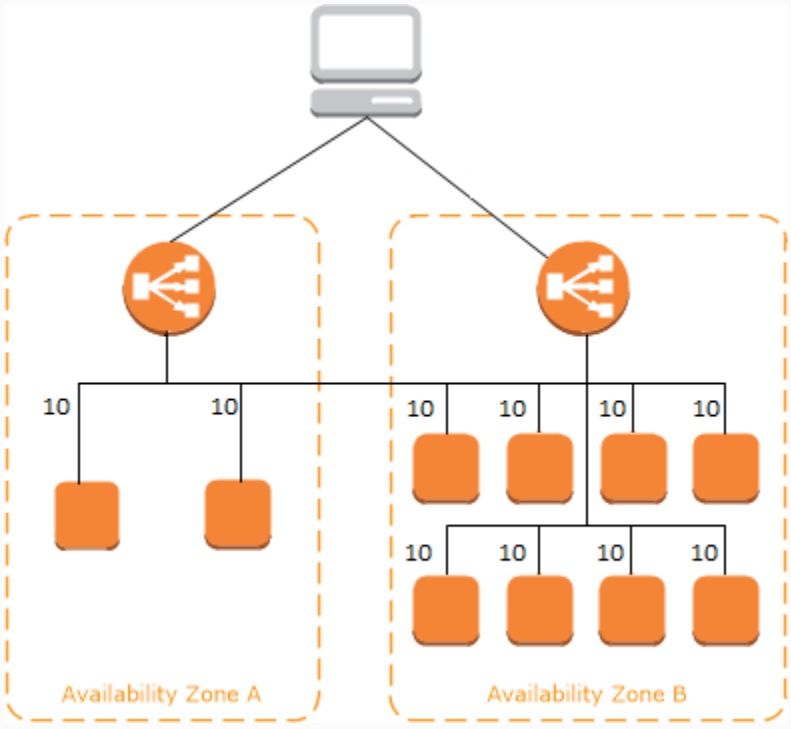
AWS Direct Connect SiteLink

Amazon VPC IP Address Manager (IPAM)
Path-based Routing
Cross-zone load balancing (Correct)

The nodes for your load balancer distribute requests from clients to registered targets. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone.

The following diagrams demonstrate the effect of cross-zone load balancing. There are two enabled Availability Zones, with two targets in Availability Zone A and eight targets in Availability Zone B. Clients send requests, and Amazon Route 53 responds to each request with the IP address of one of the load balancer nodes. This distributes traffic such that each load balancer node receives 50% of the traffic from the clients. Each load balancer node distributes its share of the traffic across the registered targets in its scope.

If cross-zone load balancing is enabled, each of the 10 targets receives 10% of the traffic. This is because each load balancer node can route 50% of the client traffic to all 10 targets.

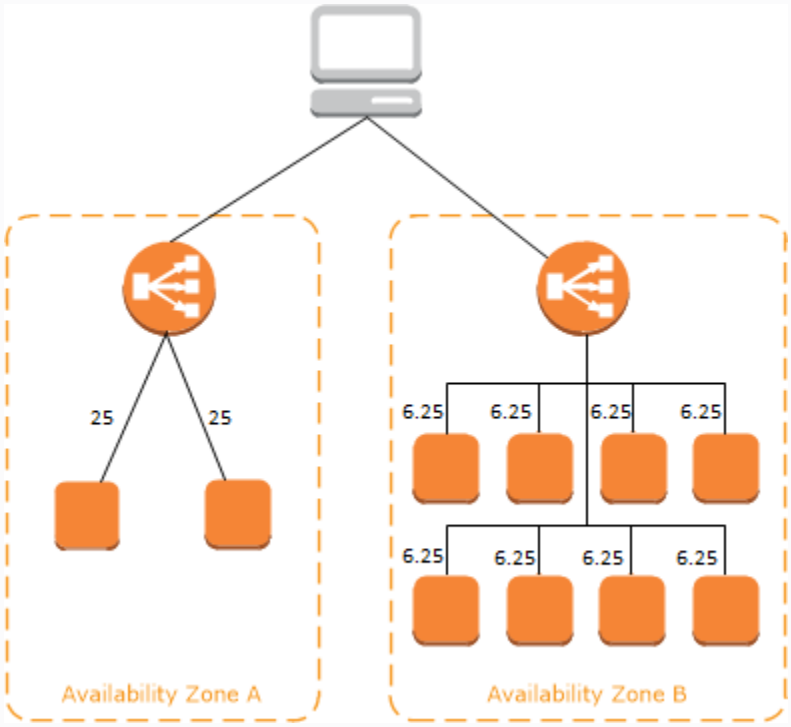


If cross-zone load balancing is disabled:

- Each of the two targets in Availability Zone A receives 25% of the traffic.

- Each of the eight targets in Availability Zone B receives 6.25% of the traffic.

This is because each load balancer node can route 50% of the client traffic only to targets in its Availability Zone.



With Application Load Balancers, cross-zone load balancing is always enabled.

With Network Load Balancers and Gateway Load Balancers, cross-zone load balancing is disabled by default. After you create the load balancer, you can enable or disable cross-zone load balancing at any time.

Hence, the right answer is to enable **cross-zone load balancing.**

Amazon VPC IP Address Manager (IPAM) is incorrect because this is merely a feature in Amazon VPC that provides network administrators with an automated IP management workflow. It does not enable your load balancers to distribute incoming requests evenly to all EC2 instances across multiple Availability Zones.

Path-based Routing is incorrect because this feature is based on the paths that are in the URL of the request. It automatically routes traffic to a particular target group based on the request URL. This feature will not set each of the load balancer nodes to distribute traffic across the registered targets in all enabled Availability Zones.

AWS Direct Connect SiteLink is incorrect because this is a feature of AWS Direct Connect connection and not of Amazon Elastic Load Balancing. The AWS Direct Connect SiteLink

feature simply lets you create connections between your on-premises networks through the AWS global network backbone.

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html

https://aws.amazon.com/elasticloadbalancing/features

https://aws.amazon.com/blogs/aws/network-address-management-and-auditing-at-scale-with-amazon-vpc-ip-address-manager/

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

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

4. QUESTION

Category: CSAA – Design Resilient Architectures

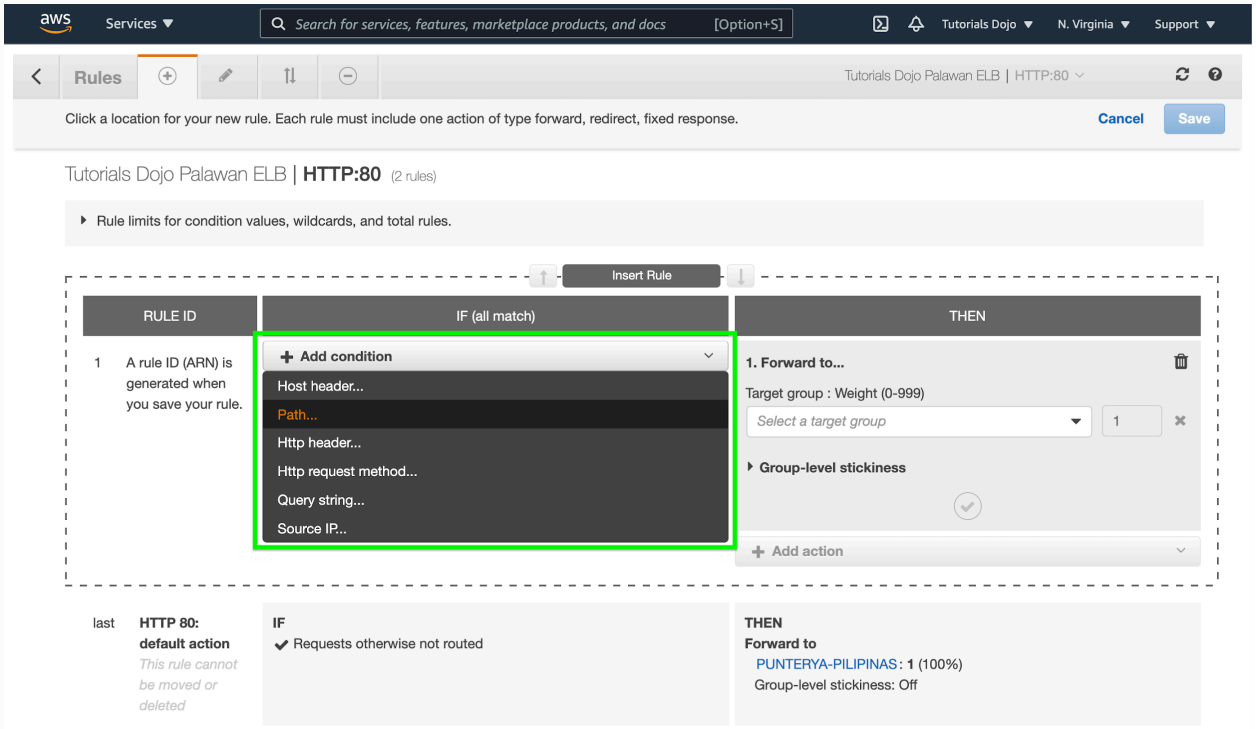
A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture consisting of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used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

Which configuration will satisfy the given requirement?

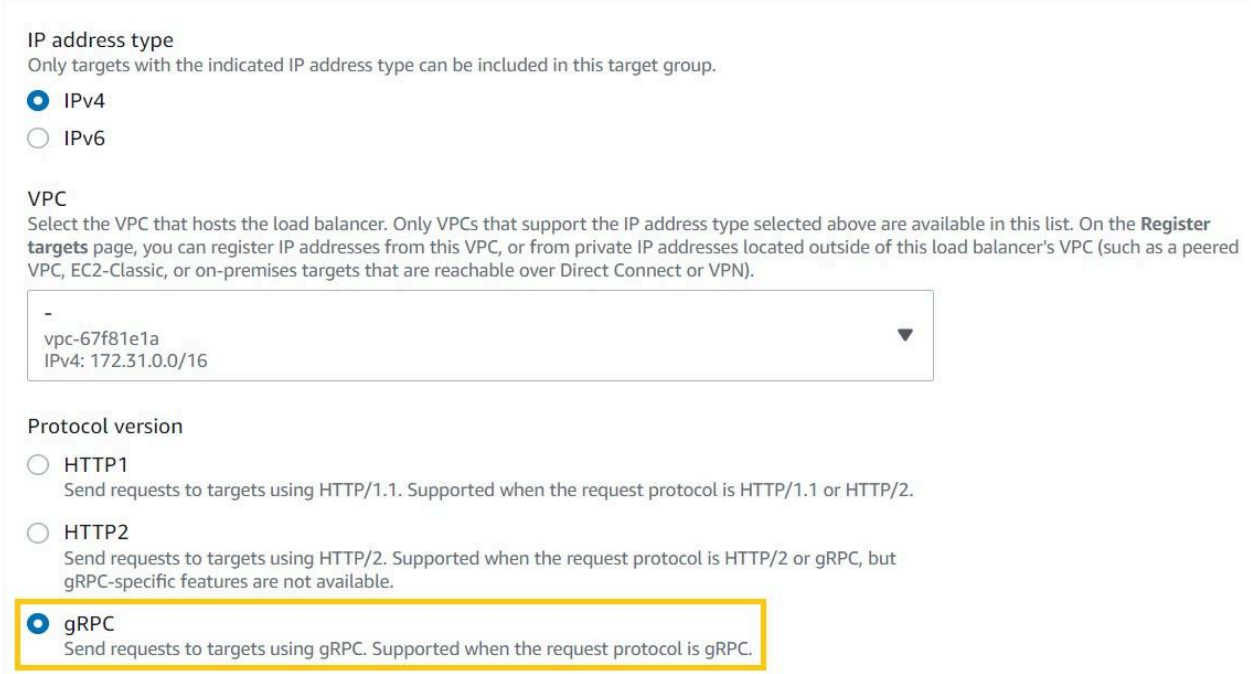
Configure an Application Load Balancer in front of the auto-scaling group. Select gRPC as the protocol version. (Correct)
Configure a Network Load Balancer in front of the auto-scaling group. Create an AWS Global Accelerator accelerator and set the load balancer as an endpoint.
Configure a Network Load Balancer in front of the auto-scaling group. Use a UDP listener for routing.

Configure a Gateway Load Balancer in front of the auto-scaling group. Ensure that the IP Listener Routing uses the GENEVE protocol on port 6081 to allow gRPC response traffic.

Application Load Balancer operates at the request level (layer 7), routing traffic to targets (EC2 instances, containers, IP addresses, and Lambda functions) based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.



If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request such as Host field, Path URL, HTTP header, HTTP method, Query string, or Source IP address.



ALBs can also route and load balance gRPC traffic between microservices or between gRPC-enabled clients and services. This will allow customers to seamlessly introduce gRPC traffic management in their architectures without changing any of the underlying infrastructure on their clients or services.

Therefore, the correct answer is: **Configure an Application Load Balancer in front of the auto-scaling group. Select gRPC as the protocol version.**

The option that says: **Configure a Network Load Balancer in front of the auto-scaling group. Use a UDP listener for routing** is incorrect. Network Load Balancers do not support gRPC.

The option that says: **Configure a Gateway Load Balancer in front of the auto-scaling group. Ensure that the IP Listener Routing uses the GENEVE protocol on port 6081 to allow gRPC response traffic** is incorrect. A Gateway Load Balancer operates as a Layer 3 Gateway and a Layer 4 Load Balancing service. Do take note that the gRPC protocol is at Layer 7 of the OSI Model so this service is not appropriate for this scenario.

The option that says: **Configure a Network Load Balancer in front of the auto-scaling group. Create an AWS Global Accelerator accelerator and set the load balancer as an endpoint** is incorrect. AWS Global Accelerator simply optimizes application performance by routing user traffic to the congestion-free, redundant AWS global network instead of the public internet.

References:

https://aws.amazon.com/elasticloadbalancing/features

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

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

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

Application Load Balancer vs Network Load Balancer vs Gateway Load Balancer:

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

5. QUESTION

Category: CSAA – Design High-Performing Architectures

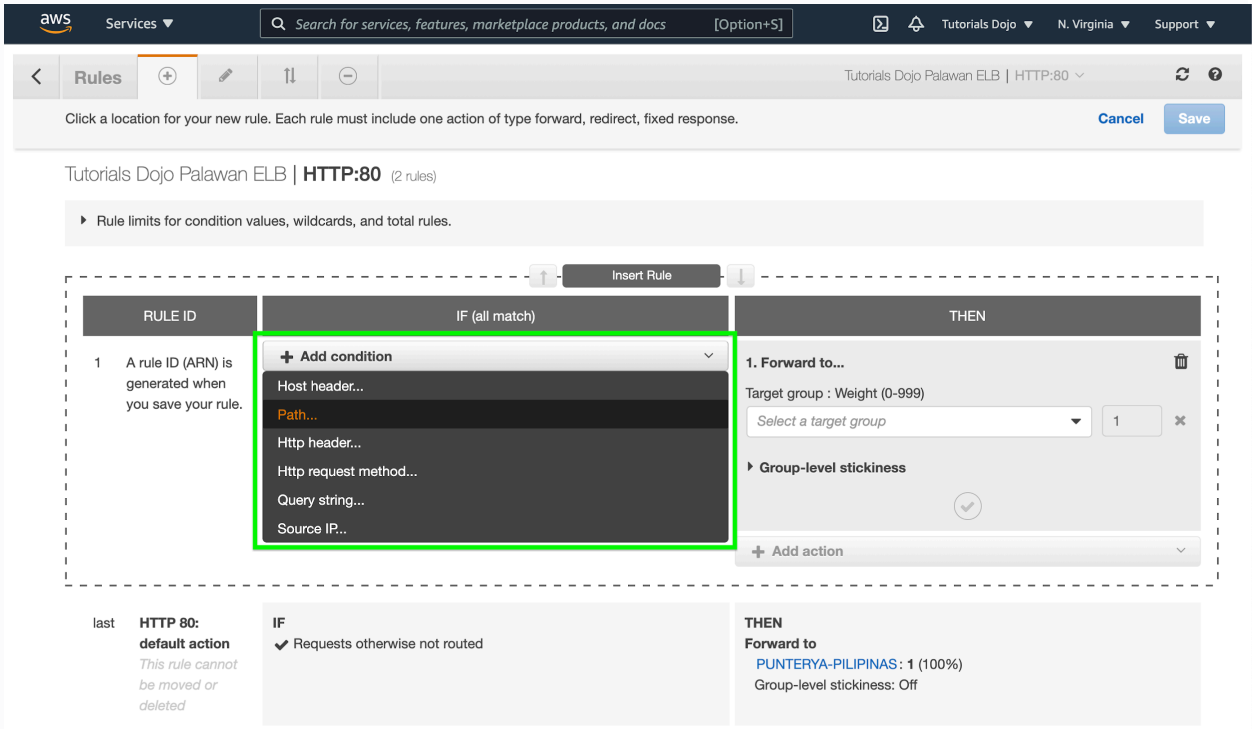
A fast food company is using AWS to host their online ordering system which uses an Auto Scaling group of EC2 instances deployed across multiple Availability Zones with an Application Load Balancer in front. To better handle the incoming traffic from various digital devices, you are planning to implement a new routing system where requests which have a URL of <server>/api/android are forwarded to one specific target group named “Android-Target-Group”. Conversely, requests which have a URL of <server>/api/ios are forwarded to another separate target group named “iOS-Target-Group”.

How can you implement this change in AWS?

Use host conditions to define rules that forward requests to different target groups based on the hostname in the host header. This enables you to support multiple domains using a single load balancer.
Replace your ALB with a Gateway Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request.
Use path conditions to define rules that forward requests to different target groups based on the URL in the request. (Correct)

> Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request.

If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request such as Host field, Path URL, HTTP header, HTTP method, Query string, or Source IP address. Path-based routing allows you to route a client request based on the URL path of the HTTP header. Each path condition has one path pattern. If the URL in a request matches the path pattern in a listener rule exactly, the request is routed using that rule.



A path pattern is case-sensitive, can be up to 128 characters in length, and can contain any of the following characters. You can include up to three wildcard characters.

- A–Z, a–z, 0–9
- _ – . $ / ~ ” ‘ @ : +
- & (using &)
- * (matches 0 or more characters)
- ? (matches exactly 1 character)

Example path patterns

- `/img/*`

- `/js/*`

You can use path conditions to define rules that forward requests to different target groups based on the URL in the request (also known as path-based routing). This type of routing is the most appropriate solution for this scenario hence, the correct answer is: **Use path conditions to define rules that forward requests to different target groups based on the URL in the request.**

The option that says: **Use host conditions to define rules that forward requests to different target groups based on the hostname in the host header. This enables you to support multiple domains using a single load balancer** is incorrect because host-based routing defines rules that forward requests to different target groups based on the hostname in the host header instead of the URL, which is what is needed in this scenario.

The option that says: **Replace your ALB with a Gateway Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request** is incorrect because a Gateway Load Balancer does not support path-based routing. You must use an Application Load Balancer.

The option that says: **Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request** is incorrect because a Network Load Balancer is used for applications that need extreme network performance and static IP. It also does not support path-based routing which is what is needed in this scenario. Furthermore, the statement mentions host-based routing even though the scenario is about path-based routing.

References:

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

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html#path-conditions

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

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

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

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

6. QUESTION

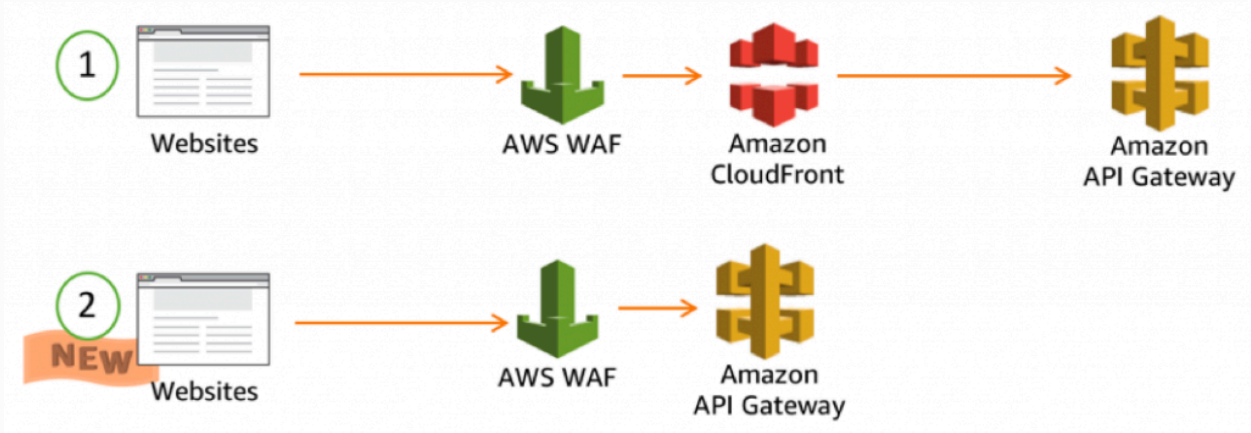
Category: CSAA – Design Secure Architectures

A company is hosting its web application in an Auto Scaling group of EC2 instances behind an Application Load Balancer. Recently, the Solutions Architect identified a series of SQL injection attempts and cross-site scripting attacks to the application, which had adversely affected their production data.

Which of the following should the Architect implement to mitigate this kind of attack?

- Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer. (Correct)
- Using AWS Firewall Manager, set up security rules that block SQL injection and cross-site scripting attacks. Associate the rules to the Application Load Balancer.
- Use Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application.
- Block all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List.

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.



At the simplest level, AWS WAF lets you choose one of the following behaviors:

Allow all requests except the ones that you specify – This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.

Block all requests except the ones that you specify – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.

Count the requests that match the properties that you specify – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

Hence, the correct answer in this scenario is: Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.

Using Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application is incorrect because Amazon GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

Using AWS Firewall Manager to set up security rules that block SQL injection and cross-site scripting attacks, then associating the rules to the Application Load Balancer is incorrect because AWS Firewall Manager just simplifies your AWS WAF and AWS Shield Advanced administration and maintenance tasks across multiple accounts and resources.

Blocking all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List is incorrect because this is an optional

layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACLs are not effective in blocking SQL injection and cross-site scripting attacks

References:

https://aws.amazon.com/waf/

https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html

Check out this AWS WAF Cheat Sheet:

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

7. QUESTION

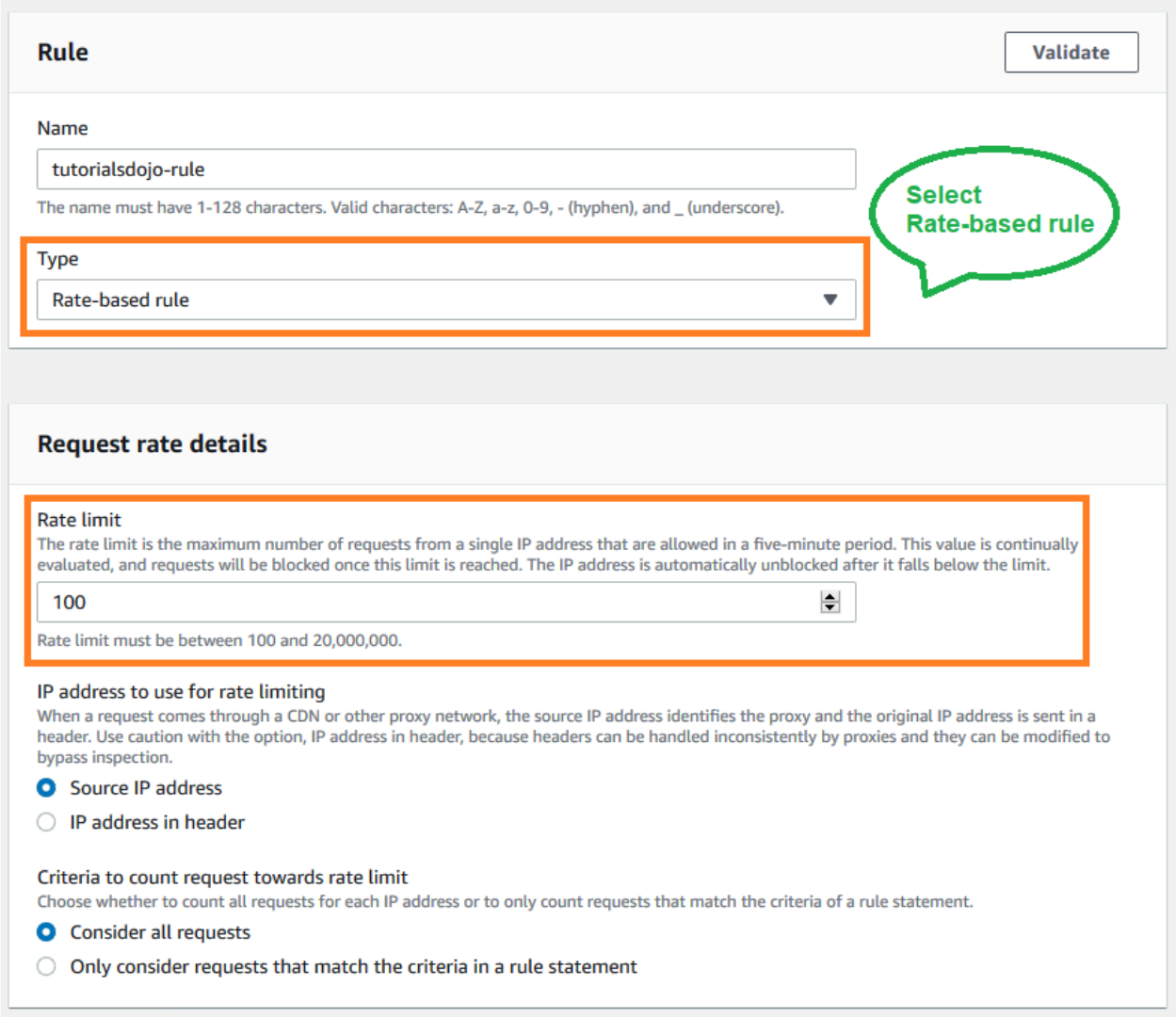
Category: CSAA – Design Secure Architectures

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

8. QUESTION

Category: CSAA – Design Secure Architectures

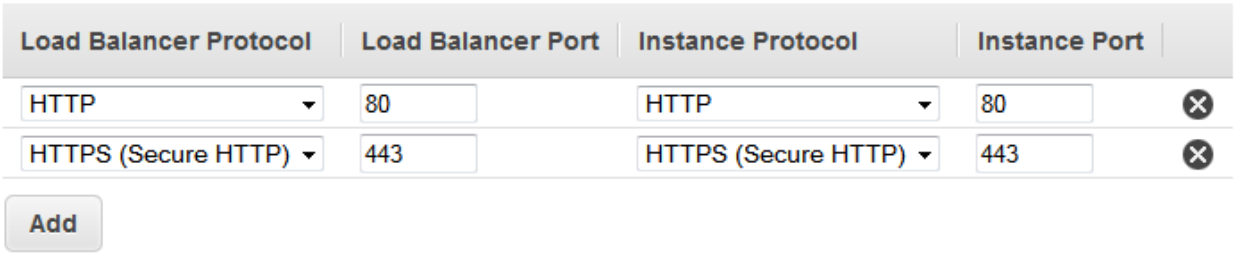
A company has a web application hosted on a fleet of EC2 instances located in two Availability Zones that are all placed behind an Application Load Balancer. As a

Solutions Architect, you have to add a health check configuration to ensure your application is highly-available.

Which health checks will you implement?

HTTP or HTTPS health check (Correct)
TCP health check
FTP health check
ICMP health check

A load balancer takes requests from clients and distributes them across the EC2 instances that are registered with the load balancer. You can create a load balancer that listens to both the HTTP (80) and HTTPS (443) ports. If you specify that the HTTPS listener sends requests to the instances on port 80, the load balancer terminates the requests, and communication from the load balancer to the instances is not encrypted. If the HTTPS listener sends requests to the instances on port 443, communication from the load balancer to the instances is encrypted.



If your load balancer uses an encrypted connection to communicate with the instances, you can optionally enable authentication of the instances. This ensures that the load balancer communicates with an instance only if its public key matches the key that you specified to the load balancer for this purpose.

The type of ELB that is mentioned in this scenario is an Application Elastic Load Balancer. This is used if you want a flexible feature set for your web applications with HTTP and HTTPS traffic. Conversely, it only allows 2 types of health check: HTTP and HTTPS.

Hence, the correct answer is: **HTTP or HTTPS health check.**

ICMP health check and **FTP health check** are incorrect as these are not supported.

TCP health check is incorrect. A TCP health check is only offered in Network Load Balancers and Classic Load Balancers.

References:

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html

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

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

EC2 Instance Health Check vs. ELB Health Check vs. Auto Scaling and Custom Health Check:

https://tutorialsdojo.com/ec2-instance-health-check-vs-elb-health-check-vs-auto-scaling-and-custom-health-check/

Comparison of AWS Services Cheat Sheets:

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

Topic-Based – IAM (SA-Associate)

1. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A company has multiple AWS sandbox accounts that are used by its development team. All developers must be given access to the contents of one of the main account's S3 buckets. For security purposes, any personally identifiable information (PII) or financial data uploaded in the bucket must be continuously monitored and removed.

How can this be done at the lowest possible cost and with the least amount of configuration effort?

- Generate a pre-signed URL for the objects on the S3 bucket. Use the Amazon S3 Storage Lens to discover personally identifiable information (PII) or financial data.
- Create an S3 bucket policy that grants access from the sandbox accounts. Use Amazon Macie to discover personally identifiable information (PII) or financial data. (Correct)
- Add S3 read permission to the IAM policy of each IAM user from the sandbox accounts. Use Amazon Detective to discover personally identifiable information (PII) or financial data.
- Configure cross-account replication on the S3 bucket. Integrate AWS Audit Manager with the S3 bucket to discover any personally identifiable information (PII) or financial data.

In Amazon S3, you can grant users in another AWS account (Account B) granular cross-account access to objects owned by your account (Account A). Depending on the type of access that you want to provide, use one of the following solutions to grant cross-account access to objects:

– AWS Identity and Access Management (IAM) policies and resource-based bucket policies (for programmatic-only access to S3 bucket objects

– IAM policies and resource-based Access Control Lists (ACLs) for programmatic-only access to S3 bucket objects

– Cross-account IAM roles for programmatic and console access to S3 bucket objects.

Not all AWS services support resource-based policies. Therefore, you can use cross-account IAM roles to centralize permission management when providing cross-account access to multiple services. Using cross-account IAM roles simplifies provisioning cross-account access to S3 objects that are stored in multiple S3 buckets. As a result, you don't need to manage multiple policies for S3 buckets. This method allows cross-account access to objects owned or uploaded by another AWS account or AWS services. If you don't use cross-account IAM roles, then the object ACL must be modified.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": ["arn:aws:iam::111111111111:user/dev1",
                        "arn:aws:iam::222222222222:user/dev2",
                        "arn:aws:iam::333333333333:user/dev3"]
            },
            "Action": "s3:GetObject",
            "Resource": [
                "arn:aws:s3:::MainBucket/*"
            ]
        }
    ]
}
```

In the scenario, the best approach to granting the developers access to the main account's S3 bucket is by configuring the bucket policy to allow IAM users from different accounts to call the GetObject method. This is a neater and simpler solution than the rest because you control access from a single location without any additional costs.

Hence, the correct answer is: Create an S3 bucket policy that grants access from the sandbox accounts. Use Amazon Macie to discover personally identifiable information (PII) or financial data.

The option that says: **Configure cross-account replication on the S3 bucket. Integrate AWS Audit Manager with the S3 bucket to discover any personally identifiable information (PII) or financial data** is incorrect. This can work, but it is an inefficient way of solving the problem. The developers only need to access the S3 objects in another account; they do not need to own a copy of them. On top of that, replication incurs additional costs. In addition, the AWS Audit Manager simply helps you continuously audit your AWS usage to simplify how you assess risk and compliance with regulations and industry standards. AWS

Audit Manager is not capable of discovering personally identifiable information (PII) or financial data in your S3 bucket.

The option that says: **Generate a pre-signed URL for the objects on the S3 bucket. Use the Amazon S3 Storage Lens to discover personally identifiable information (PII) or financial data** is incorrect. Since objects shared using presigned URLs are time-limited, you'd have to regenerate the URL for each object every time it expires and resend the new link to the developers. This approach does not scale well and is not a good use for the S3 presigned URL. Moreover, the Amazon S3 Storage Lens feature just provides a single view of object storage usage and activity across your entire Amazon S3 storage.

The option that says: **Add S3 read permission to the IAM policy of each IAM user from the sandbox accounts. Use Amazon Detective to discover personally identifiable information (PII) or financial data** is incorrect. You would have to jump from one account to another to set this up. It works, but depending on the number of accounts and IAM users, it will entail a lot of configuration overhead. Although Amazon Detective is a security service, it does not have any capability to discover any PII or financial data in your S3 bucket. Its primary purpose is to analyze and visualize security data to rapidly get to the root cause of potential security issues.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/cross-account-access-s3/

https://docs.aws.amazon.com/AmazonS3/latest/userguide/example-walkthroughs-managing-access-example2.html

https://aws.amazon.com/macie/

Check out this Amazon S3 Cheat Sheet:

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

2. QUESTION

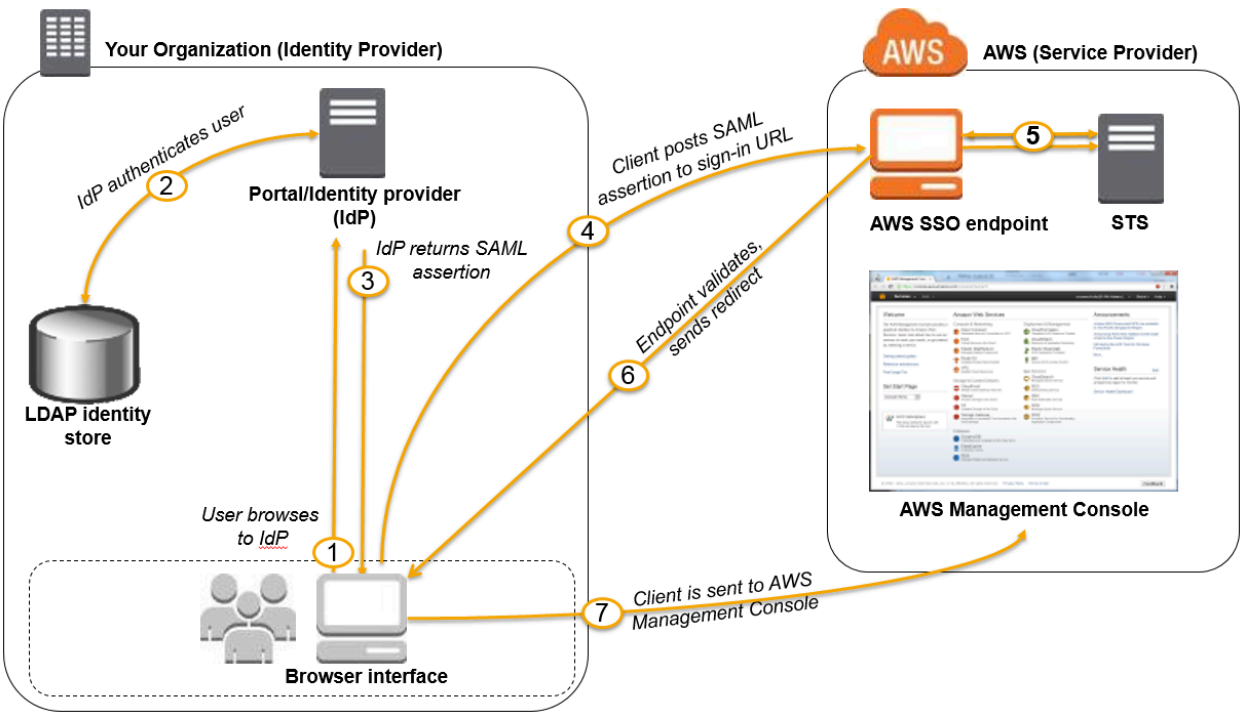
Category: CSAA – Design Secure Architectures

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

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/

3. 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.)

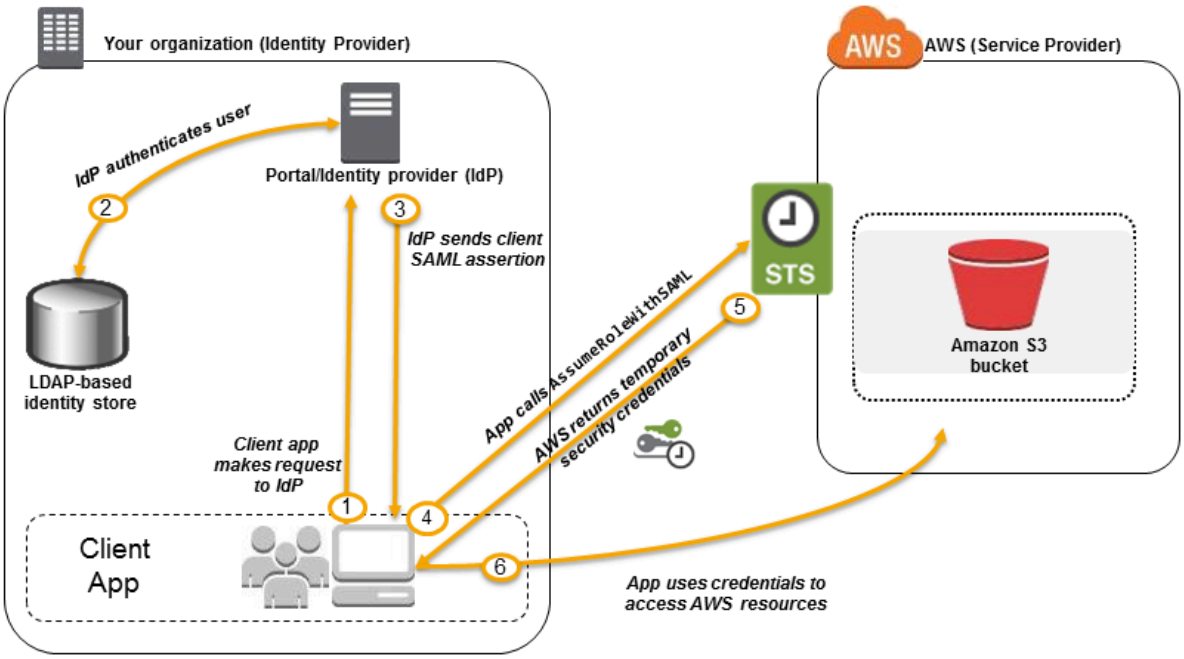
Configure an IAM role and an IAM Policy to access the bucket. (Correct)
Map each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents.
Set up a Federation proxy or an Identity provider, and use AWS Security Token Service to generate temporary tokens. (Correct)
Set up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket.
Use 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others.

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, and use 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/

Check out this AWS IAM Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Secure Architectures

A Solutions Architect created a brand new IAM User with a default setting using AWS CLI. This is intended to be used to send API requests to Amazon S3, DynamoDB, Lambda, and other AWS resources of the company's cloud infrastructure.

Which of the following must be done to allow the user to make API calls to the AWS resources?

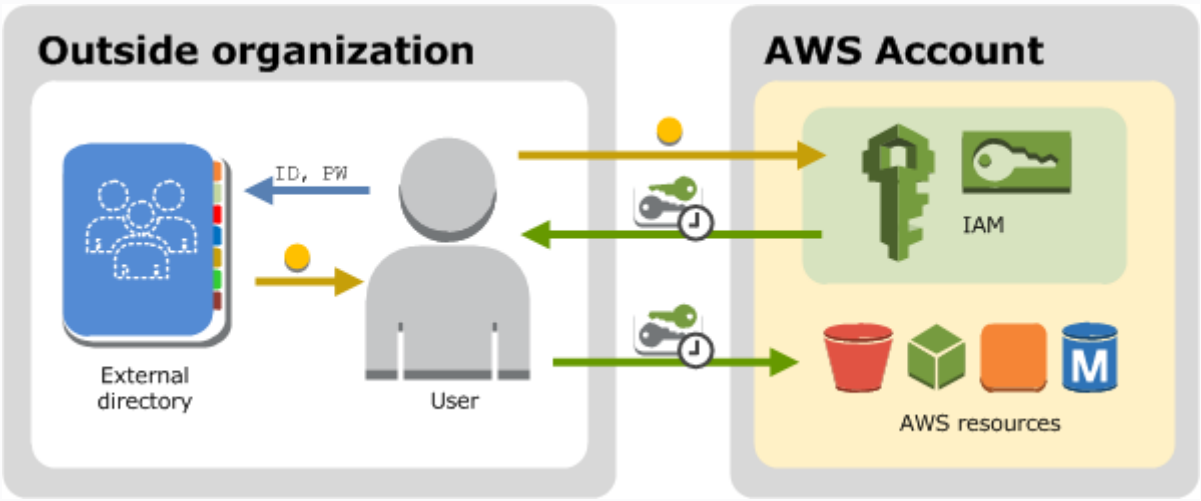
Assign an IAM Policy to the user to allow it to send API calls.
Enable Multi-Factor Authentication for the user.
Do nothing as the IAM User is already capable of sending API calls to your AWS resources.
Create a set of Access Keys for the user and attach the necessary permissions. (Correct)

You can choose the credentials that are right for your IAM user. When you use the AWS Management Console to create a user, you must choose to include at least a console password or access keys. By default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. You must create the type of credentials for an IAM user based on the needs of your user.

Access keys are long-term credentials for an IAM user or the AWS account root user. You can use access keys to sign programmatic requests to the AWS CLI or AWS API (directly or using the AWS SDK). Users need their own access keys to make programmatic calls to

AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.

To fill this need, you can create, modify, view, or rotate access keys (access key IDs and secret access keys) for IAM users. When you create an access key, IAM returns the access key ID and secret access key. You should save these in a secure location and give them to the user.



The option that says: **Do nothing as the IAM User is already capable of sending API calls to your AWS resources** is incorrect because by default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. Take note that in the scenario, you created the new IAM user using the AWS CLI and not via the AWS Management Console, where you must choose to at least include a console password or access keys when creating a new IAM user.

Enabling Multi-Factor Authentication for the user is incorrect because this will still not provide the required Access Keys needed to send API calls to your AWS resources. You have to grant the IAM user with Access Keys to meet the requirement.

Assigning an IAM Policy to the user to allow it to send API calls is incorrect because adding a new IAM policy to the new user will not grant the needed Access Keys needed to make API calls to the AWS resources.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html#id_users_creds

Check out this AWS IAM Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Secure Architectures

An Intelligence Agency developed a missile tracking application that is hosted on both development and production AWS accounts. The Intelligence agency's junior developer only has access to the development account. She has received security clearance to access the agency's production account but the access is only temporary and only write access to EC2 and S3 is allowed.

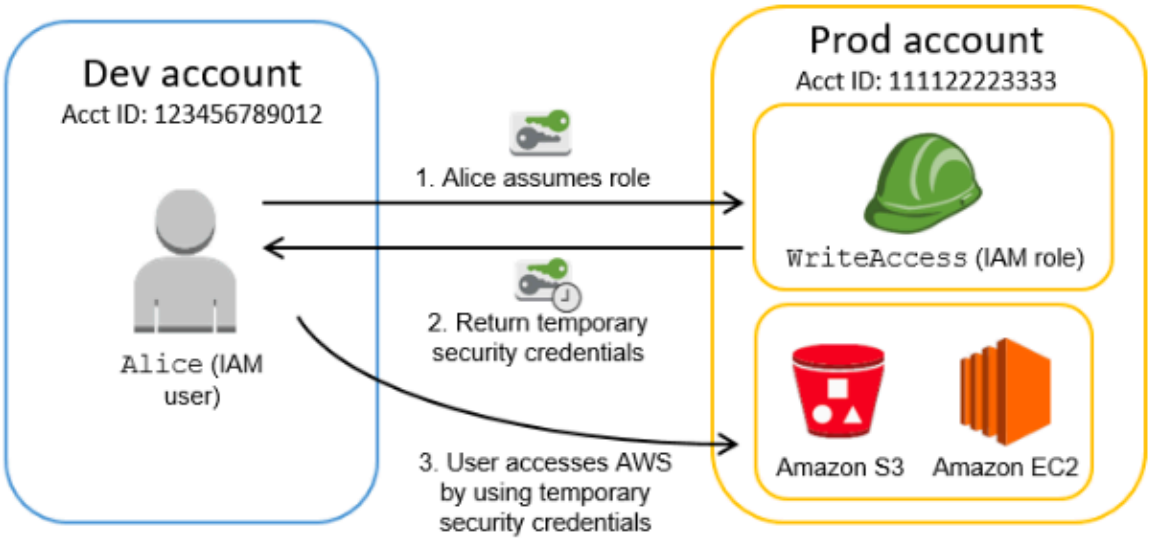
Which of the following allows you to issue short-lived access tokens that act as temporary security credentials to allow access to your AWS resources?

Use AWS Cognito to issue JSON Web Tokens (JWT)
All of the given options are correct.
Use AWS IAM Identity Center
Use AWS Security Token Service (STS) (Correct)

AWS Security Token Service (STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

1. Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
2. STS returns a set of temporary security credentials.
3. Alice uses the temporary security credentials to access services and resources in the Prod account. Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.



Using AWS Cognito to issue JSON Web Tokens (JWT) is incorrect because the Amazon Cognito service is primarily used for user authentication and not for providing access to your AWS resources. A JSON Web Token (JWT) is meant to be used for user authentication and session management.

Using AWS AWS IAM Identity Center is incorrect because this is simply a successor to the AWS Single Sign-On service that helps you securely create or connect your workforce identities and manage their access centrally across AWS accounts and applications. IAM Identity Center is the recommended approach for workforce authentication and authorization on AWS for organizations of any size and type, but not for generating tokens.

The option that says **All of the above** is incorrect as only STS has the ability to provide temporary security credentials.

References:

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

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

Check out this AWS IAM Cheat Sheet:

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

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

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

6. QUESTION

Category: CSAA – Design Secure Architectures

A Solutions Architect is managing a company's AWS account of approximately 300 IAM users. They have a new company policy that requires changing the associated permissions of all 100 IAM users that control the access to Amazon S3 buckets.

What will the Solutions Architect do to avoid the time-consuming task of applying the policy to each user?

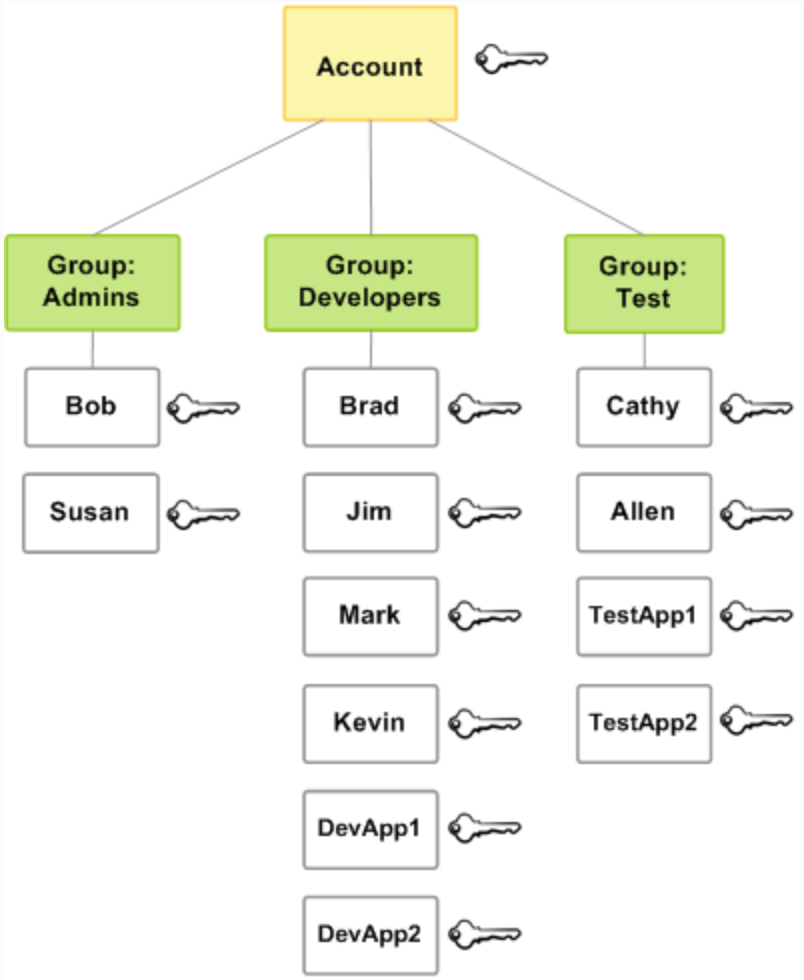
Create a new IAM group and then add the users that require access to the S3 bucket. Afterwards, apply the policy to IAM group. (Correct)
Create a new IAM role and add each user to the IAM role.
Create a new policy and apply it to multiple IAM users using a shell script.
Create a new S3 bucket access policy with unlimited access for each IAM user.

In this scenario, the best option is to **Create a new IAM group and then add the users that require access to the S3 bucket. Afterward, apply the policy to the IAM group.** This will enable you to easily add, remove, and manage the users instead of manually adding a policy to each and every 100 IAM users.

Creating a new policy and applying it to multiple IAM users using a shell script is incorrect because you need a new IAM Group for this scenario and not assign a policy to each user via a shell script. This method can save you time but afterward, it will be difficult to manage all 100 users that are not contained in an IAM Group.

Creating a new S3 bucket access policy with unlimited access for each IAM user is incorrect because you need a new IAM Group and the method is also time-consuming.

Creating a new IAM role and adding each user to the IAM role is incorrect because you need to use an IAM Group and not an IAM role.



Reference:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

Check out this AWS IAM Cheat Sheet:

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

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

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

7. QUESTION

Category: CSAA – Design Secure Architectures

A company needs to integrate the Lightweight Directory Access Protocol (LDAP) directory service from the on-premises data center to the AWS VPC using IAM. The identity store which is currently being used is not compatible with SAML.

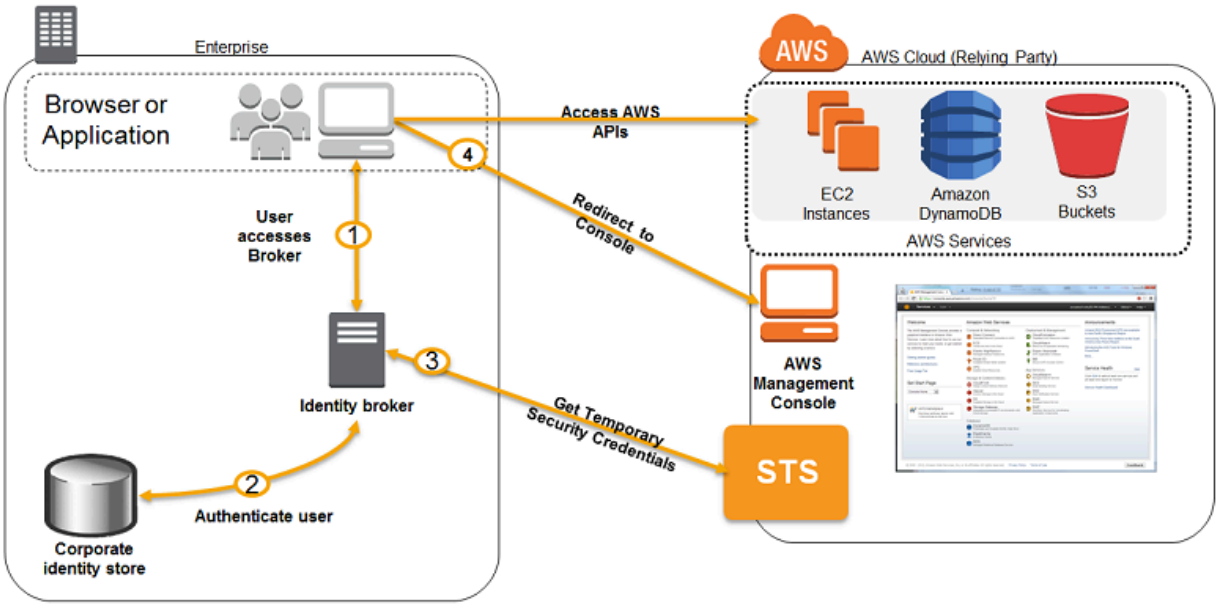
Which of the following provides the most valid approach to implement the integration?

Use IAM roles to rotate the IAM credentials whenever LDAP credentials are updated.
Use an IAM policy that references the LDAP identifiers and AWS credentials.
Use AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP.
Develop an on-premises custom identity broker application and use STS to issue short-lived AWS credentials. (Correct)

If your identity store is not compatible with SAML 2.0 then you can build a custom identity broker application to perform a similar function. The broker application authenticates users, requests temporary credentials for users from AWS, and then provides them to the user to access AWS resources.

The application verifies that employees are signed into the existing corporate network's identity and authentication system, which might use LDAP, Active Directory, or another system. The identity broker application then obtains temporary security credentials for the employees.

To get temporary security credentials, the identity broker application calls either `AssumeRole` or `GetFederationToken` to obtain temporary security credentials, depending on how you want to manage the policies for users and when the temporary credentials should expire. The call returns temporary security credentials consisting of an AWS access key ID, a secret access key, and a session token. The identity broker application makes these temporary security credentials available to the internal company application. The app can then use the temporary credentials to make calls to AWS directly. The app caches the credentials until they expire, and then requests a new set of temporary credentials.



Using an IAM policy that references the LDAP identifiers and AWS credentials is incorrect because using an IAM policy is not enough to integrate your LDAP service to IAM. You need to use SAML, STS, or a custom identity broker.

Using AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP is incorrect because the scenario did not require SSO and in addition, the identity store that you are using is not SAML-compatible.

Using IAM roles to rotate the IAM credentials whenever LDAP credentials are updated is incorrect because manually rotating the IAM credentials is not an optimal solution to integrate your on-premises and VPC network. You need to use SAML, STS, or a custom identity broker.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_federated-users.html

https://aws.amazon.com/blogs/aws/aws-identity-and-access-management-now-with-identity-federation/

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

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

8. QUESTION

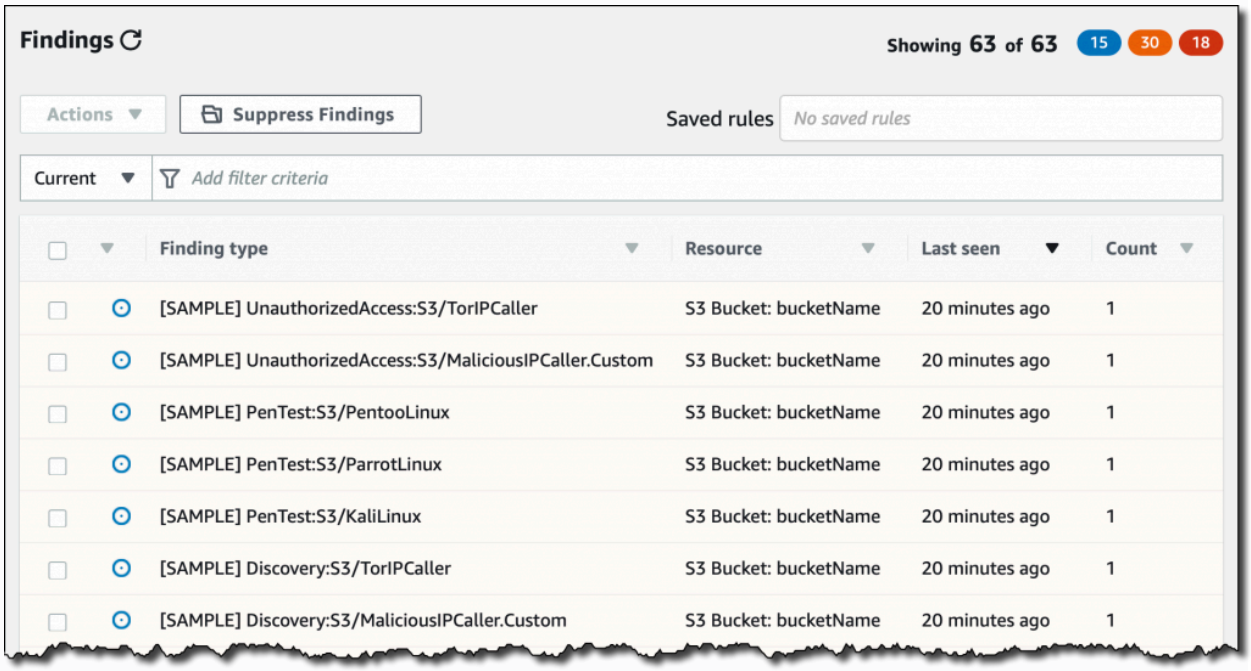
Category: CSAA – Design Secure Architectures

A company has an application that continually sends encrypted documents to Amazon S3. The company requires that the configuration for data access is in line with their strict compliance standards. They should also be alerted if there is any risk of unauthorized access or suspicious access patterns.

Which step is needed to meet the requirements?

Use Amazon Macie to monitor and detect access patterns on S3.
Use Amazon GuardDuty to monitor malicious activity on S3. (Correct)
Use Amazon Inspector to alert whenever a security violation is detected on S3.
Use Amazon Rekognition to monitor and recognize patterns on S3.

Amazon GuardDuty can generate findings based on suspicious activities such as requests coming from known malicious IP addresses, changing of bucket policies/ACLs to expose an S3 bucket publicly, or suspicious API call patterns that attempt to discover misconfigured bucket permissions.



To detect possibly malicious behavior, GuardDuty uses a combination of anomaly detection, machine learning, and continuously updated threat intelligence.

Hence, the correct answer is: Use Amazon GuardDuty to monitor malicious activity on S3.

The option that says: **Use Amazon Rekognition to monitor and recognize patterns on S3** is incorrect because Amazon Rekognition is simply a service that can identify the objects, people, text, scenes, and activities on your images or videos, as well as detect any inappropriate content.

The option that says: **Use Amazon Macie to monitor and detect access patterns on S3** is incorrect because Macie cannot detect usage patterns on S3 data. While Amazon Macie is capable of detecting policy changes in S3 buckets, this is not enough to detect unauthorized or suspicious access patterns.

The option that says: Use Amazon Inspector to alert whenever a security violation is detected on S3 is incorrect because Inspector is basically an automated security assessment service that helps improve the security and compliance of applications deployed on AWS.

References:

https://aws.amazon.com/guardduty/

https://aws.amazon.com/blogs/aws/new-using-amazon-guardduty-to-protect-your-s3-buckets/

Check out this Amazon GuardDuty Cheat Sheet:

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

Topic-Based – Lambda (SA-Associate)

1. 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.)

- 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.
- 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.
- 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.
- 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)

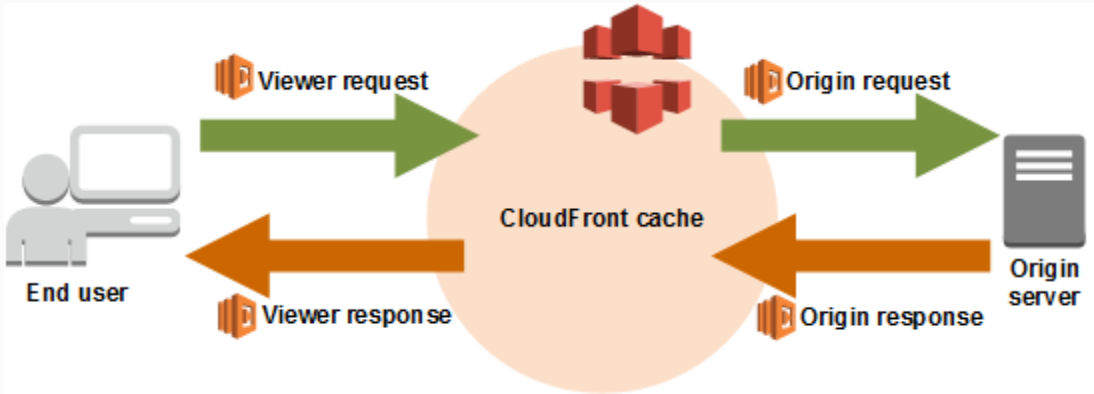
Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests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

2. QUESTION

Category: CSAA – Design Resilient Architectures

An application is using a Lambda function to process complex financial data that run for 15 minutes on average. Most invocations were successfully processed. However, you noticed that there are a few terminated invocations throughout the day, which caused data discrepancy in the application.

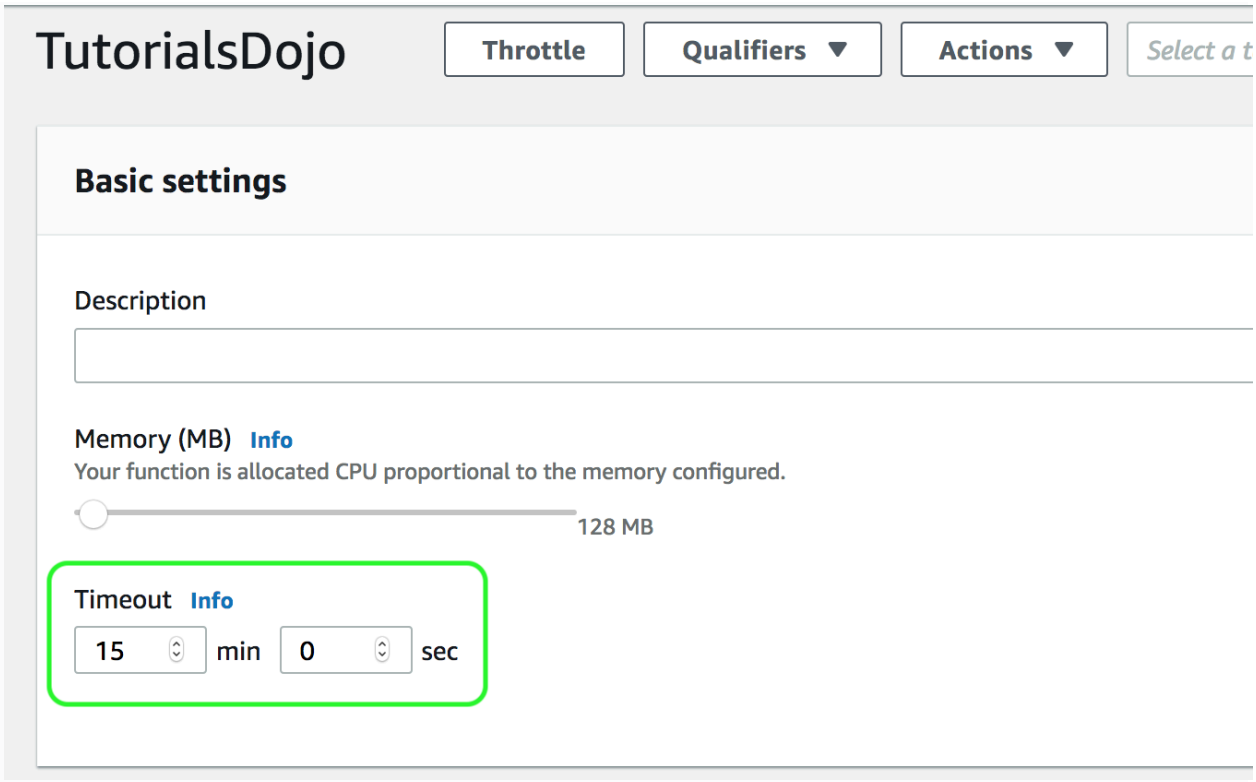
Which of the following is the most likely cause of this issue?

The failed Lambda functions have been running for over 15 minutes and reached the maximum execution time. (Correct)
The Lambda function contains a recursive code and has been running for over 15 minutes.
The failed Lambda Invocations contain a `ServiceException` error which means that the AWS Lambda service encountered an internal error.
The concurrent execution limit has been reached.

A Lambda function consists of code and any associated dependencies. In addition, a Lambda function also has configuration information associated with it. Initially, you specify the configuration information when you create a Lambda function. Lambda provides an API for you to update some of the configuration data.

You pay for the AWS resources that are used to run your Lambda function. To prevent your Lambda function from running indefinitely, you specify a timeout. When the specified timeout is reached, AWS Lambda terminates execution of your Lambda function. It is recommended that you set this value based on your expected execution time. The default timeout is 3 seconds, and the maximum execution duration per request in AWS Lambda is 900 seconds, which is equivalent to 15 minutes.

Hence, the correct answer is the option that says: **The failed Lambda functions have been running for over 15 minutes and reached the maximum execution time**.



Take note that you can invoke a Lambda function synchronously either by calling the `Invoke` operation or by using an AWS SDK in your preferred runtime. If you anticipate a long-running Lambda function, your client may time out before function execution completes. To avoid this, update the client timeout or your SDK configuration.

The option that says: **The concurrent execution limit has been reached** is incorrect because, by default, the AWS Lambda limits the total concurrent executions across all functions within a given region to 1000. By setting a concurrency limit on a function, Lambda guarantees that allocation will be applied specifically to that function, regardless of the amount of traffic processing the remaining functions. If that limit is exceeded, the function will be throttled but not terminated, which is in contrast with what is happening in the scenario.

The option that says: **The Lambda function contains a recursive code and has been running for over 15 minutes** is incorrect because having a recursive code in your Lambda function does not directly result to an abrupt termination of the function execution. This is a scenario wherein the function automatically calls itself until some arbitrary criteria is met. This could lead to an unintended volume of function invocations and escalated costs, but not an abrupt termination because Lambda will throttle all invocations to the function.

The option that says: **The failed Lambda Invocations contain a `ServiceException` error which means that the AWS Lambda service encountered an internal error** is incorrect. Although this is a valid root cause, it is unlikely to have several ServiceException errors throughout the day unless there is an outage or disruption in AWS. Since the

scenario says that the Lambda function runs for about 10 to 15 minutes, the maximum execution duration is the most likely cause of the issue and not the AWS Lambda service encountered an internal error.

References:

https://docs.aws.amazon.com/lambda/latest/dg/limits.html

https://docs.aws.amazon.com/lambda/latest/dg/resource-model.html

Check out this AWS Lambda Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design High-Performing Architectures

A media company wants to ensure that the images it delivers through Amazon CloudFront are compatible across various user devices. The company plans to serve images in WebP format to user agents that support it and return to JPEG format for those that don't. Additionally, they want to add a custom header to the response for tracking purposes.

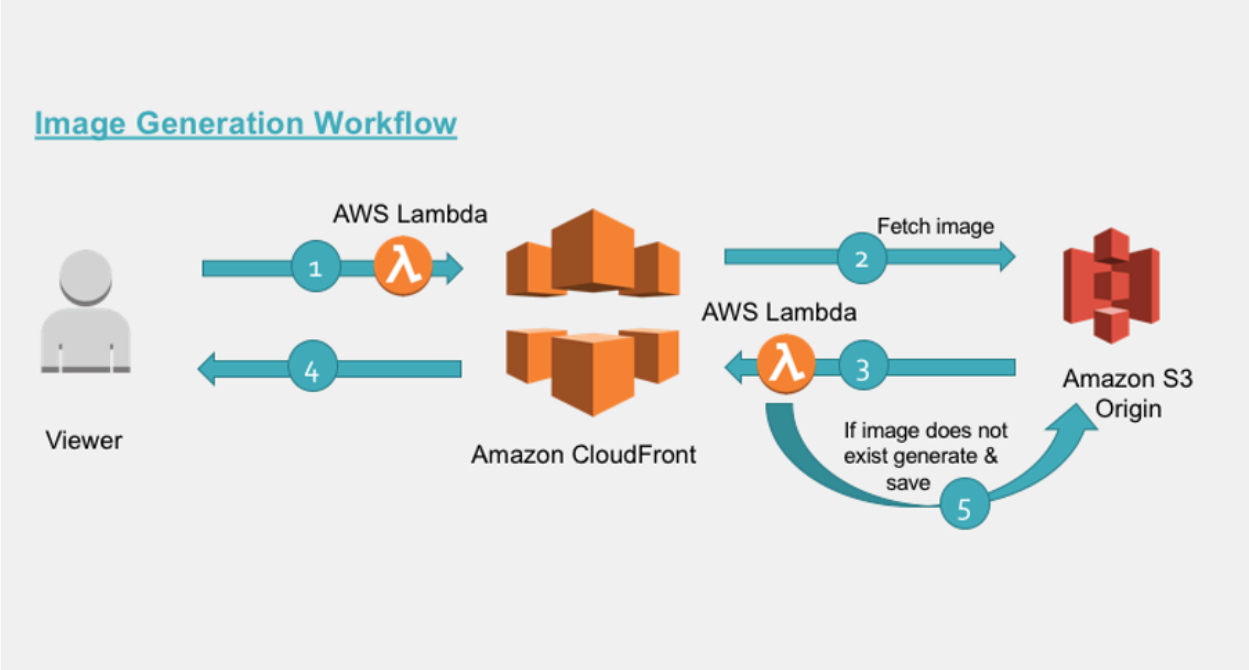
As a solution architect, what approach would you recommend to meet these requirements while minimizing operational overhead?

Configure CloudFront behaviors to handle different image formats based on the User-Agent header. Use Lambda@Edge functions to modify the response headers and serve the appropriate format. (Correct)
Create multiple CloudFront distributions, each serving a specific image format (WebP or JPEG). Route incoming requests based on the User-Agent header to the respective distribution using Amazon Route 53.
Generate a CloudFront response headers policy. Utilize the policy to deliver the suitable image format according to the User-Agent HTTP header in the incoming request.

Implement an image conversion service on EC2 instances and integrate it with CloudFront. Use Lambda functions to modify the response headers and serve the appropriate format based on the User-Agent header.

Amazon CloudFront is a content delivery network (CDN) service that enables the efficient distribution of web content to users across the globe. It reduces latency by caching static and dynamic content in multiple edge locations worldwide and improves the overall user experience.

Lambda@Edge allows you to run Lambda functions at the edge locations of the CloudFront CDN. With this, you can perform various tasks, such as modifying HTTP headers, generating dynamic responses, implementing security measures, and customizing content based on user preferences, device type, location, or other criteria.



When a request is made to a CloudFront distribution, Lambda@Edge enables you to intercept the request and execute custom code before CloudFront processes it. Similarly, you can intercept the response generated by CloudFront and modify it before it's returned to the viewer. In the given scenario, Lambda@Edge can be used to dynamically serve different image formats based on the User-agent header received by CloudFront. Additionally, you can inject custom response headers before CloudFront returns the response to the viewer.

Hence the correct answer is: **Configure CloudFront behaviors to handle different image formats based on the User-Agent header. Use Lambda@Edge functions to modify the response headers and serve the appropriate format.**

The option that says: **Create multiple CloudFront distributions, each serving a specific image format (WebP or JPEG). Route incoming requests based on the User-Agent header to the respective distribution using Amazon Route 53** is incorrect because creating multiple CloudFront distributions for each image format is unnecessary and just increases operational overhead.

The option that says: **Generate a CloudFront response headers policy. Utilize the policy to deliver the suitable image format according to the User-Agent HTTP header in the incoming request** is incorrect. CloudFront response headers policies simply tell which HTTP headers should be included or excluded in the responses sent by CloudFront. You cannot use them to dynamically select and serve image formats based on the User-agent.

The option that says: **Implement an image conversion service on EC2 instances and integrate it with CloudFront. Use Lambda functions to modify the response headers and serve the appropriate format based on the User-Agent header** is incorrect. Building an image conversion service using EC2 instances requires additional operational management. You can instead use Lambda@Edge functions to modify response headers and serve the correct image format based on the User-agent header.

References:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/lambda-at-the-edge.html

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

Check out these AWS Lambda and Amazon CloudFront Cheat Sheets:

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

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

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

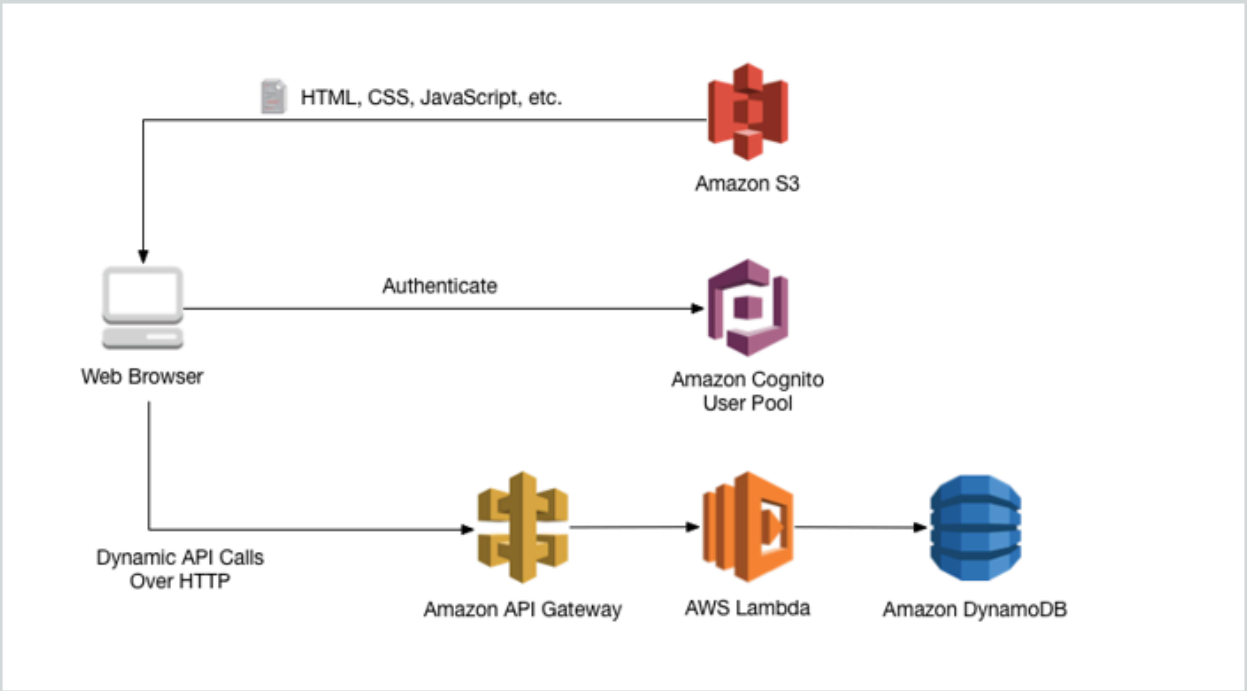
Which of the following should you recommend as the most cost-effective and scalable solution to meet the above requirement?

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

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

Lambda counts a request each time it starts executing in response to an event notification or invoke call, including test invokes from the console. You are charged for the total number of requests across all your functions.

Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 1ms. The price depends on the amount of memory you allocate to your function. The Lambda free tier includes 1M free requests per month and over 400,000 GB seconds of compute time per month.



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

Hence, the correct answer is: **Use AWS Lambda and Amazon API Gateway.**

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

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

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

provided by the AWS Cloud. Although EFA is scalable, the Spot Fleet configuration of this option doesn't have Auto Scaling involved.

References:

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

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

Check out this AWS Lambda Cheat Sheet:

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

EC2 Container Service (ECS) vs Lambda:

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

5. QUESTION

Category: CSAA – Design High-Performing Architectures

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

Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Amazon Athena and durably store the results to an Amazon S3 bucket.

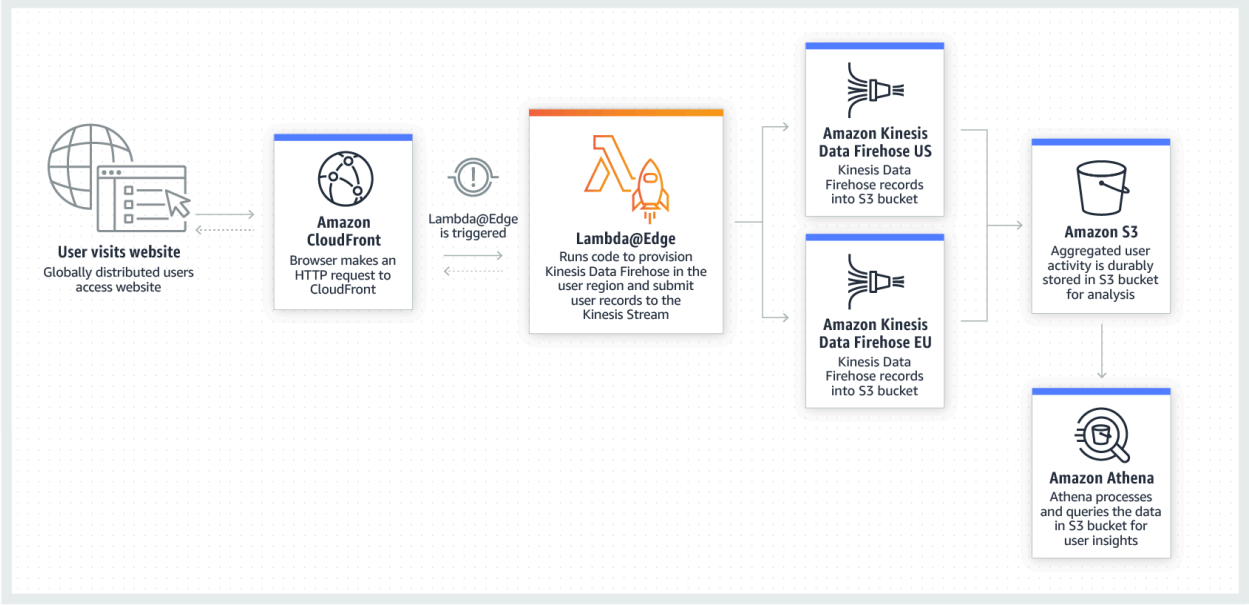
Use a CloudFront web distribution and Route 53 with a latency-based routing policy, in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.

Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket. (Correct)

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

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

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



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

The options that say: **Use a CloudFront web distribution and Route 53 with a latency-based routing policy, in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket** and **Use a CloudFront web distribution and Route 53 with a Geoproximity routing policy in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket** are both incorrect because you can only route traffic using Route 53 since it does not have any computing capability. This solution would not be able to process and return the data in close geographical proximity to your users since it is not using Lambda@Edge.

The option that says: **Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Amazon Athena and durably store the results to an Amazon S3 bucket** is incorrect. Although using Lambda@Edge is correct, Amazon Athena is just an interactive query service that enables you to easily analyze data in Amazon S3 using standard SQL. Kinesis should be used to process the streaming data in real time.

References:

https://aws.amazon.com/lambda/edge/

https://aws.amazon.com/blogs/networking-and-content-delivery/global-data-ingestion-with-amazon-cloudfront-and-lambdaedge/

6. QUESTION

Category: CSAA – Design Resilient Architectures

A company is using Amazon VPC that has a CIDR block of `10.31.0.0/27` that is connected to the on-premises data center. There was a requirement to create a Lambda function that will process massive amounts of cryptocurrency transactions every minute

and then store the results to EFS. After setting up the serverless architecture and connecting the Lambda function to the VPC, the Solutions Architect noticed an increase in invocation errors with EC2 error types such as `EC2ThrottledException` at certain times of the day.

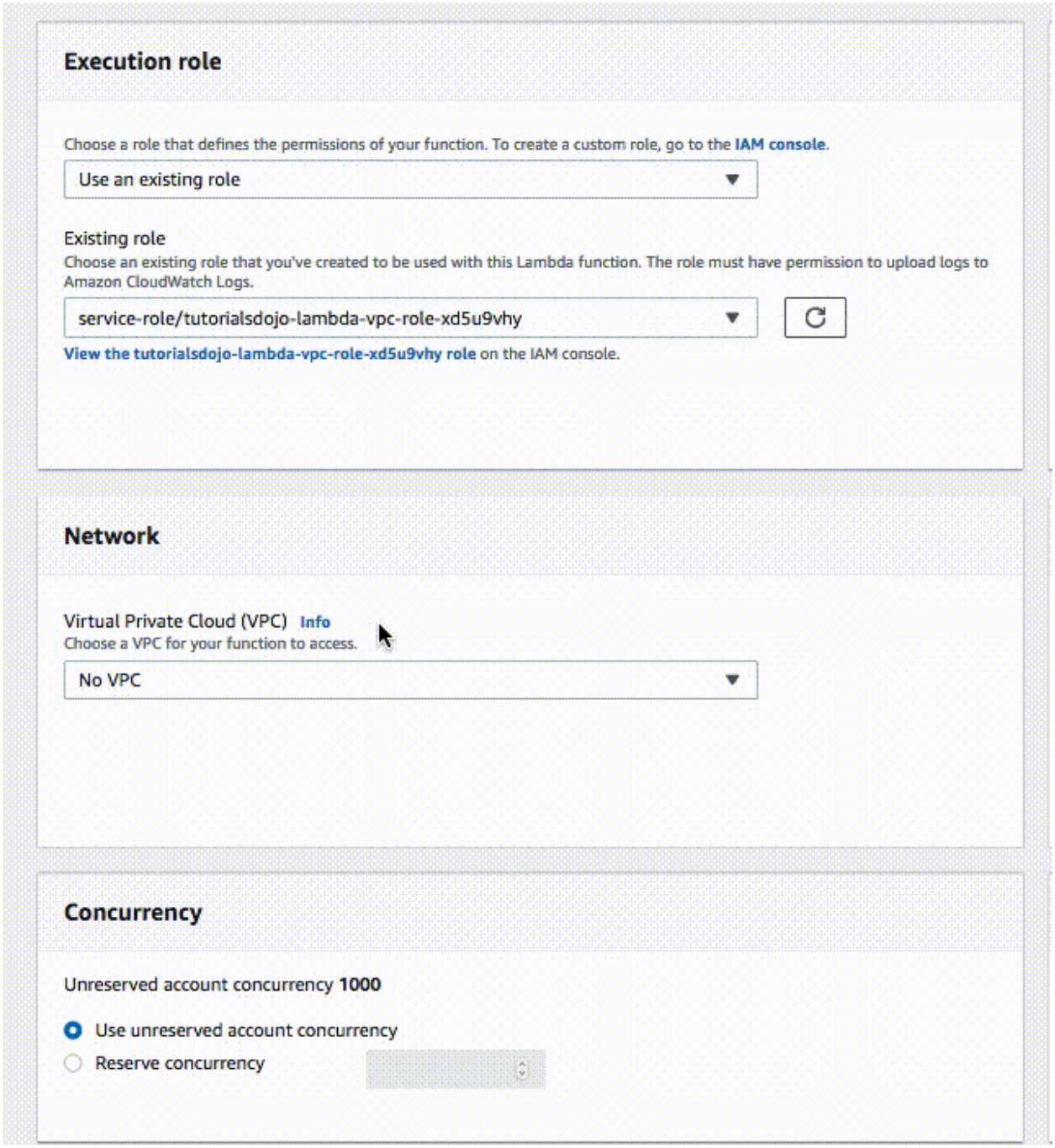
Which of the following are the possible causes of this issue? (Select TWO.)

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

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

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

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



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

By specifying subnets in each of the Availability Zones, your Lambda function can run in another Availability Zone if one goes down or runs out of IP addresses. If your VPC does not have sufficient ENIs or subnet IPs, your Lambda function will not scale as requests

increase, and you will see an increase in invocation errors with EC2 error types like `EC2ThrottledException`. For asynchronous invocation, if you see an increase in errors without corresponding CloudWatch Logs, invoke the Lambda function synchronously in the console to get the error responses.

Hence, the correct answers for this scenario are:

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

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

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

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

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

References:

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

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

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

Check out this AWS Lambda Cheat Sheet:

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

7. QUESTION

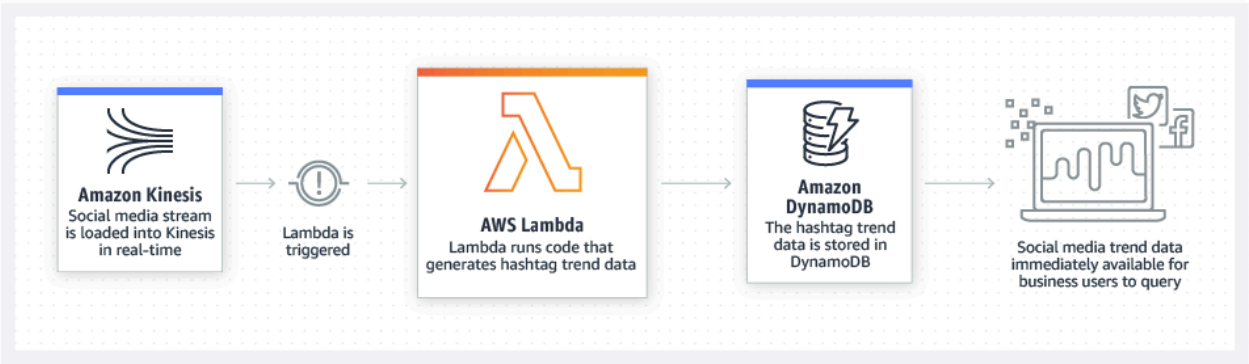
Category: CSAA – Design Cost-Optimized Architectures

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

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

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

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



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

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

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

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

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

References:

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

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

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

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

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

8. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A wellness company is currently working on a wearable device that monitors key health metrics such as heart rate, sleep, and steps per day. The device is designed to send data to an Amazon S3 bucket for storage and analysis. On a daily basis, the device produces 1 MB of data. In order to quickly process and summarize this data, the company requires 512 MB of memory and must complete the task within a maximum of 10 seconds.

Which solution can fulfill these requirements in the MOST cost-effective manner?

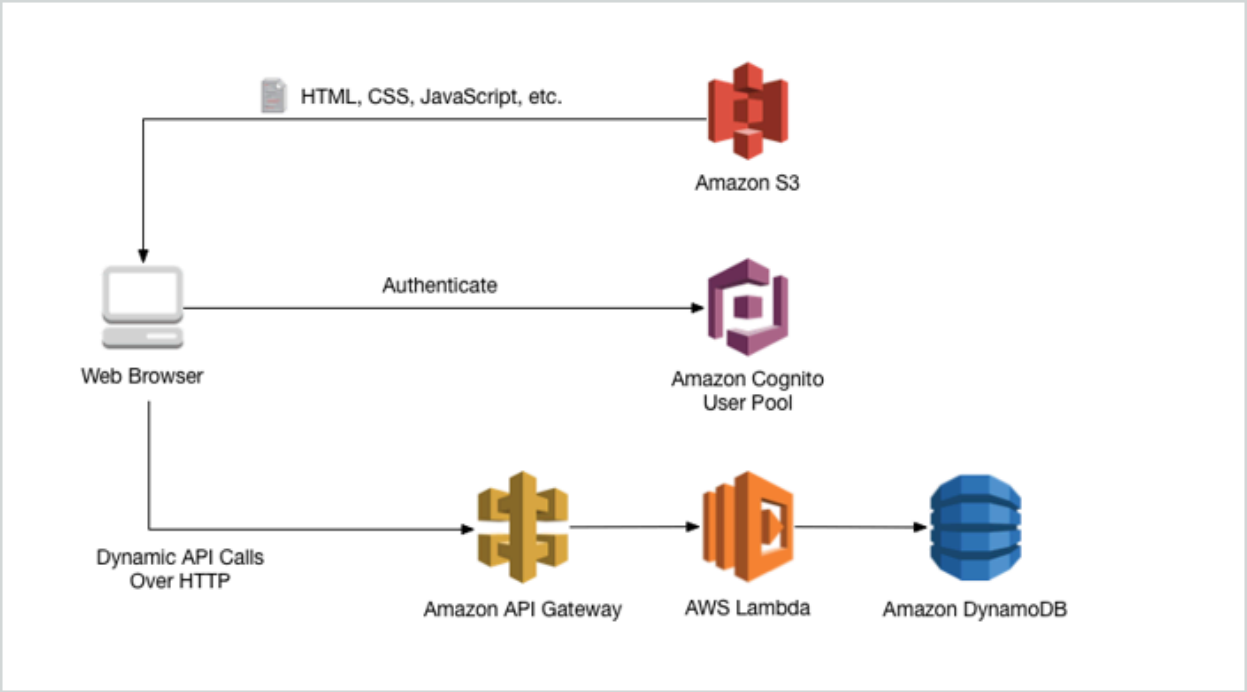
Store the data in Amazon Redshift and process it with AWS Lambda.
Use AWS Lambda with a Python library for processing. (Correct)
Create an AWS Glue PySpark job to process the data.
Use Amazon Kinesis Data Firehose to send the data from the device to Amazon S3. Process the data on an EC2 instance with at least 512 MB of memory.

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

Lambda counts a request each time it starts executing in response to an event notification or invoke call, including test invokes from the console. You are charged for the total number of requests across all your functions.

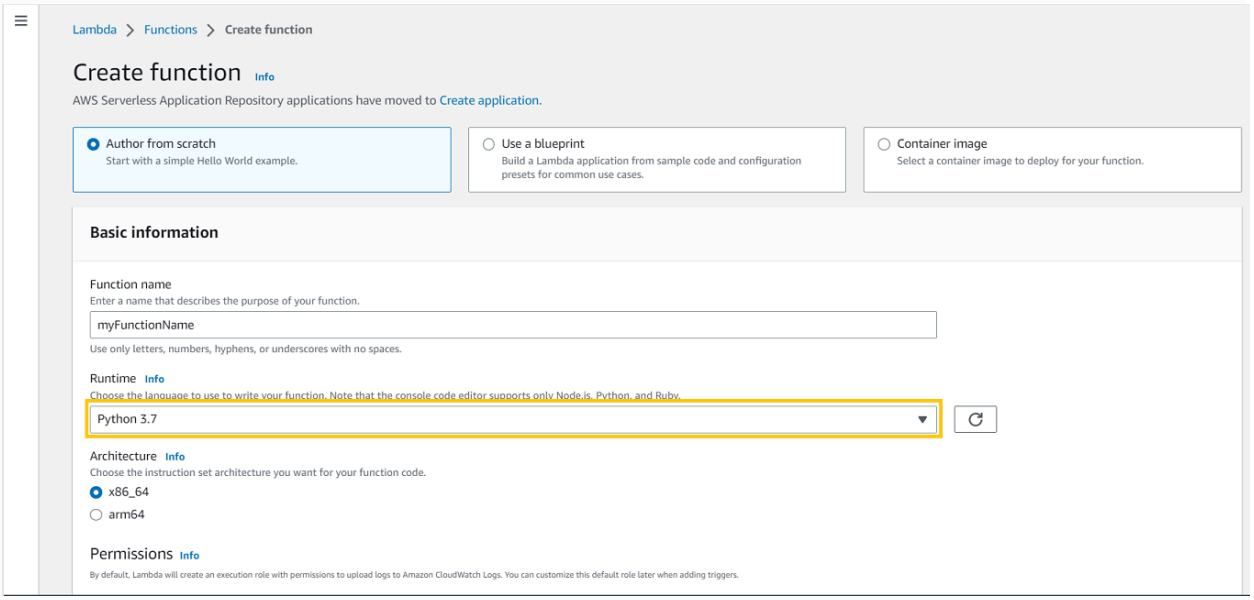
Duration is calculated from the time your code begins executing until it returns or otherwise terminates, rounded up to the nearest 1ms. The price depends on the amount of memory

you allocate to your function. The Lambda free tier includes 1M free requests per month and over 400,000 GB seconds of compute time per month.



The seamless integration of AWS Lambda with other AWS services, like Amazon S3 where health data is stored, enables effortless data transfer and processing. This guarantees smooth retrieval of data from the wearable device and its subsequent analysis.

With serverless computing, AWS Lambda eliminates the need for infrastructure management, and automatic scaling based on workload is possible. Python is a versatile and widely-used programming language with extensive libraries, making it ideal for efficient data processing, analysis, and summarization due to its simplicity.



Hence, the correct answer is: **Use AWS Lambda with a Python library for processing.**

The option that says: Create an AWS Glue PySpark job to process the data is incorrect. While it's a valid solution, it's a bit of overkill to use AWS Glue for processing 1 MB of data per day. AWS Glue has a minimum billing duration of 1 minute (Glue version 2.0 and later), which is not cost-effective for processing small amounts of data quickly (in 10 seconds).

The option that says: **Use Amazon Kinesis Data Firehose to send the data from the device to Amazon S3. Process the data on an EC2 instance with at least 512 MB of memory** is incorrect. Kinesis Data Firehose is meant for delivering streaming data to a storage. Using it to transport 1 MB of data per day is not cost-effective and would only complicate the process. Additionally, processing data on an EC2 instance requires more management and can be more costly in comparison to a serverless solution like AWS Lambda.

The option that says: Store the data in Amazon Redshift and process it with AWS Lambda. is incorrect. If you're looking for a reliable data warehousing service to handle analytics workloads on large datasets, Amazon Redshift is a great option. Given the scale of the data being generated (1 MB per day), using Redshift would be a poor fit. It would be significantly more expensive than necessary for this small amount of data, and it would not offer any significant benefit over using S3 as the storage backend and processing with AWS Lambda.

References:

https://aws.amazon.com/what-is/python/

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

Check out this AWS Lambda Cheat Sheet:

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

Topic-Based – RDS (SA-Associate)

1. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

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

You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read

throughput. Read replicas can also be promoted when needed to become standalone DB instances.

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

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

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

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

Hence, the correct answers are:

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

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

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

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

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

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

2. QUESTION

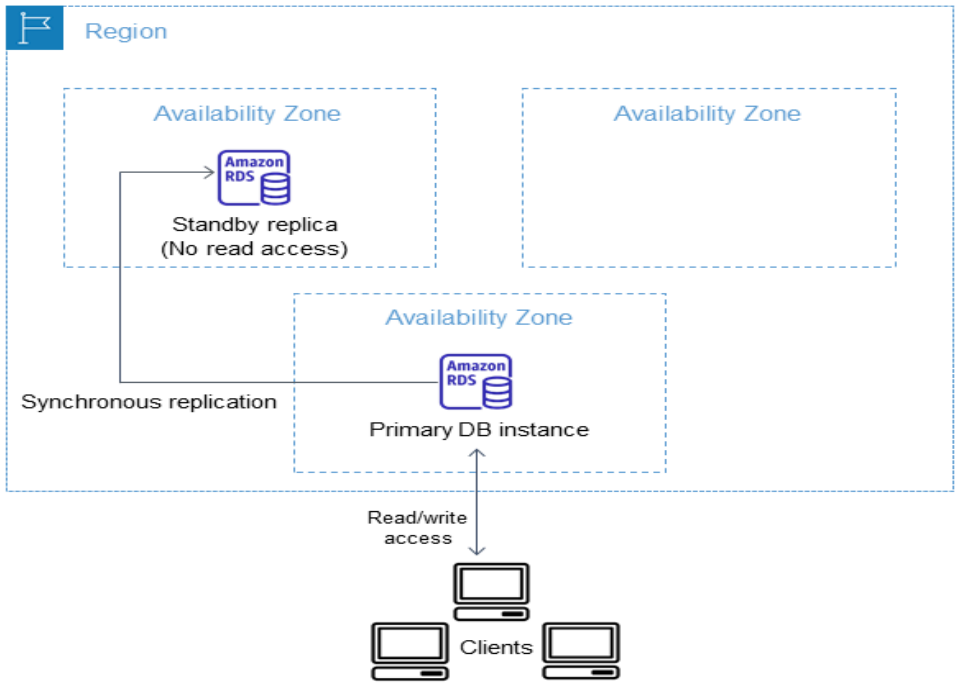
Category: CSAA – Design Resilient Architectures

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

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



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

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

3. 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?

- DynamoDB Read Replica
- RDS Read Replica
- RDS DB instance running as a Multi-AZ deployment (Correct)
- CloudFront running as a Multi-AZ deployment

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 DB instance running as a Multi-AZ deployment is correct among the options provided because it ensures synchronous data replication, making it the correct choice for improving the database's high availability in this scenario.

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/

Check out this Amazon RDS Cheat Sheet:

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

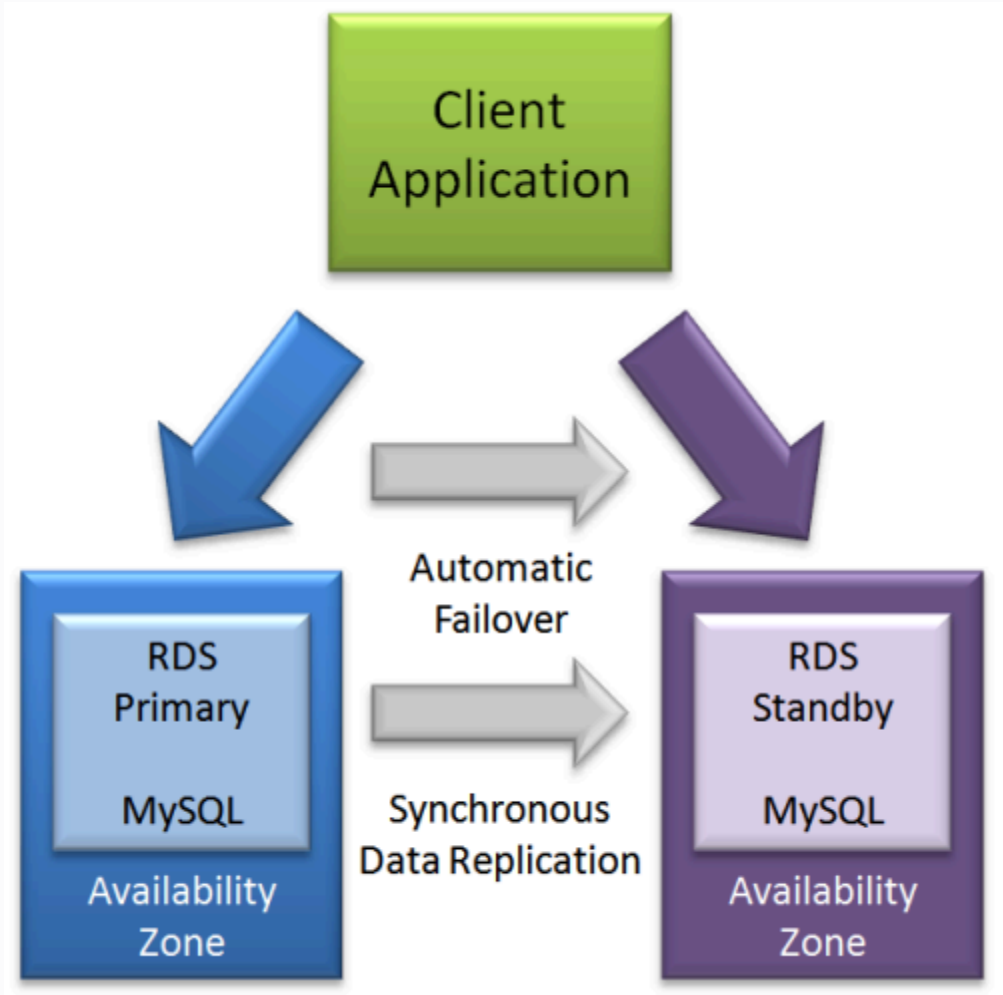
4. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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



Hence, the correct answer is: **The canonical name record (CNAME) is switched from the primary to standby instance.**

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

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

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

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Secure Architectures

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

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

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

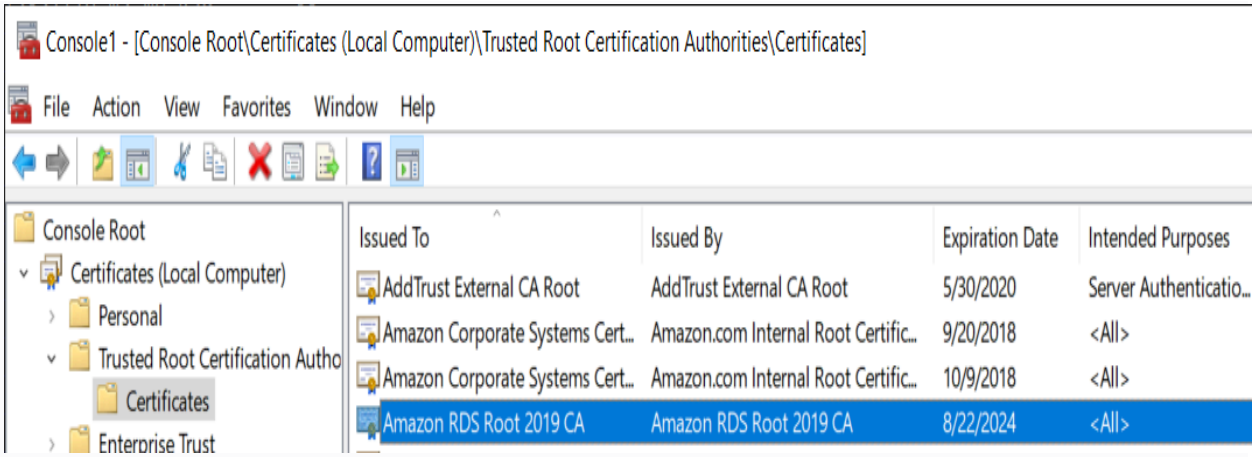
You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS regions for all supported SQL Server editions.

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

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

– Force SSL for all connections — this happens transparently to the client, and the client doesn't have to do any work to use SSL.

– Encrypt specific connections — this sets up an SSL connection from a specific client computer, and you must do work on the client to encrypt connections.



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

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

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

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

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

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

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

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

References:

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

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

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

Check out this Amazon RDS Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Secure Architectures

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

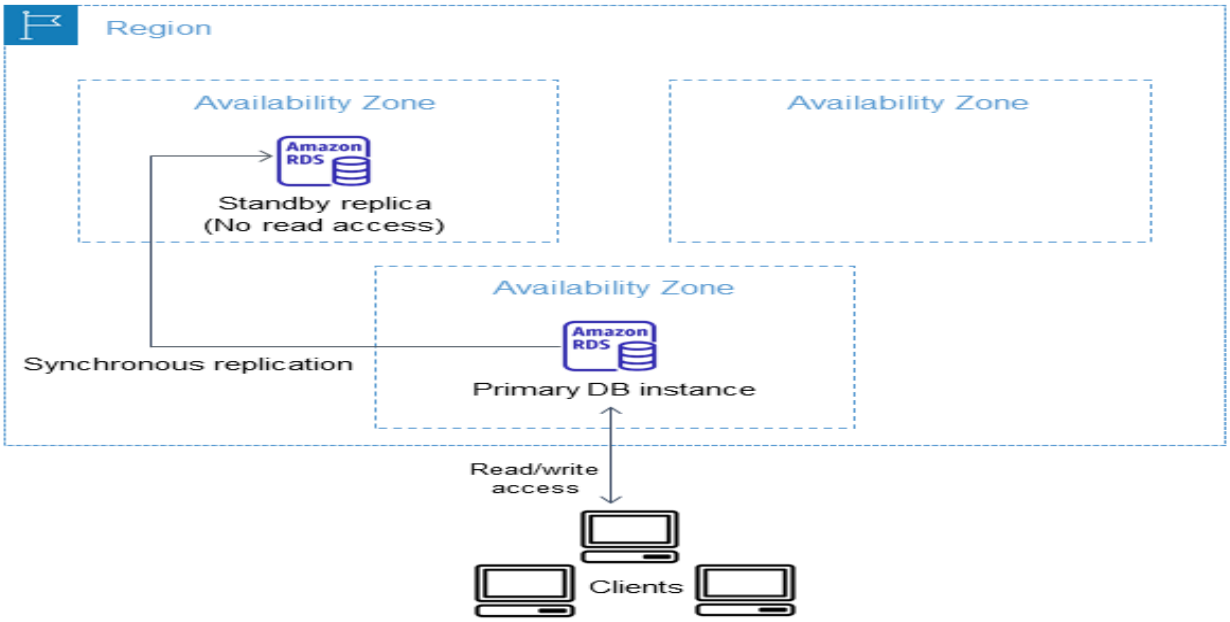
In the event of Read Replica failure
Storage failure on secondary DB instance
Storage failure on primary (Correct)
Compute unit failure on secondary DB instance
Loss of availability in primary Availability Zone (Correct)

Amazon RDS provides high availability and failover support for DB instances using Multi-AZ deployments. Amazon RDS uses several different technologies to provide failover support.

Multi-AZ deployments for Oracle, PostgreSQL, MySQL, and MariaDB DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM).

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

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



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

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

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

Hence, the correct answers are:

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

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

– Storage failure on secondary DB instance

– In the event of Read Replica failure

– Compute unit failure on secondary DB instance

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

7. 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)

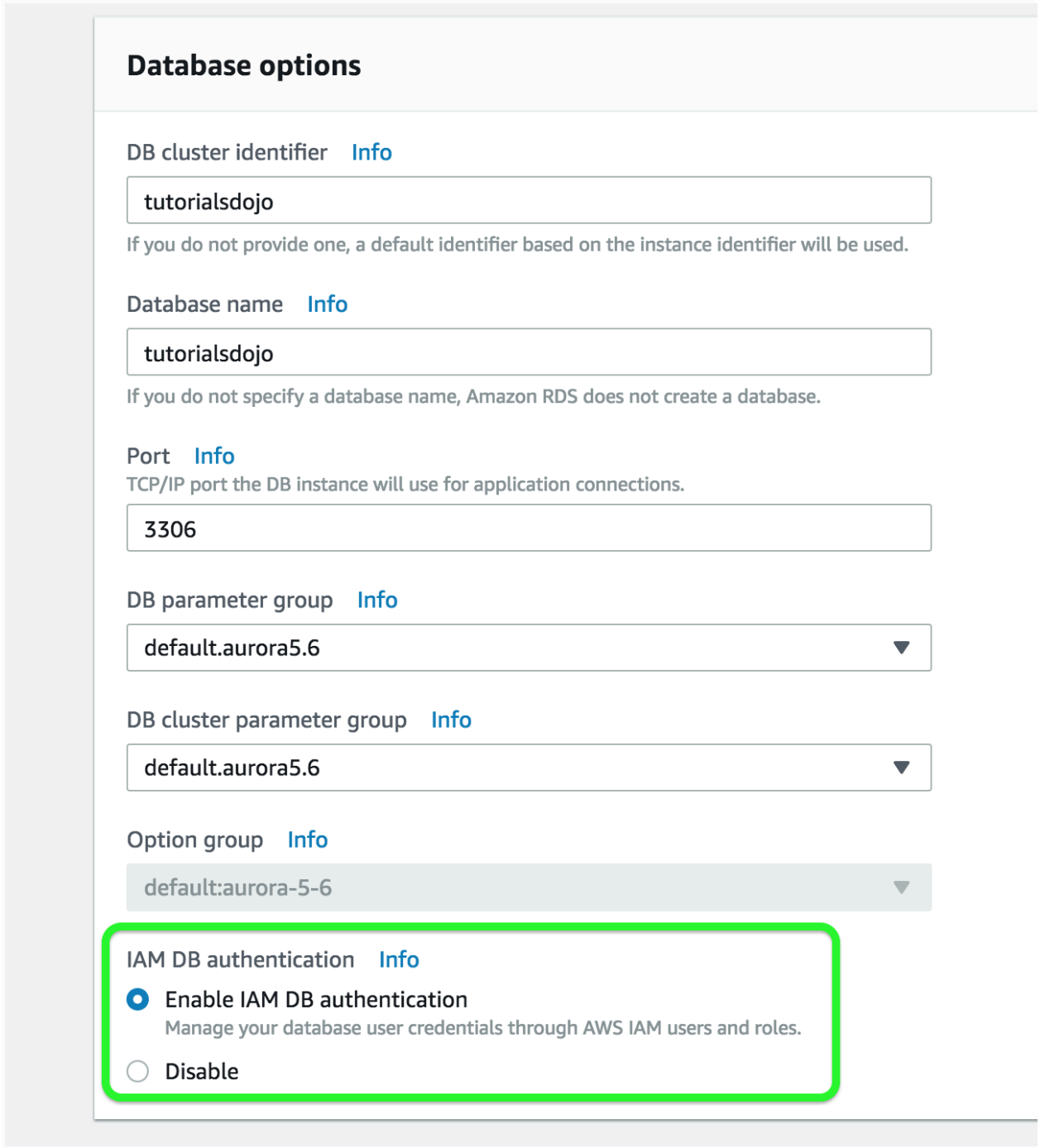
Create an IAM Role and assign it to your EC2 instances which will grant exclusive access to your RDS instance.

Use a combination of IAM and STS to restrict access to your RDS instance via a temporary token.

Configure SSL in your application to encrypt the database connection to RDS.

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/

8. QUESTION

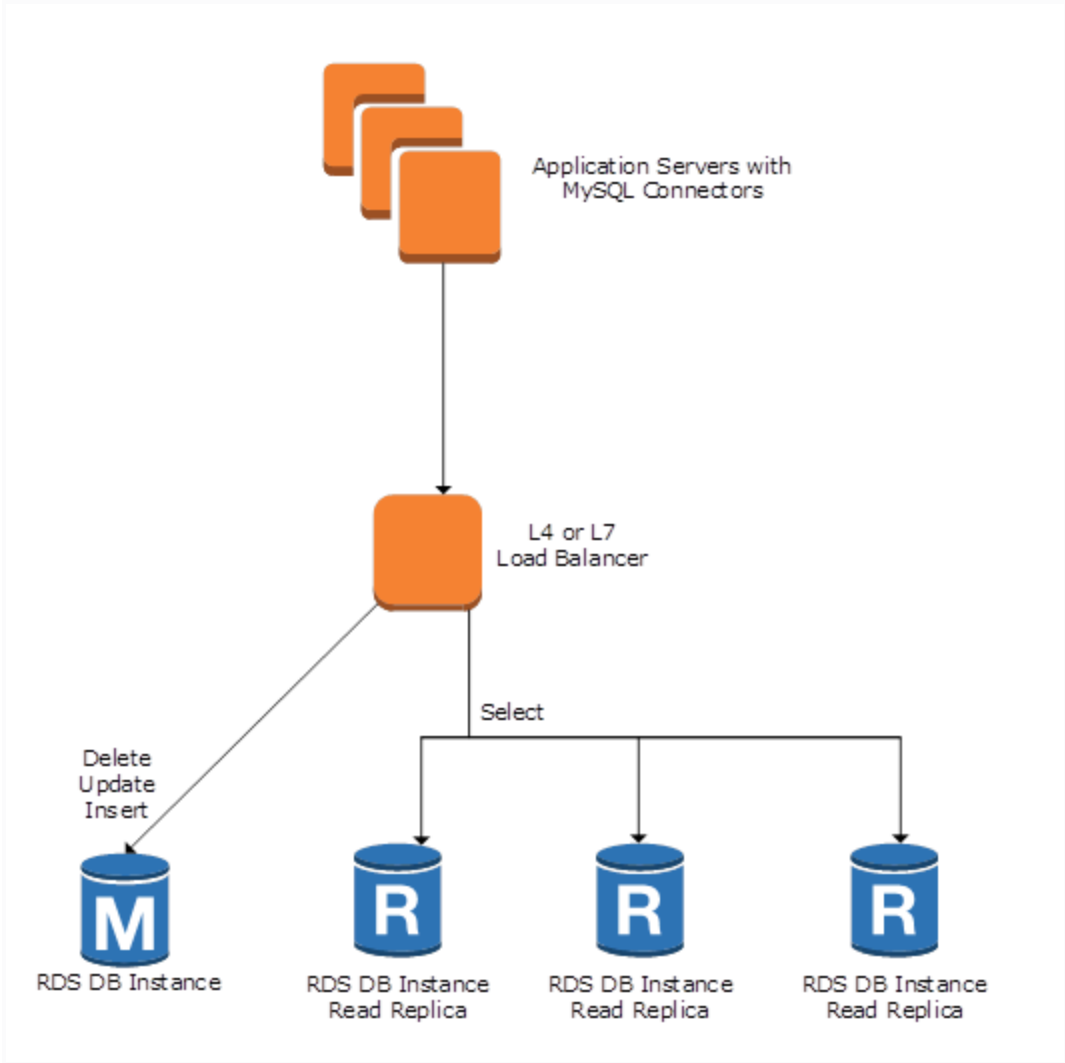
Category: CSAA – Design High-Performing Architectures

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

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

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

Amazon RDS Read Replicas provide enhanced performance and durability for database (DB) instances. This feature makes it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. You can create one or more replicas of a given source DB Instance and serve high-volume application read traffic from multiple copies of your data, thereby increasing aggregate read throughput. Read replicas can also be promoted when needed to become standalone DB instances. Read replicas are available in Amazon RDS for MySQL, MariaDB, Oracle, and PostgreSQL as well as Amazon Aurora.



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

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

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

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

Topic-Based – S3 (SA-Associate)

1. 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?

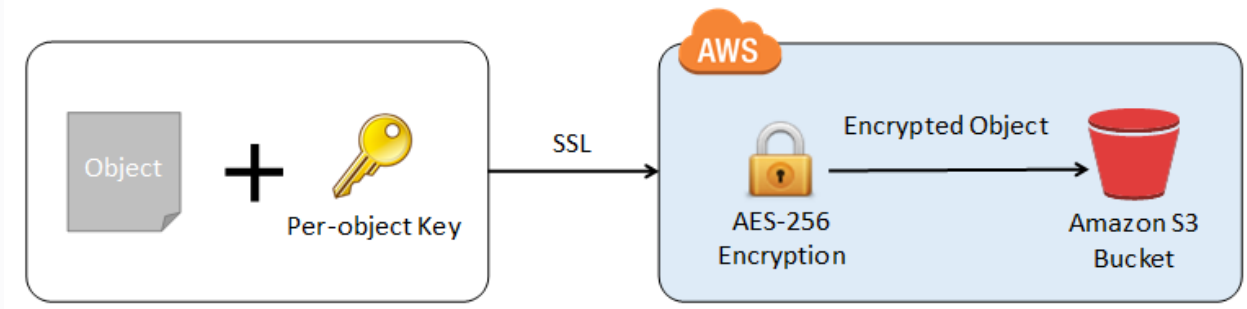
Use S3 client-side encryption with a client-side master key. (Correct)
Use S3 server-side encryption with a KMS managed key.
Use S3 server-side encryption with a customer provided key.
Use S3 client-side encryption with a KMS-managed customer master key.

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.



This is how client-side encryption using client-side master key works:

When uploading an object – You provide a client-side master key to the Amazon S3 encryption client. The client uses the master key only to encrypt the data encryption key that it generates randomly. The process works like this:

1. The Amazon S3 encryption client generates a one-time-use symmetric key (also known as a data encryption key or data key) locally. It uses the data key to encrypt the data of a single Amazon S3 object. The client generates a separate data key for each object.

2. The client encrypts the data encryption key using the master key that you provide. The client uploads the encrypted data key and its material description as part of the object metadata. The client uses the material description to determine which client-side master key to use for decryption.

3. The client uploads the encrypted data to Amazon S3 and saves the encrypted data key as object metadata (`x-amz-meta-x-amz-key`) in Amazon S3.

When downloading an object – The client downloads the encrypted object from Amazon S3. Using the material description from the object's metadata, the client determines which master key to use to decrypt the data key. The client uses that master key to decrypt the data key and then uses the data key to decrypt the object.

Hence, the correct answer is to **use S3 client-side encryption with a client-side master key**.

Using S3 client-side encryption with a KMS-managed customer master key is incorrect because in client-side encryption with a KMS-managed customer master key, you provide an AWS KMS customer master key ID (CMK ID) to AWS. The scenario clearly indicates that both the master keys and the unencrypted data should never be sent to AWS.

Using S3 server-side encryption with a KMS managed key is incorrect because the scenario mentioned that the unencrypted data should never be sent to AWS, which means that you have to use client-side encryption in order to encrypt the data first before sending to AWS. In this way, you can ensure that there is no unencrypted data being uploaded to AWS. In addition, the master key used by Server-Side Encryption with AWS KMS–Managed

Keys (SSE-KMS) is uploaded and managed by AWS, which directly violates the requirement of not uploading the master key.

Using S3 server-side encryption with customer provided key is incorrect because just as mentioned above, you have to use client-side encryption in this scenario instead of server-side encryption. For the S3 server-side encryption with customer-provided key (SSE-C), you actually provide the encryption key as part of your request to upload the object to S3. Using this key, Amazon S3 manages both the encryption (as it writes to disks) and decryption (when you access your objects).

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

2. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

- 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)
- Enable Amazon S3 Intelligent-Tiering

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

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

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

– Change the versioning state of your bucket

– Permanently delete an object version

MFA Delete requires two forms of authentication together:

– Your security credentials

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

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

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

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

Reference:

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

Check out this Amazon S3 Cheat Sheet:

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

3. QUESTION

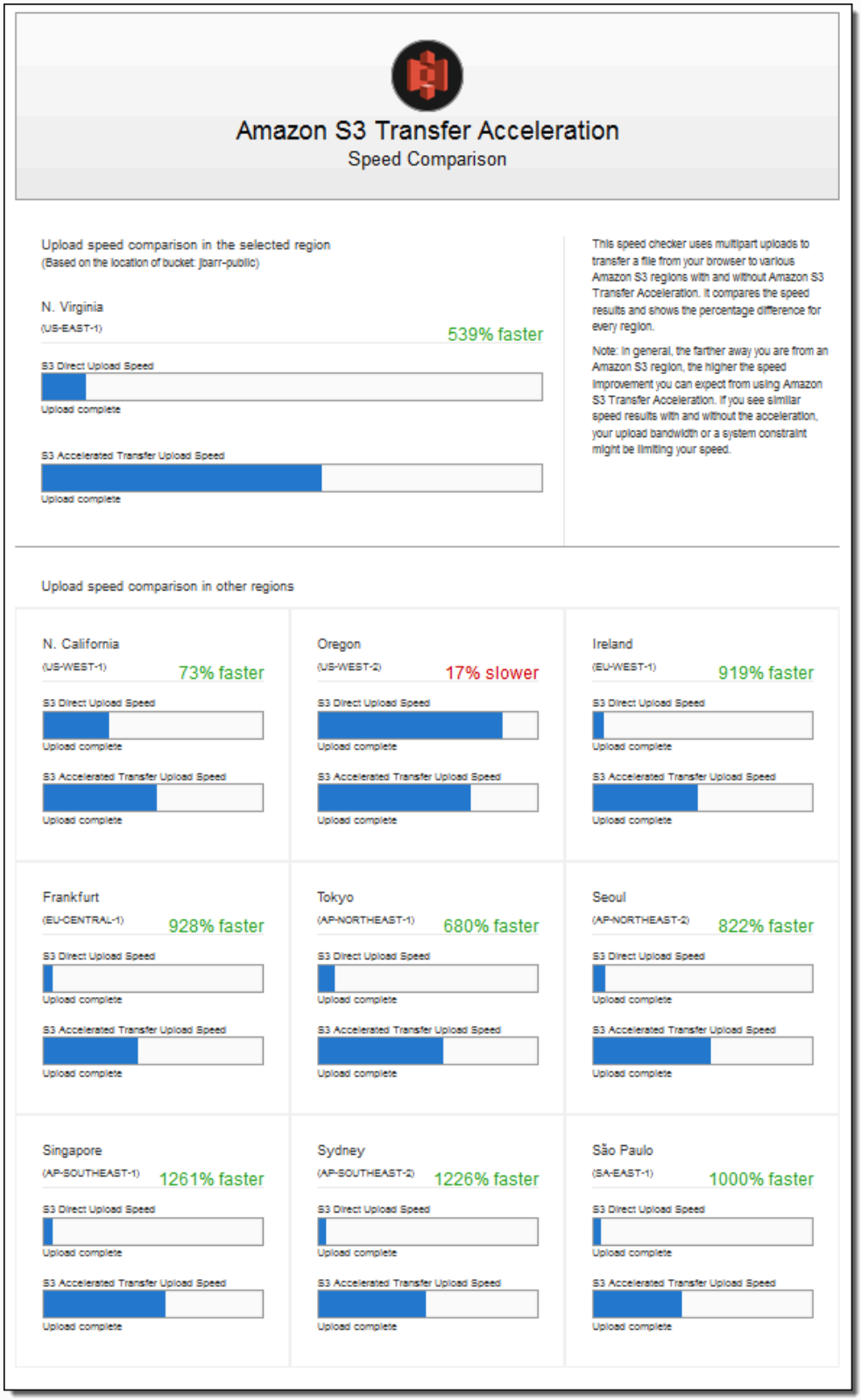
Category: CSAA – Design High-Performing Architectures

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)
- Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket.
- Use AWS Snowball Edge to transfer large amounts of data.
- 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/

4. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A start-up company that offers an intuitive financial data analytics service has consulted you about their AWS architecture. They have a fleet of Amazon EC2 worker instances that process financial data and then outputs reports which are used by their clients. You must store the generated report files in a durable storage. The number of files to be stored can grow over time as the start-up company is expanding rapidly overseas and

hence, they also need a way to distribute the reports faster to clients located across the globe.

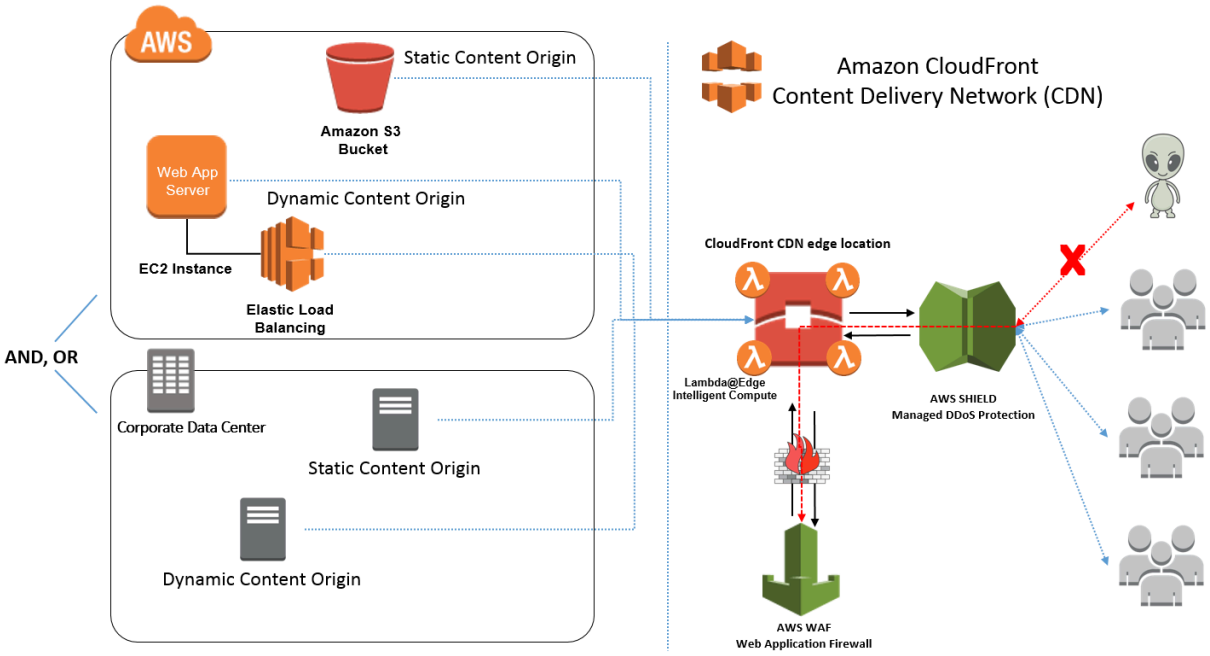
Which of the following is a cost-efficient and scalable storage option that you should use for this scenario?

- Use Amazon Redshift as the data storage and CloudFront as the CDN.
- Use Amazon S3 as the data storage and CloudFront as the CDN. (Correct)
- Use multiple EC2 instance stores for data storage and ElastiCache as the CDN.
- Use Amazon S3 Glacier as the data storage and ElastiCache as the CDN.

A Content Delivery Network (CDN) is a critical component of nearly any modern web application. It used to be that CDN merely improved the delivery of content by replicating commonly requested files (static content) across a globally distributed set of caching servers. However, CDNs have become much more useful over time.

For caching, a CDN will reduce the load on an application origin and improve the experience of the requestor by delivering a local copy of the content from a nearby cache edge, or Point of Presence (PoP). The application origin is off the hook for opening the connection and delivering the content directly as the CDN takes care of the heavy lifting. The end result is that the application origins don’t need to scale to meet demands for static

content.



Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.

Amazon S3 offers a highly durable, scalable, and secure destination for backing up and archiving your critical data. This is the correct option as the start-up company is looking for a durable storage to store the audio and text files. In addition, ElastiCache is only used for caching and not specifically as a Global Content Delivery Network (CDN).

Using Amazon Redshift as the data storage and CloudFront as the CDN is incorrect as Amazon Redshift is usually used as a Data Warehouse.

Using Amazon S3 Glacier as the data storage and ElastiCache as the CDN is incorrect as Amazon S3 Glacier is usually used for data archives.

Using multiple EC2 instance stores for data storage and ElastiCache as the CDN is incorrect as data stored in an instance store is not durable.

References:

https://aws.amazon.com/s3/

https://aws.amazon.com/caching/cdn/

Check out this Amazon S3 Cheat Sheet:

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

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

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

5. QUESTION

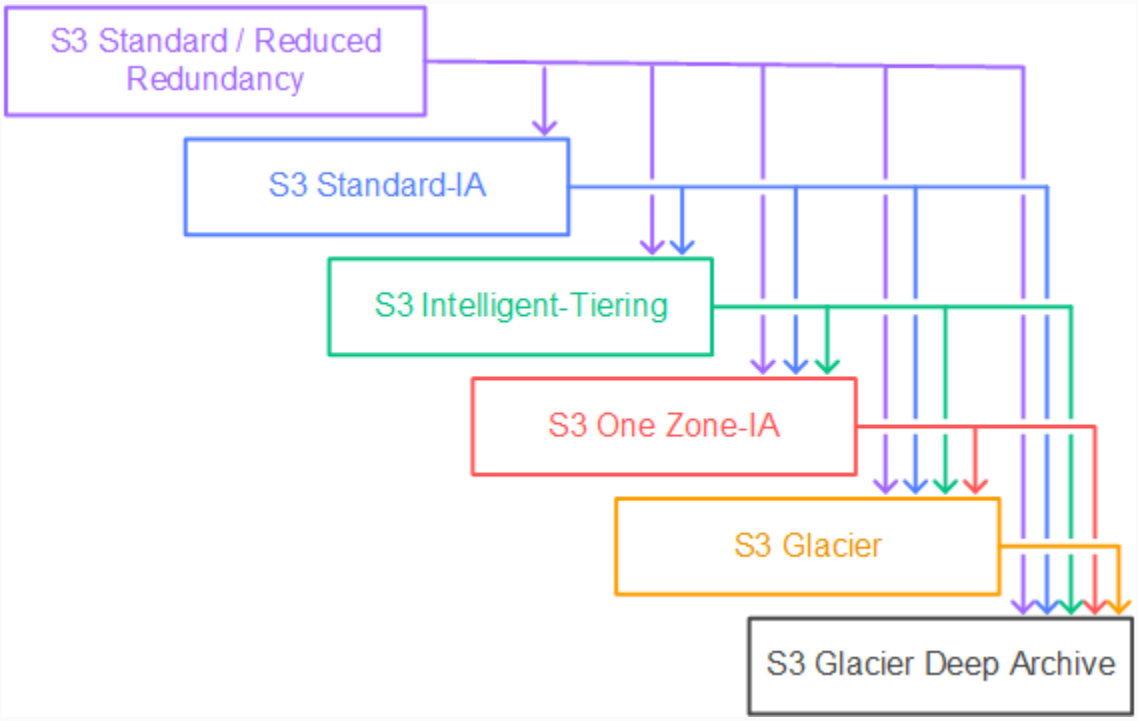
Category: CSAA – Design Cost-Optimized Architectures

There are a few, easily reproducible but confidential files that your client wants to store in AWS without worrying about storage capacity. For the first month, all of these files will be accessed frequently but after that, they will rarely be accessed at all. The old files will only be accessed by developers so there is no set retrieval time requirement. However, the files under a specific `tdojo-finance` prefix in the S3 bucket will be used for post-processing that requires millisecond retrieval time.

Given these conditions, which of the following options would be the most cost-effective solution for your client's storage needs?

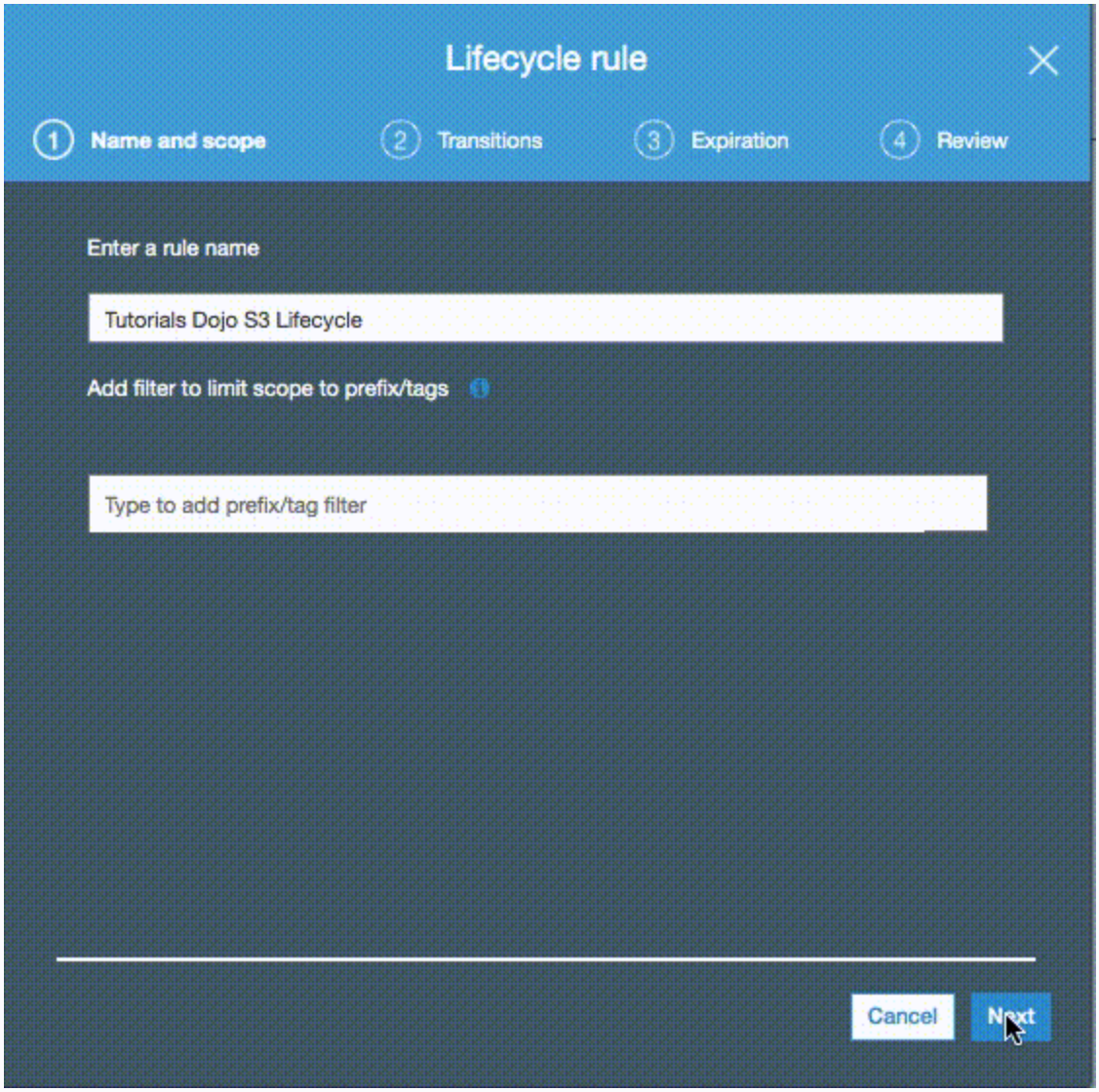
Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy. (Correct)
Store the files in S3 then after a month, change the storage class of the bucket to Intelligent-Tiering using lifecycle policy.
Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to S3-IA while the remaining go to Glacier using lifecycle policy.
Store the files in S3 then after a month, change the storage class of the bucket to S3-IA using lifecycle policy.

Initially, the files will be accessed frequently, and S3 is a durable and highly available storage solution for that. After a month has passed, the files won't be accessed frequently anymore, so it is a good idea to use lifecycle policies to move them to a storage class that would have a lower cost for storing them.



Since the files are easily reproducible and some of them are needed to be retrieved quickly based on a specific prefix filter (`tdojo-finance`), S3-One Zone IA would be a good choice for storing them. The other files that do not contain such prefix would then be moved to Glacier for low-cost archival. This setup would also be the most cost-effective for the client.

Hence, the correct answer is: **Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy**.



The option that says: **Storing the files in S3 then after a month, changing the storage class of the bucket to S3-IA using lifecycle policy** is incorrect. Although it is valid to move the files to S3-IA, this solution still costs more compared with using a combination of S3-One Zone IA and Glacier.

The option that says: **Storing the files in S3 then after a month, changing the storage class of the bucket to Intelligent-Tiering using lifecycle policy** is incorrect. While S3 Intelligent-Tiering can automatically move data between two access tiers (frequent access and infrequent access) when access patterns change, it is more suitable for scenarios where you don't know the access patterns of your data. It may take some time for S3 Intelligent-Tiering to analyze the access patterns before it moves the data to a cheaper storage class like S3-IA which means you may still end up paying more in the beginning. In addition, you already know the access patterns of the files which means you can directly change the storage class immediately and save cost right away.

The option that says: **Storing the files in S3 then after a month, changing the storage class of the `tdojo-finance` prefix to S3-IA while the remaining go to Glacier using lifecycle policy** is incorrect. Even though S3-IA costs less than the S3 Standard storage class, it is still more expensive than S3-One Zone IA. Remember that the files are easily reproducible so you can safely move the data to S3-One Zone IA and in case there is an outage, you can simply generate the missing data again.

References:

https://aws.amazon.com/blogs/compute/amazon-s3-adds-prefix-and-suffix-filters-for-lambda-function-triggering

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

https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-configuration-examples.html

https://aws.amazon.com/s3/pricing

Check out this Amazon S3 Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Secure Architectures

For data privacy, a healthcare company has been asked to comply with the Health Insurance Portability and Accountability Act (HIPAA). The company stores all its backups on an Amazon S3 bucket. It is required that data stored on the S3 bucket must be encrypted.

What is the best option to do this? (Select TWO.)

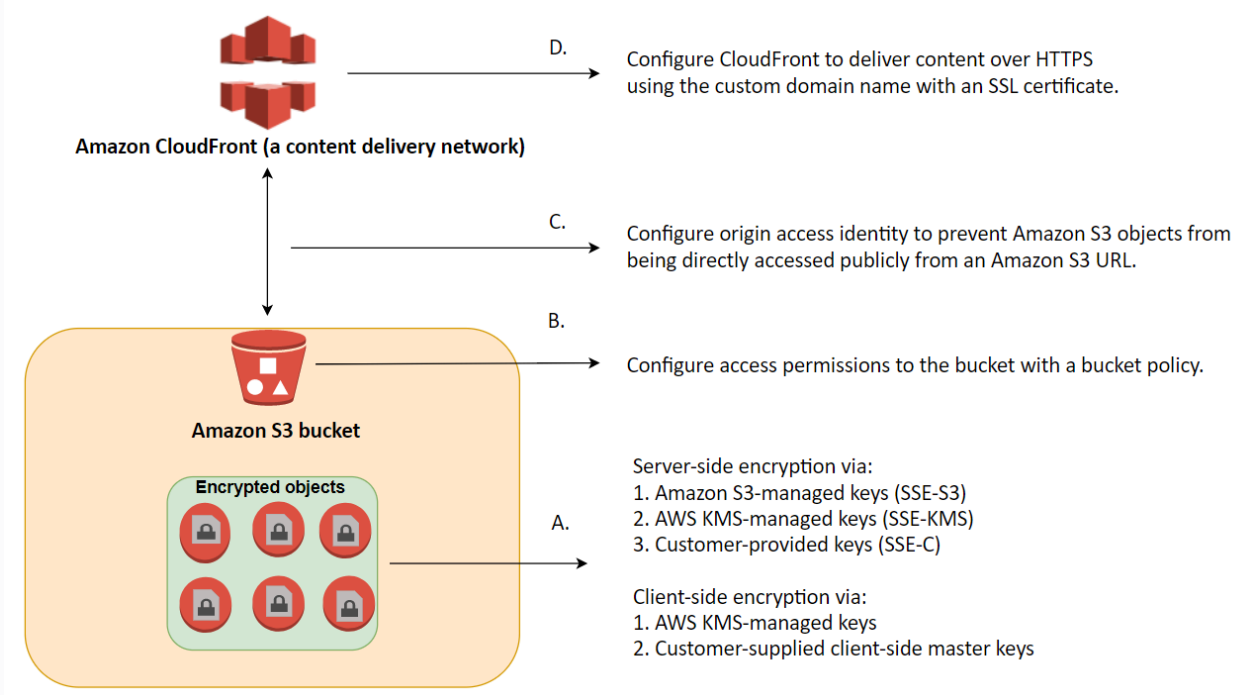
Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption. (Correct)
Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys. (Correct)

Store the data on EBS volumes with encryption enabled instead of using Amazon S3.

Store the data in encrypted EBS snapshots.

Enable Server-Side Encryption on an S3 bucket to make use of AES-128 encryption.

Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a pre-signed URL, that URL works the same way for both encrypted and unencrypted objects.



You have three mutually exclusive options depending on how you choose to manage the encryption keys:

1. Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
2. Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
3. Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

The options that say: **Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys** and **Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption** are correct because these options are using client-side encryption and Amazon S3-Managed Keys (SSE-S3) respectively. *Client-side encryption* is the act of encrypting data before sending it to Amazon S3 while SSE-S3 uses AES-256 encryption.

Storing the data on EBS volumes with encryption enabled instead of using Amazon S3 and **storing the data in encrypted EBS snapshots** are incorrect because both options use EBS encryption and not S3.

Enabling Server-Side Encryption on an S3 bucket to make use of AES-128 encryption is incorrect as S3 doesn't provide AES-128 encryption, only AES-256.

References:

http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

Check out this Amazon S3 Cheat Sheet:

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

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

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

7. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

- It provides high latency and low throughput performance.
- Ideal to use for data archiving.

It is designed for data that is accessed less frequently. (Correct)

It is designed for data that requires rapid access when needed. (Correct)

It automatically moves data to the most cost-effective access tier without any operational overhead.

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

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

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

Key Features:

– Same low latency and high throughput performance of Standard

– Designed for durability of 99.999999999% of objects

– Designed for 99.9% availability over a given year

– Backed with the Amazon S3 Service Level Agreement for availability

– Supports SSL encryption of data in transit and at rest

– Lifecycle management for automatic migration of objects

Hence, the correct answers are:

– **It is designed for data that is accessed less frequently.**

– **It is designed for data that requires rapid access when needed.**

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

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

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

References:

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

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

Check out this Amazon S3 Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

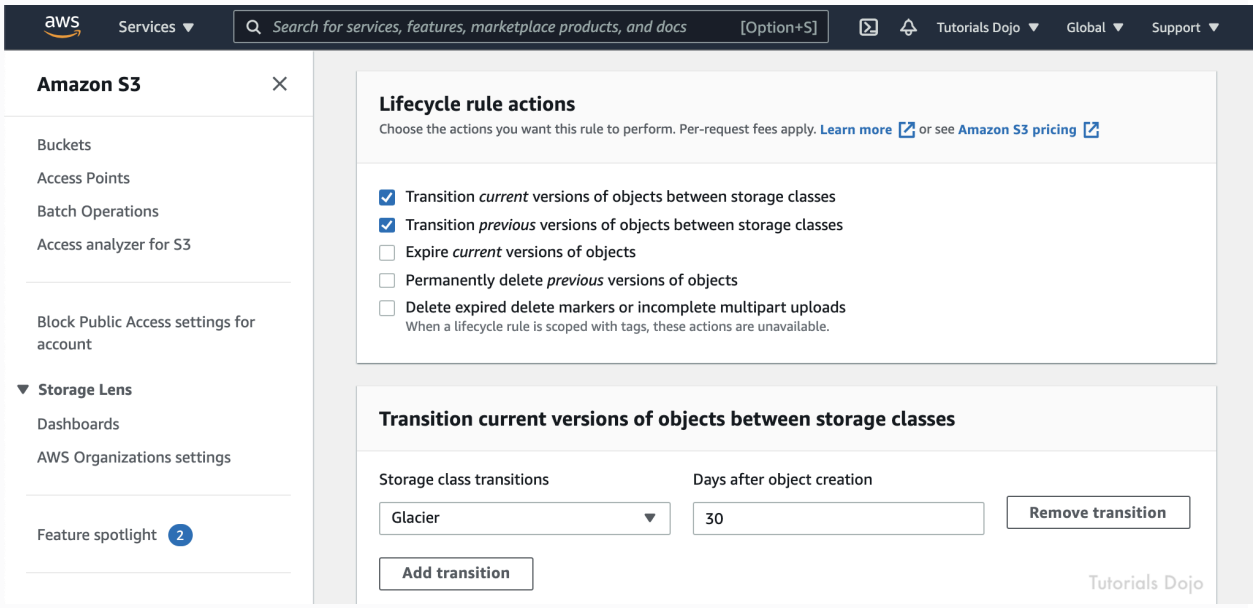
A Solutions Architect is working for a financial company. The manager wants to have the ability to automatically transfer obsolete data from their S3 bucket to a low-cost storage system in AWS after a certain period of time.

What is the best solution that the Architect can provide to them?

Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier.
Use Amazon SQS.
Use Lifecycle Policies in S3 to move obsolete data to Glacier. (Correct)
Use Amazon Timestream.

In this scenario, you can use lifecycle policies in S3 to automatically move obsolete data to Glacier.

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



These actions can be classified as follows:

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

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

The option that says: **Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier** is incorrect because you don't need to create a scheduled job in EC2 as you can simply use the lifecycle policy in S3.

The option that says: **Use Amazon SQS** is incorrect as SQS is not a storage service. Amazon SQS is primarily used to decouple your applications by queueing the incoming requests of your application.

The option that says: **Use Amazon Timestream** is incorrect. While Amazon Timestream is great for storing and analyzing time-series data, it doesn't directly address the requirement of moving data from S3 to a lower-cost storage option based on the age of the data. The best solution for this specific use case would be to use Lifecycle Policies in S3 to move obsolete data to Glacier, which is a low-cost storage service in AWS. This can be done by setting up some rules (e.g., which folder) and it will transition the data.

References:

http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

https://aws.amazon.com/blogs/aws/archive-s3-to-glacier/

Check out this Amazon S3 Cheat Sheet:

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

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

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

Topic-Based – SQS (SA-Associate)

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A software company has resources hosted in AWS and on-premises servers. You have been requested to create a decoupled architecture for applications which make use of both resources.

Which of the following options are valid? (Select TWO.)

- Use RDS to utilize both on-premises servers and EC2 instances for your decoupled application
- Use SQS to utilize both on-premises servers and EC2 instances for your decoupled application (Correct)
- Use DynamoDB to utilize both on-premises servers and EC2 instances for your decoupled application
- Use SWF to utilize both on-premises servers and EC2 instances for your decoupled application (Correct)
- Use VPC peering to connect both on-premises servers and EC2 instances for your decoupled application

Amazon Simple Queue Service (SQS) and Amazon Simple Workflow Service (SWF) are the services that you can use for creating a decoupled architecture in AWS. Decoupled architecture is a type of computing architecture that enables computing components or layers to execute independently while still interfacing with each other.

Amazon SQS offers reliable, highly-scalable hosted queues for storing messages while they travel between applications or microservices. Amazon SQS lets you move data between distributed application components and helps you decouple these components. Amazon SWF is a web service that makes it easy to coordinate work across distributed application components.

Using RDS to utilize both on-premises servers and EC2 instances for your decoupled application and **using DynamoDB to utilize both on-premises servers and EC2**

instances for your decoupled application are incorrect as RDS and DynamoDB are database services.

Using VPC peering to connect both on-premises servers and EC2 instances for your decoupled application is incorrect because you can't create a VPC peering for your on-premises network and AWS VPC.

References:

https://aws.amazon.com/sqs/

http://docs.aws.amazon.com/amazonswf/latest/developerguide/swf-welcome.html

Check out this Amazon SQS Cheat Sheet:

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

Amazon Simple Workflow (SWF) vs AWS Step Functions vs Amazon SQS:

https://tutorialsdojo.com/amazon-simple-workflow-swf-vs-aws-step-functions-vs-amazon-sqs/

Comparison of AWS Services Cheat Sheets:

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

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?

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)

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 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.

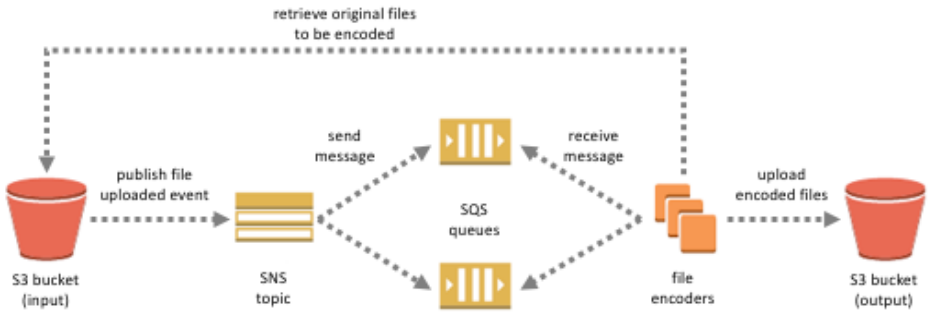
Amazon S3 supports the following destinations where it can publish events:

– 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

Check out this Amazon S3 Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design High-Performing Architectures

The start-up company that you are working for has a batch job application that is currently hosted on an EC2 instance. It is set to process messages from a queue created in SQS with default settings. You configured the application to process the messages once a week. After 2 weeks, you noticed that not all messages are being processed by the application.

What is the root cause of this issue?

Missing permissions in SQS.
The SQS queue is set to short-polling.
Amazon SQS has automatically deleted the messages that have been in a queue for more than the maximum message retention period. (Correct)
The batch job application is configured to long polling.

Amazon SQS automatically deletes messages that have been in a queue for more than the maximum message retention period. The default message retention period is 4 days. Since the queue is configured to the default settings and the batch job application only processes the messages once a week, the messages that are in the queue for more than 4 days are deleted. This is the root cause of the issue.

To fix this, you can increase the message retention period to a maximum of 14 days using the SetQueueAttributes action.

References:

https://aws.amazon.com/sqs/faqs/

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-message-lifecycle.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sq

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A company has a web-based ticketing service that utilizes Amazon SQS and a fleet of EC2 instances. The EC2 instances that consume messages from the SQS queue are configured to poll the queue as often as possible to keep end-to-end throughput as high as possible. The Solutions Architect noticed that polling the queue in tight loops is using unnecessary CPU cycles, resulting in increased operational costs due to empty responses.

In this scenario, what should the Solutions Architect do to make the system more cost-effective?

Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero. (Correct)
Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to zero.
Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to zero.
Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero.

In this scenario, the application is deployed in a fleet of EC2 instances that are polling messages from a single SQS queue. Amazon SQS uses short polling by default, querying only a subset of the servers (based on a weighted random distribution) to determine whether any messages are available for inclusion in the response. Short polling works for scenarios that require higher throughput. However, you can also configure the queue to use Long polling instead, to reduce cost.

The ReceiveMessageWaitTimeSeconds is the queue attribute that determines whether you are using Short or Long polling. By default, its value is zero which means it is using Short polling. If it is set to a value greater than zero, then it is Long polling.

Hence, **configuring Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero is the correct answer.**

Quick facts about SQS Long Polling:

– Long polling helps reduce your cost of using Amazon SQS by reducing the number of empty responses when there are no messages available to return in reply to a *ReceiveMessage* request sent to an Amazon SQS queue and eliminating false empty responses when messages are available in the queue but aren't included in the response.

– Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the response to the `ReceiveMessage` request contains at least one of the available messages, up to the maximum number of messages specified in the `ReceiveMessage` action.

– Long polling eliminates false empty responses by querying all (rather than a limited number) of the servers. Long polling returns messages as soon any message becomes available.

Reference:

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-long-polling.html

Check out this Amazon SQS Cheat Sheet:

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

5. QUESTION

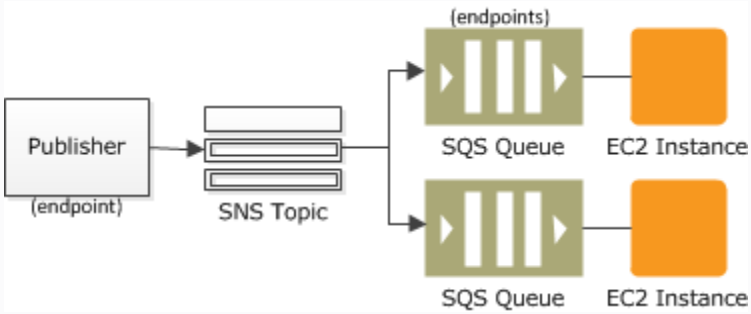
Category: CSAA – Design High-Performing Architectures

An e-commerce application is using a fanout messaging pattern for its order management system. For every order, it sends an Amazon SNS message to an SNS topic, and the message is replicated and pushed to multiple Amazon SQS queues for parallel asynchronous processing. A Spot EC2 instance retrieves the message from each SQS queue and processes the message. There was an incident that while an EC2 instance is currently processing a message, the instance was abruptly terminated, and the processing was not completed in time.

In this scenario, what happens to the SQS message?

The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout.
When the message visibility timeout expires, the message becomes available for processing by other EC2 instances (Correct)
The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online.
The message will be sent to a Dead Letter Queue in AWS DataSync.

A “fanout” pattern is when an Amazon SNS message is sent to a topic and then replicated and pushed to multiple Amazon SQS queues, HTTP endpoints, or email addresses. This allows for parallel asynchronous processing. For example, you could develop an application that sends an Amazon SNS message to a topic whenever an order is placed for a product. Then, the Amazon SQS queues that are subscribed to that topic would receive identical notifications for the new order. The Amazon EC2 server instance attached to one of the queues could handle the processing or fulfillment of the order, while the other server instance could be attached to a data warehouse for analysis of all orders received.



When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS

doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a *visibility timeout*, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

The option that says: **The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout** is incorrect because the message will not be automatically assigned to the same EC2 instance once it is abruptly terminated. When the message visibility timeout expires, the message becomes available for processing by other EC2 instances.

The option that says: **The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online** is incorrect because the message will not be deleted and won't be duplicated in the SQS queue when the EC2 instance comes online.

The option that says: **The message will be sent to a Dead Letter Queue in AWS DataSync** is incorrect because although the message could be programmatically sent to a Dead Letter Queue (DLQ), it won't be handled by AWS DataSync but by Amazon SQS instead. AWS DataSync is primarily used to simplify your migration with AWS. It makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3 or Amazon Elastic File System (Amazon EFS).

References:

http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

https://docs.aws.amazon.com/sns/latest/dg/sns-common-scenarios.html

Check out this Amazon SQS Cheat Sheet:

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

6. QUESTION

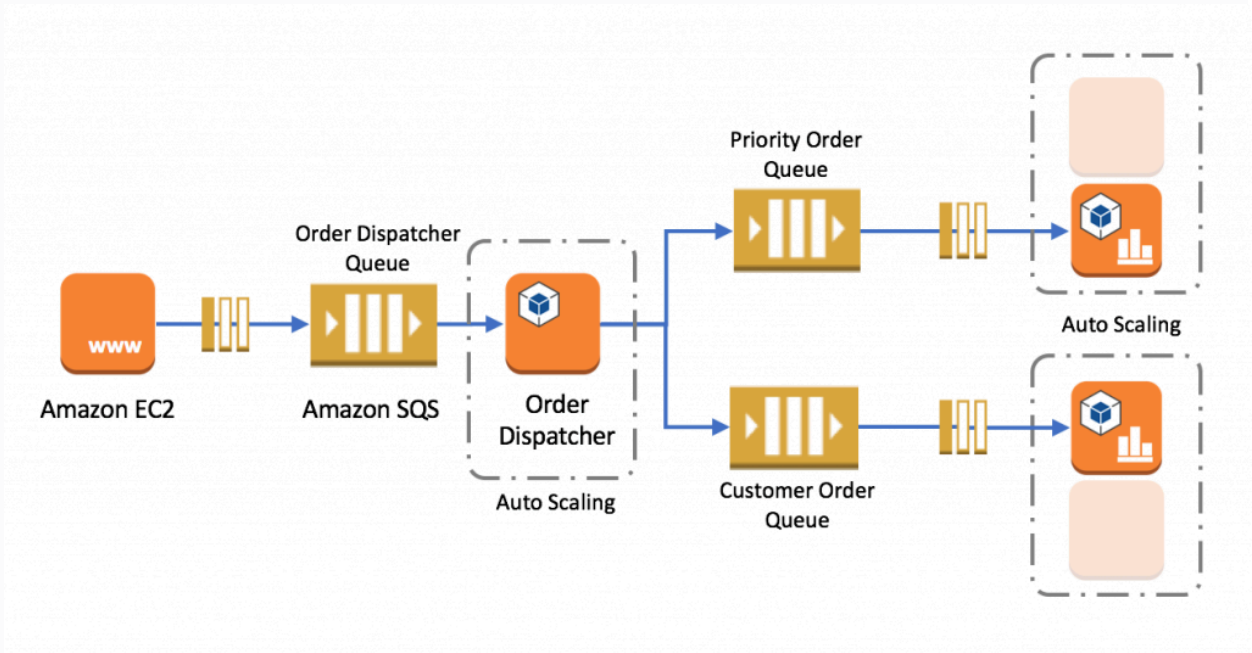
Category: CSAA – Design High-Performing Architectures

A company launched a website that accepts high-quality photos and turns them into a downloadable video montage. The website offers a free and a premium account that guarantees faster processing. All requests by both free and premium members go through a single SQS queue and then processed by a group of EC2 instances that generate the videos. The company needs to ensure that the premium users who paid for the service have higher priority than the free members.

How should the company re-design its architecture to address this requirement?

- For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members.
- Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue. (Correct)
- Use Amazon Kinesis to process the photos and generate the video montage in real-time.
- Use Amazon S3 to store and process the photos and then generate the video montage afterward.

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume without losing messages or requiring other services to be available.



In this scenario, it is best to create 2 separate SQS queues for each type of member. The SQS queues for the premium members can be polled first by the EC2 Instances and once completed, the messages from the free members can be processed next.

Hence, the correct answer is: **Create an SQS queue for free members and another one for premium members. Configure your EC2 instances to consume messages from the premium queue first and if it is empty, poll from the free members' SQS queue.**

The option that says: **For the requests made by premium members, set a higher priority in the SQS queue so it will be processed first compared to the requests made by free members** is incorrect as you cannot set a priority to individual items in the SQS queue.

The option that says: **Using Amazon Kinesis to process the photos and generate the video montage in real time** is incorrect as Amazon Kinesis is used to process streaming data and it is not applicable in this scenario.

The option that says: **Using Amazon S3 to store and process the photos and then generating the video montage afterward** is incorrect as Amazon S3 is used for durable storage and not for processing data.

Reference:

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-best-practices.html

Check out this Amazon SQS Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design High-Performing Architectures

An insurance company plans to implement a message filtering feature in their web application. To implement this solution, they need to create separate Amazon SQS queues for each type of quote request. The entire message processing should not exceed 24 hours.

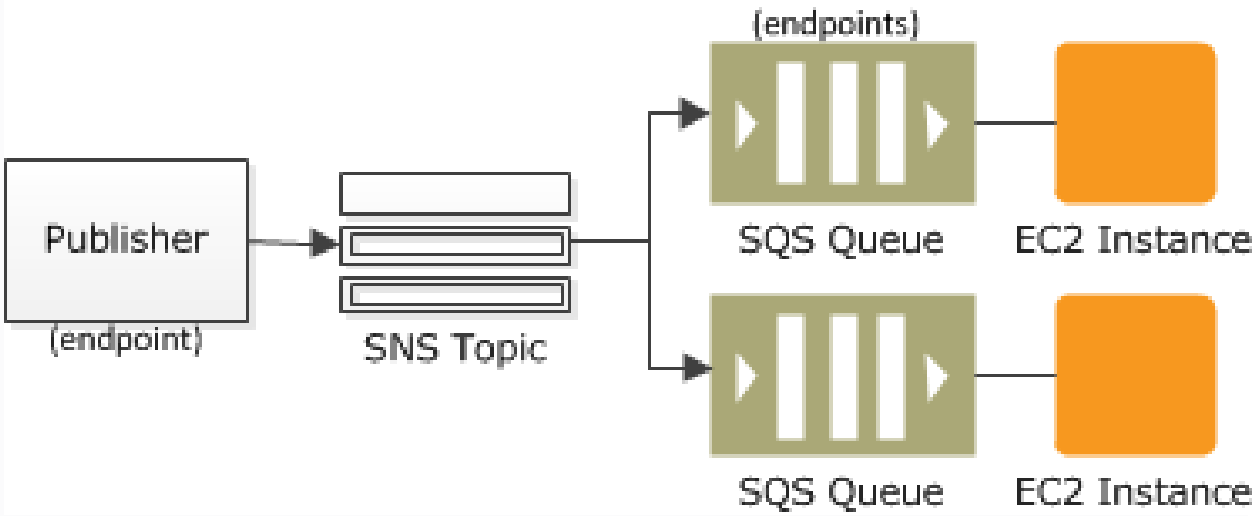
As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

- Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type. (Correct)
- Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type.
- Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type.
- Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type.

Amazon SNS is a fully managed pub/sub messaging service. With Amazon SNS, you can use topics to simultaneously distribute messages to multiple subscribing endpoints such as Amazon SQS queues, AWS Lambda functions, HTTP endpoints, email addresses, and mobile devices (SMS, Push).

Amazon SQS is a message queue service used by distributed applications to exchange messages through a polling model. It can be used to decouple sending and receiving components without requiring each component to be concurrently available.

A fanout scenario occurs when a message published to an SNS topic is replicated and pushed to multiple endpoints, such as Amazon SQS queues, HTTP(S) endpoints, and Lambda functions. This allows for parallel asynchronous processing.



For example, you can develop an application that publishes a message to an SNS topic whenever an order is placed for a product. Then, two or more SQS queues that are subscribed to the SNS topic receive identical notifications for the new order. An Amazon Elastic Compute Cloud (Amazon EC2) server instance attached to one of the SQS queues can handle the processing or fulfillment of the order. And you can attach another Amazon EC2 server instance to a data warehouse for analysis of all orders received.

By default, an Amazon SNS topic subscriber receives every message published to the topic. You can use Amazon SNS message filtering to assign a filter policy to the topic subscription, and the subscriber will only receive a message that they are interested in. Using Amazon SNS and Amazon SQS together, messages can be delivered to applications that require immediate notification of an event. This method is known as fanout to Amazon SQS queues.

Hence, the correct answer is: **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Set the filter policies in the SNS subscriptions to publish the message to the designated SQS queue based on its quote request type.**

The option that says: **Create one Amazon SNS topic and configure the Amazon SQS queues to subscribe to the SNS topic. Publish the same messages to all SQS queues. Filter the messages in each queue based on the quote request type** is incorrect because this option will distribute the same messages on all SQS queues instead of its

designated queue. You need to fan-out the messages to multiple SQS queues using a filter policy in Amazon SNS subscriptions to allow parallel asynchronous processing. By doing so, the entire message processing will not exceed 24 hours.

The option that says: **Create multiple Amazon SNS topics and configure the Amazon SQS queues to subscribe to the SNS topics. Publish the message to the designated SQS queue based on the quote request type** is incorrect because to implement the solution asked in the scenario, you only need to use one Amazon SNS topic. To publish it to the designated SQS queue, you must set a filter policy that allows you to fanout the messages. If you didn't set a filter policy in Amazon SNS, the subscribers would receive all the messages published to the SNS topic. Thus, using multiple SNS topics is not an appropriate solution for this scenario.

The option that says: **Create a data stream in Amazon Kinesis Data Streams. Use the Amazon Kinesis Client Library to deliver all the records to the designated SQS queues based on the quote request type** is incorrect because Amazon KDS is not a message filtering service. You should use Amazon SNS and SQS to distribute the topic to the designated queue.

References:

https://aws.amazon.com/getting-started/hands-on/filter-messages-published-to-topics/

https://docs.aws.amazon.com/sns/latest/dg/sns-message-filtering.html

https://docs.aws.amazon.com/sns/latest/dg/sns-sqs-as-subscriber.html

Check out these Amazon SNS and SQS Cheat Sheets:

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

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

8. QUESTION

Category: CSAA – Design Resilient Architectures

An investment bank has a distributed batch processing application which is hosted in an Auto Scaling group of Spot EC2 instances with an SQS queue. You configured your components to use client-side buffering so that the calls made from the client will be buffered first and then sent as a batch request to SQS.

What is a period of time during which the SQS queue prevents other consuming components from receiving and processing a message?

Visibility Timeout (Correct)
Component Timeout
Processing Timeout
Receiving Timeout

The visibility timeout is a period of time during which Amazon SQS prevents other consuming components from receiving and processing a message.

When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a *visibility timeout*, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

References:

https://aws.amazon.com/sqs/faqs/

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

Check out this Amazon SQS Cheat Sheet:

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

Topic-Based – VPC (SA-Associate)

1. QUESTION

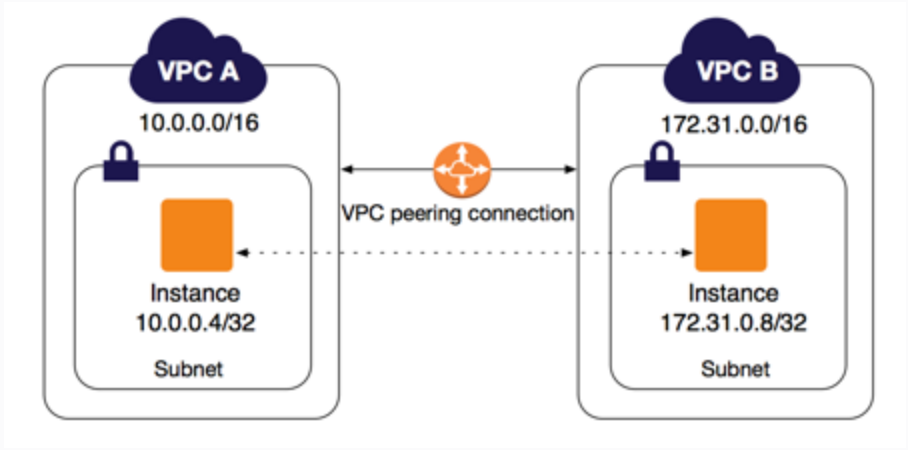
Category: CSAA – Design Resilient Architectures

A large insurance company has an AWS account that contains three VPCs (DEV, UAT and PROD) in the same region. UAT is peered to both PROD and DEV using a VPC peering connection. All VPCs have non-overlapping CIDR blocks. The company wants to push minor code releases from Dev to Prod to speed up time to market.

Which of the following options helps the company accomplish this?

Do nothing. Since these two VPCs are already connected via UAT, they already have a connection to each other.
Change the DEV and PROD VPCs to have overlapping CIDR blocks to be able to connect them.
Create a new entry to PROD in the DEV route table using the VPC peering connection as the target.
Create a new VPC peering connection between PROD and DEV with the appropriate routes. (Correct)

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.



AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

Creating a new entry to PROD in the DEV route table using the VPC peering connection as the target is incorrect because even if you configure the route tables, the two VPCs will still be disconnected until you set up a VPC peering connection between them.

Changing the DEV and PROD VPCs to have overlapping CIDR blocks to be able to connect them is incorrect because you cannot peer two VPCs with overlapping CIDR blocks.

The option that says: **Do nothing. Since these two VPCs are already connected via UAT, they already have a connection to each other** is incorrect as transitive VPC peering is not allowed hence, even though DEV and PROD are both connected in UAT, these two VPCs do not have a direct connection to each other.

Reference:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-peering.html

Check out these Amazon VPC and VPC Peering Cheat Sheets:

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

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

2. QUESTION

Category: CSAA – Design Secure Architectures

An insurance company utilizes SAP HANA for its day-to-day ERP operations. Since they can't migrate this database due to customer preferences, they need to integrate it with the current AWS workload in the VPC in which they are required to establish a site-to-site VPN connection.

What needs to be configured outside of the VPC for them to have a successful site-to-site VPN connection?

An EIP to the Virtual Private Gateway
An Internet-routable IP address (static) of the customer gateway's external interface for the on-premises network (Correct)
The main route table in your VPC to route traffic through a NAT instance
A dedicated NAT instance in a public subnet

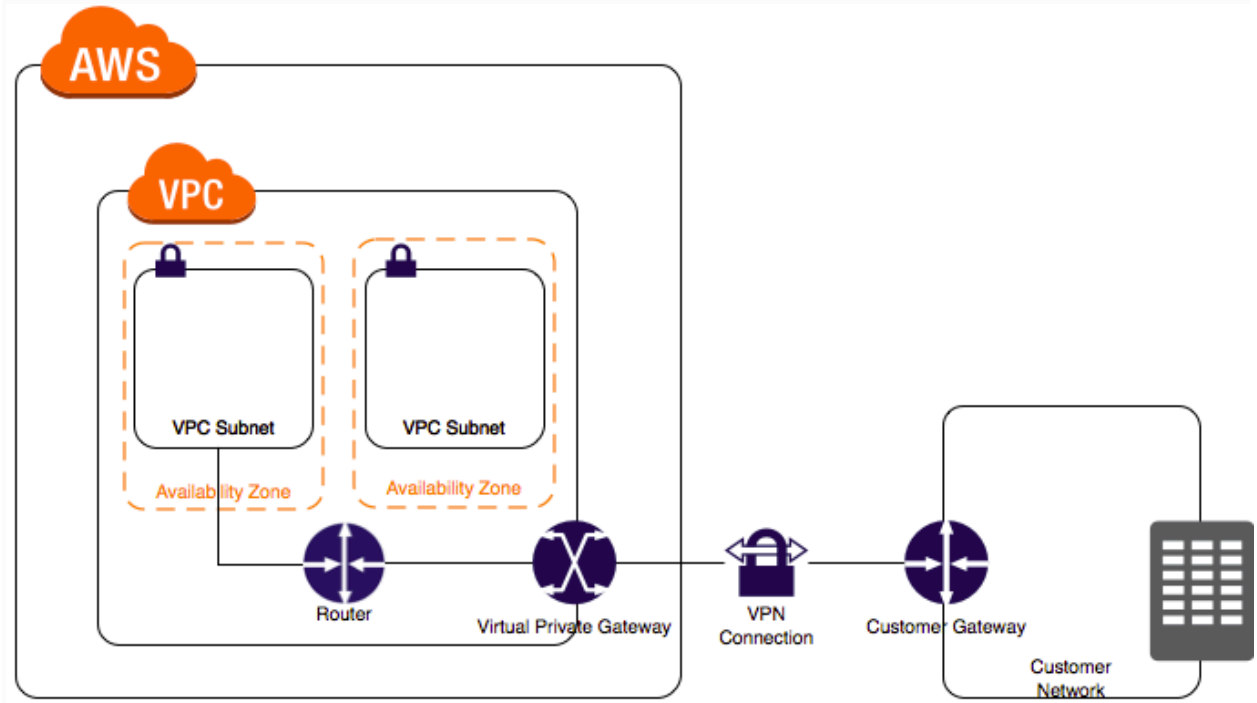
By default, instances that you launch into a virtual private cloud (VPC) can't communicate with your own network. You can enable access to your network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS managed VPN connection.

Although the term VPN connection is a general term, in the Amazon VPC documentation, a VPN connection refers to the connection between your VPC and your own network. AWS supports Internet Protocol security (IPsec) VPN connections.

A customer gateway is a physical device or software application on your side of the VPN connection.

To create a VPN connection, you must create a customer gateway resource in AWS, which provides information to AWS about your customer gateway device. Next, you have to set up an Internet-routable IP address (static) of the customer gateway's external interface.

The following diagram illustrates single VPN connections. The VPC has an attached virtual private gateway, and your remote network includes a customer gateway, which you must configure to enable the VPN connection. You set up the routing so that any traffic from the VPC bound for your network is routed to the virtual private gateway.



The options that say: **A dedicated NAT instance in a public subnet** and **the main route table in your VPC to route traffic through a NAT instance** are incorrect since you don't need a NAT instance for you to be able to create a VPN connection.

An EIP to the Virtual Private Gateway is incorrect since you do not attach an EIP to a VPG.

References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html

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

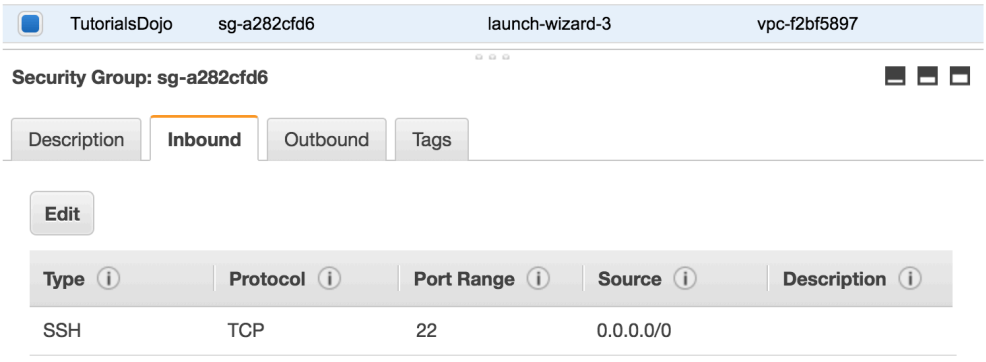
Check out this Amazon VPC Cheat Sheet:

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

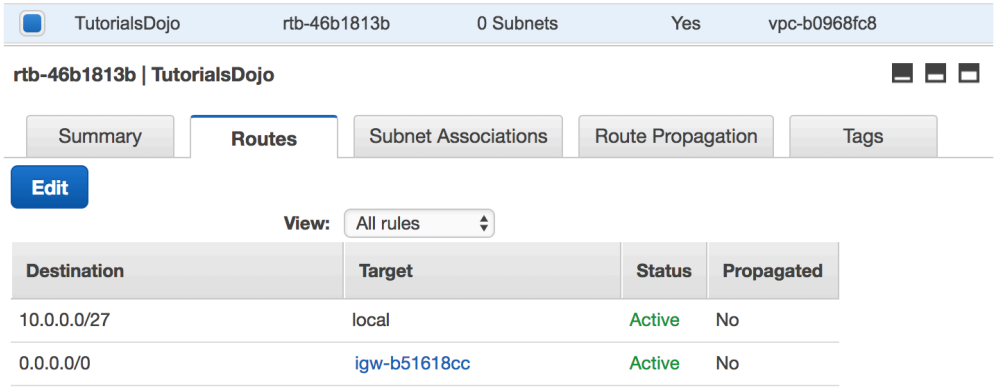
3. QUESTION

Category: CSAA – Design Secure Architectures

A company has an On-Demand EC2 instance located in a subnet in AWS that hosts a web application. The security group attached to this EC2 instance has the following Inbound Rules:



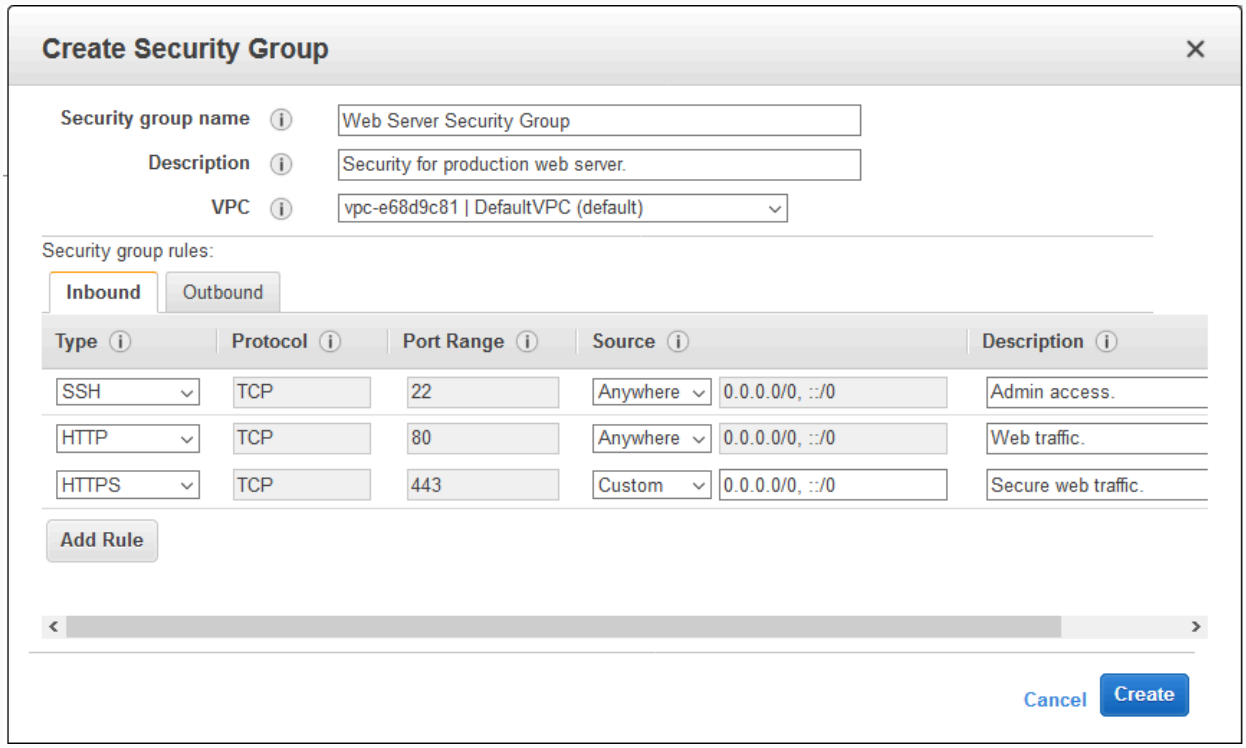
The Route table attached to the VPC is shown below. You can establish an SSH connection into the EC2 instance from the Internet. However, you are not able to connect to the web server using your Chrome browser.



Which of the below steps would resolve the issue?

- In the Security Group, remove the SSH rule.
- In the Security Group, add an Inbound HTTP rule. (Correct)
- In the Route table, add this new route entry: 10.0.0.0/27 -> local
- In the Route table, add this new route entry: 0.0.0.0 -> igw-b51618cc

In this particular scenario, you can already connect to the EC2 instance via SSH. This means that there is no problem in the Route Table of your VPC. To fix this issue, you simply need to update your Security Group and add an Inbound rule to allow HTTP traffic.



The option that says: **In the Security Group, remove the SSH rule** is incorrect as doing so will not solve the issue. It will just disable SSH traffic that is already available.

The options that say: **In the Route table, add this new route entry: 0.0.0.0 -> igw-b51618cc** and **In the Route table, add this new route entry: 10.0.0.0/27 -> local** are incorrect as there is no need to change the Route Tables.

Reference:

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

Check out this Amazon VPC Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Secure Architectures

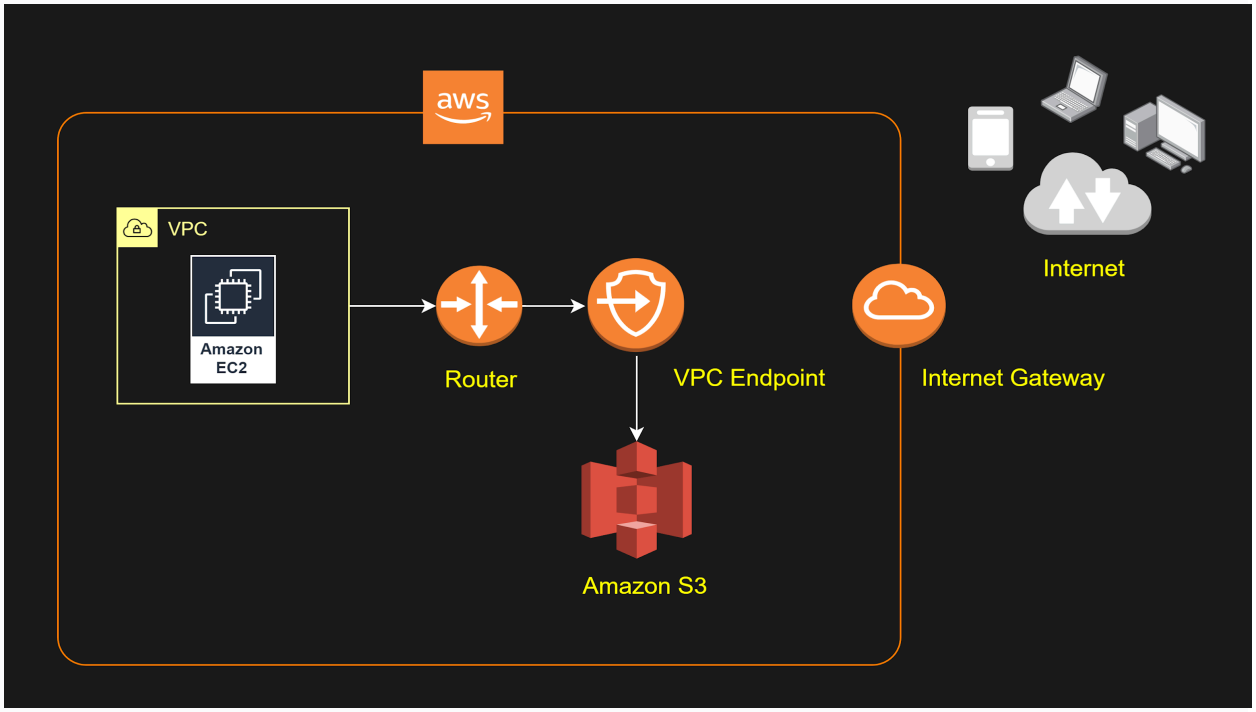
A web application is hosted on an EC2 instance that processes sensitive financial information which is launched in a private subnet. All of the data are stored in an Amazon S3 bucket. Financial information is accessed by users over the Internet. The security team of the company is concerned that the Internet connectivity to Amazon S3 is a security risk.

In this scenario, what will you do to resolve this security vulnerability in the most cost-effective manner?

- Change the web architecture to access the financial data through a Gateway VPC Endpoint. (Correct)
- Change the web architecture to access the financial data in your S3 bucket through a VPN connection.
- Change the web architecture to access the financial data hosted in your S3 bucket by creating a custom VPC endpoint service.
- Change the web architecture to access the financial data in S3 through an interface VPC endpoint, which is powered by AWS PrivateLink.

Take note that your VPC lives within a larger AWS network and the services, such as S3, DynamoDB, RDS, and many others, are located outside of your VPC, but still within the AWS network. By default, the connection that your VPC uses to connect to your S3 bucket or any other service traverses the public Internet via your Internet Gateway.

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other service does not leave the Amazon network.



There are two types of VPC endpoints: *interface endpoints* and *gateway endpoints*. You have to create the type of VPC endpoint required by the supported service.

An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service. A gateway endpoint is a gateway that is a target for a specified route in your route table, used for traffic destined to a supported AWS service.

Gateway endpoints for Amazon S3	Interface endpoints for Amazon S3
In both cases, your network traffic remains on the AWS network.	
Use Amazon S3 public IP addresses	Use private IP addresses from your VPC to access Amazon S3
Does not allow access from on premises	Allow access from on premises
Does not allow access from another AWS Region	Allow access from a VPC in another AWS Region using VPC peering or AWS Transit Gateway
Not billed	Billed

Tutorials Dojo

Hence, the correct answer is: **Change the web architecture to access the financial data through a Gateway VPC Endpoint.**

The option that says: **Changing the web architecture to access the financial data in your S3 bucket through a VPN connection** is incorrect because a VPN connection still goes through the public Internet. You have to use a VPC Endpoint in this scenario and not VPN, to privately connect your VPC to supported AWS services such as S3.

The option that says: **Changing the web architecture to access the financial data hosted in your S3 bucket by creating a custom VPC endpoint service** is incorrect

because a “VPC endpoint service” is quite different from a “VPC endpoint”. With the VPC endpoint service, you are the service provider where you can create your own application in your VPC and configure it as an AWS PrivateLink-powered service (referred to as an endpoint service). Other AWS principals can create a connection from their VPC to your endpoint service using an interface VPC endpoint.

The option that says: **Changing the web architecture to access the financial data in S3 through an interface VPC endpoint, which is powered by AWS PrivateLink** is incorrect. Although you can use an Interface VPC Endpoint to satisfy the requirement, this type entails an associated cost, unlike a Gateway VPC Endpoint. Remember that you won’t get billed if you use a Gateway VPC endpoint for your Amazon S3 bucket, unlike an Interface VPC endpoint that is billed for hourly usage and data processing charges. Take note that the scenario explicitly asks for the most cost-effective solution.

References:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html

https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html

Check out this Amazon VPC Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Resilient Architectures

A company has multiple VPCs with IPv6 enabled for its suite of web applications. The Solutions Architect tried to deploy a new Amazon EC2 instance but she received an error saying that there is no IP address available on the subnet.

How should the Solutions Architect resolve this problem?

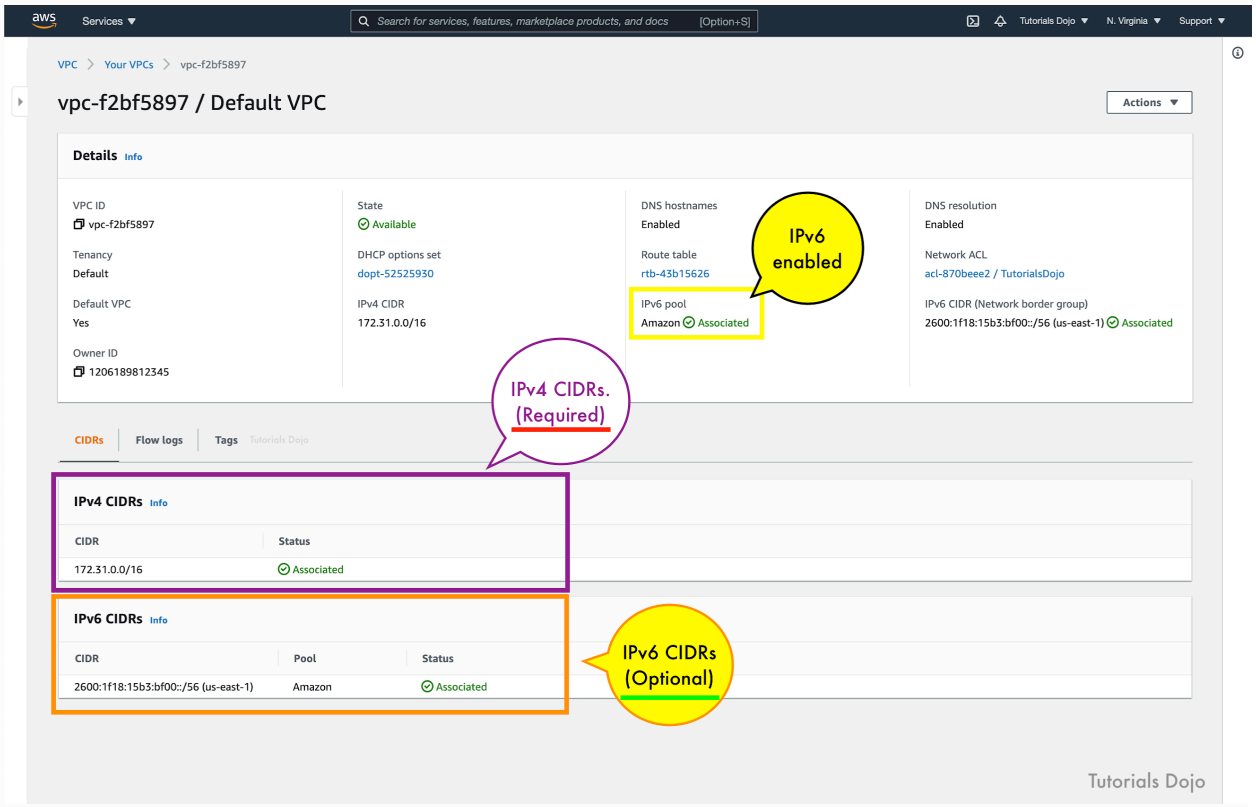
Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance.
Disable the IPv4 support in the VPC and use the available IPv6 addresses.

Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC.

Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance. (Correct)

Amazon Virtual Private Cloud (VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 for most resources in your virtual private cloud, helping to ensure secure and easy access to resources and applications.

A subnet is a range of IP addresses in your VPC. You can launch AWS resources into a specified subnet. When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. You can also optionally assign an IPv6 CIDR block to your VPC, and assign IPv6 CIDR blocks to your subnets.



If you have an existing VPC that supports IPv4 only and resources in your subnet that are configured to use IPv4 only, you can enable IPv6 support for your VPC and resources. Your VPC can operate in dual-stack mode — your resources can communicate over IPv4, or

IPv6, or both. IPv4 and IPv6 communication are independent of each other. You cannot disable IPv4 support for your VPC and subnets since this is the default IP addressing system for Amazon VPC and Amazon EC2.

By default, a new EC2 instance uses an IPv4 addressing protocol. To fix the problem in the scenario, you need to create a new IPv4 subnet and deploy the EC2 instance in the new subnet.

Hence, the correct answer is: **Set up a new IPv4 subnet with a larger CIDR range. Associate the new subnet with the VPC and then launch the instance.**

The option that says: **Set up a new IPv6-only subnet with a large CIDR range. Associate the new subnet with the VPC then launch the instance** is incorrect because you need to add IPv4 subnet first before you can create an IPv6 subnet.

The option that says: **Ensure that the VPC has IPv6 CIDRs only. Remove any IPv4 CIDRs associated with the VPC** is incorrect because you can't have a VPC with IPv6 CIDRs only. The default IP addressing system in VPC is IPv4. You can only change your VPC to dual-stack mode where your resources can communicate over IPv4, or IPv6, or both, but not exclusively with IPv6 only.

The option that says: **Disable the IPv4 support in the VPC and use the available IPv6 addresses** is incorrect because you cannot disable the IPv4 support for your VPC and subnets since this is the default IP addressing system.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-migrate-ipv6.html

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html

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

Check out this Amazon VPC Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Secure Architectures

A company hosted a web application on a Linux Amazon EC2 instance in the public subnet that uses a non-default network ACL. The instance uses a default security group and has an attached Elastic IP address. The network ACL is configured to block all

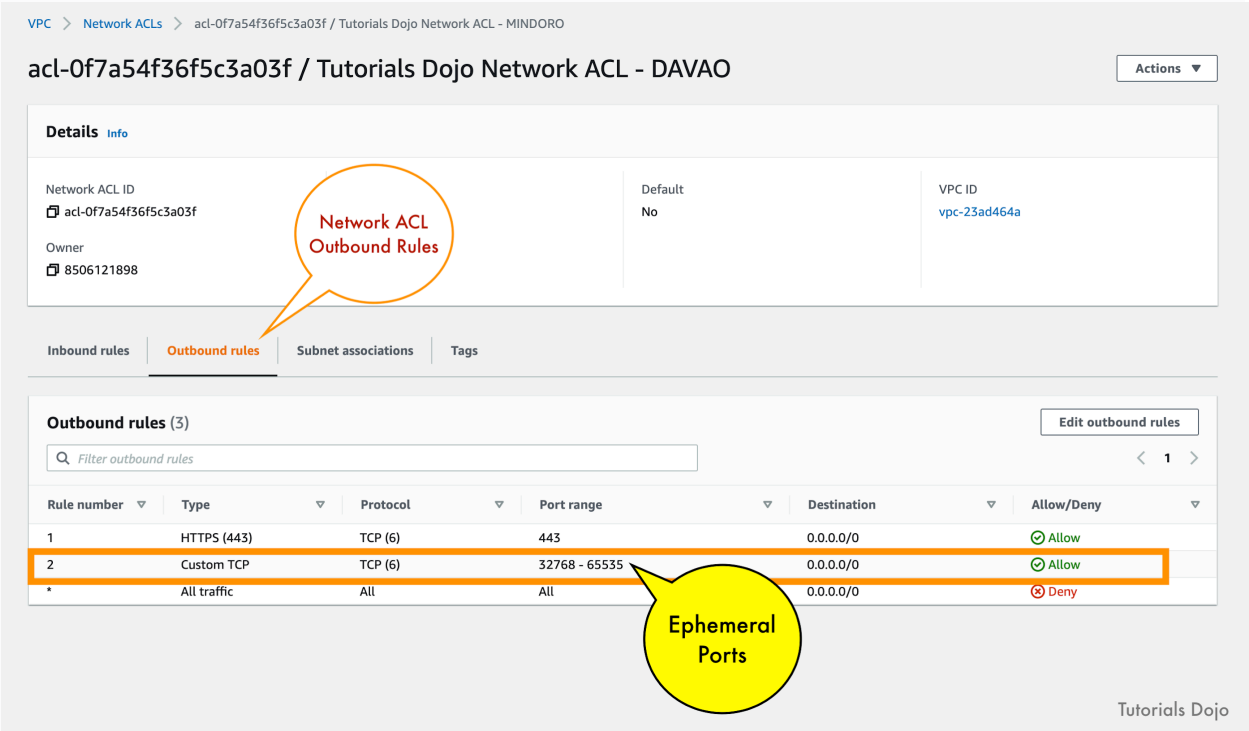
inbound and outbound traffic. The Solutions Architect must allow incoming traffic on port 443 to access the application from any source.

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

- In the Security Group, add a new rule to allow TCP connection on port 443 from source `0.0.0.0/0` (Correct)
- In the Security Group, create a new rule to allow TCP connection on port 443 to destination `0.0.0.0/0`
- In the Network ACL, update the rule to allow outbound TCP connection on port `32768 - 65535` to destination `0.0.0.0/0`
- In the Network ACL, update the rule to allow both inbound and outbound TCP connection on port 443 from source `0.0.0.0/0` and to destination `0.0.0.0/0`
- In the Network ACL, update the rule to allow inbound TCP connection on port 443 from source `0.0.0.0/0` and outbound TCP connection on port `32768 - 65535` to destination `0.0.0.0/0` (Correct)

In order to connect to a service running on an instance, you need to make sure that both inbound traffic on the port that the service is listening on and outbound traffic from ephemeral ports are allowed in the associated network ACL. When a client connects to a server, a random port is generated (like 1024-65535) from the ephemeral port range with this becoming the client's source port.

The designated ephemeral port then becomes the destination port for return traffic from the service, so outbound traffic from the ephemeral port must be allowed in the network ACL. By default, network ACLs allow all inbound and outbound traffic. If your network ACL is more restrictive, then you need to explicitly allow traffic from the ephemeral port range.



The client that initiates the request chooses the ephemeral port range. The range varies depending on the client's operating system.

– Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000.

– Requests originating from Elastic Load Balancing use ports 1024-65535.

– Windows operating systems through Windows Server 2003 use ports 1025-5000.

– Windows Server 2008 and later versions use ports 49152-65535.

– A NAT gateway uses ports 1024-65535.

– AWS Lambda functions use ports 1024-65535.

For example, if a request comes into a web server in your VPC from a Windows 10 client on the Internet, your network ACL must have an outbound rule to enable traffic destined for ports 49152 – 65535. If an instance in your VPC is the client initiating a request, your network ACL must have an inbound rule to enable traffic destined for the ephemeral ports specific to the type of instance (Amazon Linux, Windows Server 2008, and so on).

In this scenario, you only need to allow the incoming traffic on port 443. Since security groups are stateful, you can apply any changes to an incoming rule and it will be automatically applied to the outgoing rule.

To enable the connection to a service running on an instance, the associated network ACL must allow both inbound traffic on the port that the service is listening on as well as outbound traffic from ephemeral ports. When a client connects to a server, a random port

from the ephemeral port range (32768 – 65535) becomes the client's source port. Since the return traffic will use an ephemeral port, outbound traffic must be allowed on these ports to destination 0.0.0.0/0.

Hence, the correct answers are:

– In the Security Group, add a new rule to allow TCP connection on port 443 from source `0.0.0.0/0`.

– In the Network ACL, update the rule to allow inbound TCP connection on port 443 from source `0.0.0.0/0` and outbound TCP connection on port `32768 - 65535` to destination `0.0.0.0/0`.

The option that says: **In the Security Group, create a new rule to allow TCP connection on port 443 to destination `0.0.0.0/0`** is incorrect because this step just allows outbound connections from the EC2 instance out to the public Internet, which is unnecessary. Remember that a default security group already includes an outbound rule that allows all outbound traffic.

The option that says: **In the Network ACL, update the rule to allow both inbound and outbound TCP connection on port 443 from source 0.0.0.0/0 and to destination `0.0.0.0/0`** is incorrect because your network ACL must have an outbound rule to allow ephemeral ports (`32768 - 65535`). These are the specific ports that will be used as the client's source port for the traffic response.

The option that says: **In the Network ACL, update the rule to allow outbound TCP connection on port `32768 - 65535` to destination `0.0.0.0/0`** is incorrect because this step is just partially right. You still need to add an inbound rule from port 443 and not just the outbound rule for the ephemeral ports (`32768 - 65535`).

References:

https://aws.amazon.com/premiumsupport/knowledge-center/connect-http-https-ec2/

https://docs.amazonaws.cn/en_us/vpc/latest/userguide/vpc-network-acls.html#nacl-ephemeral-ports

Check out this Amazon VPC Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Secure Architectures

A local bank has an in-house application that handles sensitive financial data in a private subnet. After the data is processed by the EC2 worker instances, they will be delivered to S3 for ingestion by other services.

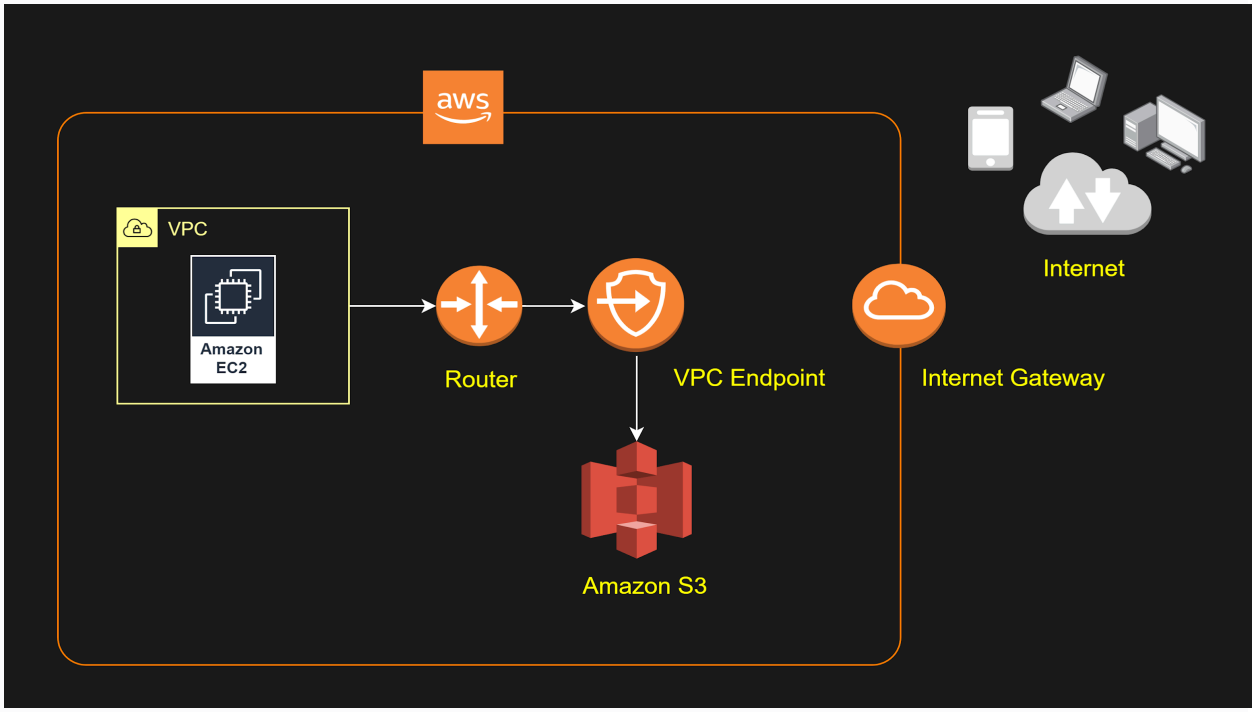
How should you design this solution so that the data does not pass through the public Internet?

- Provision a NAT gateway in the private subnet with a corresponding route entry that directs the data to S3.
- Create an Internet gateway in the public subnet with a corresponding route entry that directs the data to S3.
- Configure a VPC Endpoint along with a corresponding route entry that directs the data to S3. (Correct)
- Configure a Transit gateway along with a corresponding route entry that directs the data to S3.

The important concept that you have to understand in this scenario is that your VPC and your S3 bucket are located within the larger AWS network. However, the traffic coming from your VPC to your S3 bucket is traversing the public Internet by default. To better protect your data in transit, you can set up a VPC endpoint so the incoming traffic from your VPC will not pass through the public Internet, but instead through the private AWS network.

A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an Internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC do not require public IP addresses to communicate with resources in the service. Traffic between your VPC and the other services does not leave the Amazon network.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.



Hence, the correct answer is: **Configure a VPC Endpoint along with a corresponding route entry that directs the data to S3.**

The option that says: **Create an Internet gateway in the public subnet with a corresponding route entry that directs the data to S3** is incorrect because the Internet gateway is used for instances in the public subnet to have accessibility to the Internet.

The option that says: **Configure a Transit gateway along with a corresponding route entry that directs the data to S3** is incorrect because the Transit Gateway is used for interconnecting VPCs and on-premises networks through a central hub. Since Amazon S3 is outside of VPC, you still won't be able to connect to it privately.

The option that says: **Provision a NAT gateway in the private subnet with a corresponding route entry that directs the data to S3** is incorrect because NAT Gateway allows instances in the private subnet to gain access to the Internet, but not vice versa.

References:

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

https://docs.aws.amazon.com/vpc/latest/userguide/vpce-gateway.html

Check out this Amazon VPC Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design Secure Architectures

A media company has two VPCs: VPC-1 and VPC-2 with peering connection between each other. VPC-1 only contains private subnets while VPC-2 only contains public subnets. The company uses a single AWS Direct Connect connection and a virtual interface to connect their on-premises network with VPC-1.

Which of the following options increase the fault tolerance of the connection to VPC-1? (Select TWO.)

- Establish a hardware VPN over the Internet between VPC-1 and the on-premises network. (Correct)
- Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.
- Establish another AWS Direct Connect connection and private virtual interface in the same AWS region as VPC-1. (Correct)
- Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.
- Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

Incorrect

In this scenario, you have two VPCs that have peering connections with each other. Note that a VPC peering connection does not support edge-to-edge routing. This means that if either VPC in a peering relationship has one of the following connections, you cannot extend the peering relationship to that connection:

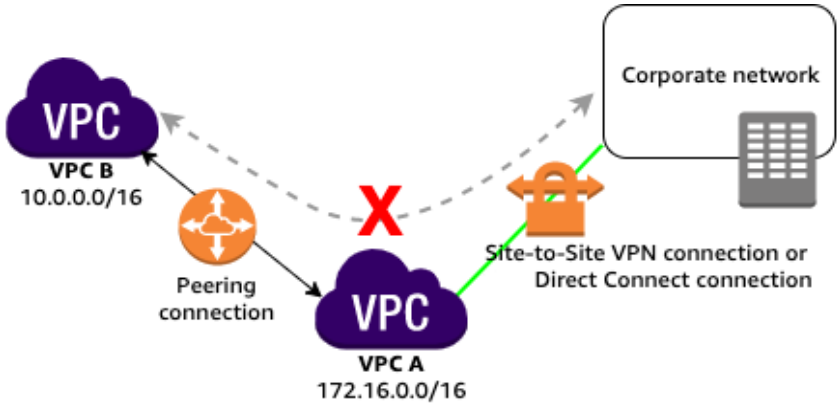
– A VPN connection or an AWS Direct Connect connection to a corporate network

– An Internet connection through an Internet gateway

– An Internet connection in a private subnet through a NAT device

– A gateway VPC endpoint to an AWS service; for example, an endpoint to Amazon S3.

– (IPv6) A ClassicLink connection. You can enable IPv4 communication between a linked EC2-Classic instance and instances in a VPC on the other side of a VPC peering connection. However, IPv6 is not supported in EC2-Classic, so you cannot extend this connection for IPv6 communication.



For example, if VPC A and VPC B are peered, and VPC A has any of these connections, then instances in VPC B cannot use the connection to access resources on the other side of the connection. Similarly, resources on the other side of a connection cannot use the connection to access VPC B.

Hence, this implies that VPC-2 cannot extend the peering relationship between VPC-1 and the on-premises network. In other words, traffic originating from the corporate network cannot establish a direct connection to VPC-1 by routing it through VPC-2, whether using a VPN connection or an AWS Direct Connect connection. The VPC peering connection is a one-to-one relationship with directly peered entities (VPC-1 and the on-premises network), and it does not support transitive communication through intermediate VPCs like VPC-2, which is why the following options are incorrect:

– Use the AWS VPN CloudHub to create a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

– Establish a hardware VPN over the Internet between VPC-2 and the on-premises network.

– Establish a new AWS Direct Connect connection and private virtual interface in the same region as VPC-2.

You can do the following to provide a highly available, fault-tolerant network connection:

– Establish a hardware VPN over the Internet between the VPC and the on-premises network.

– Establish another AWS Direct Connect connection and private virtual interface in the same AWS region as VPC-1.

References:

https://docs.aws.amazon.com/vpc/latest/peering/invalid-peering-configurations.html#edge-to-edge-vgw

https://docs.aws.amazon.com/whitepapers/latest/hybrid-connectivity/vpn-connection-as-a-backup-to-aws-dx-connection-example.html

https://aws.amazon.com/answers/networking/aws-multiple-data-center-ha-network-connectivity/

Check out these Amazon VPC and AWS Direct Connect Cheat Sheets:

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

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

Topic-Based – CloudFront (SA-Associate)

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A global news network created a CloudFront distribution for their web application. However, you noticed that the application's origin server is being hit for each request instead of the AWS Edge locations, which serve the cached objects. The issue occurs even for the commonly requested objects.

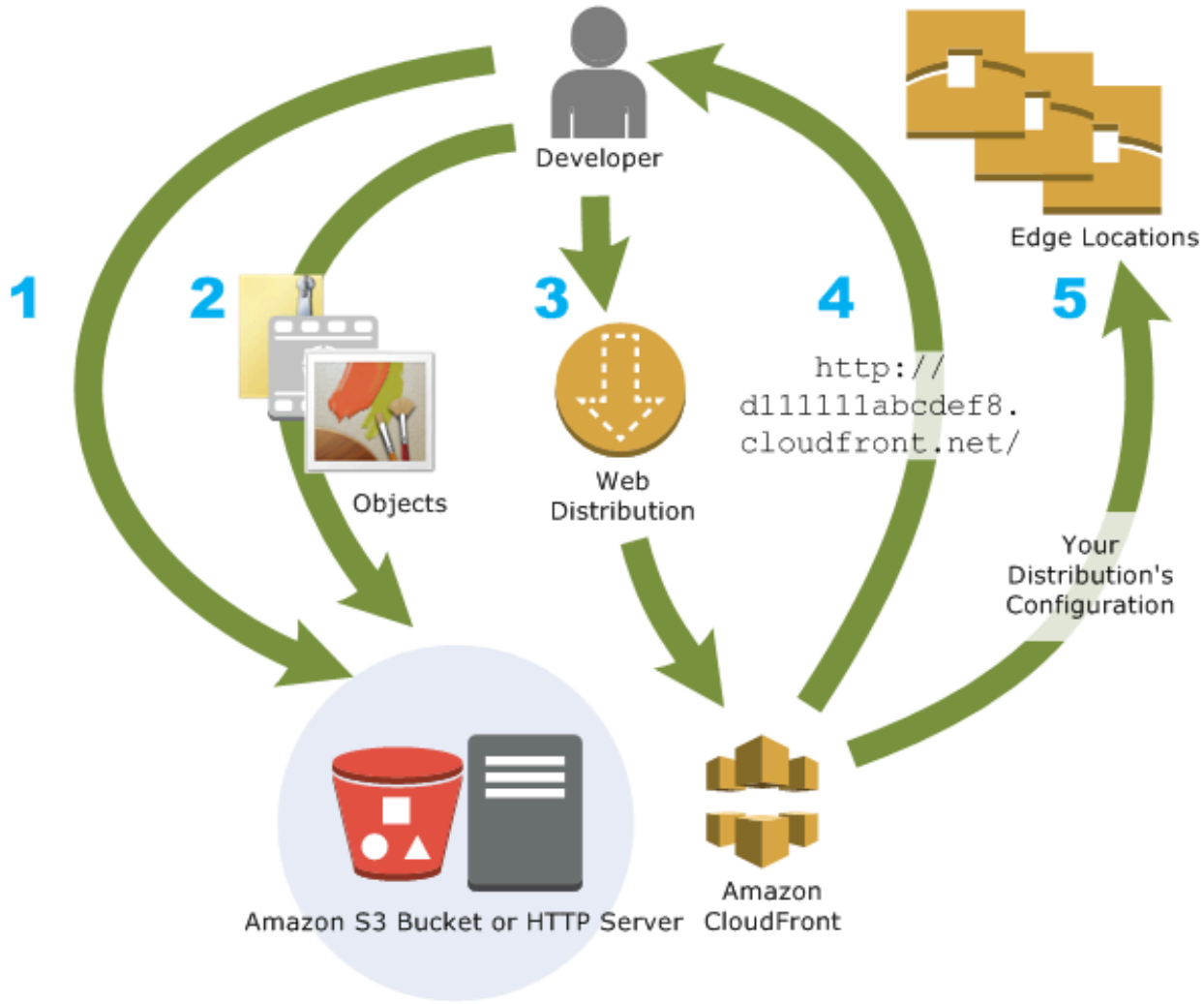
What could be a possible cause of this issue?

An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server.
The file sizes of the cached objects are too large for CloudFront to handle.
The Cache-Control max-age directive is set to zero. (Correct)
There are two primary origins configured in your Amazon CloudFront Origin Group.

Incorrect

You can control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin. Reducing the duration allows you to serve dynamic content. Increasing the duration means your users get better performance because your objects are more likely to be served directly from the edge cache. A longer duration also reduces the load on your origin.

Typically, CloudFront serves an object from an edge location until the cache duration that you specified passes — that is, until the object expires. After it expires, the next time the edge location gets a user request for the object, CloudFront forwards the request to the origin server to verify that the cache contains the latest version of the object.



The `Cache-Control` and `Expires` headers control how long objects stay in the cache. The `Cache-Control max-age` directive lets you specify how long (in seconds) you want an object to remain in the cache before CloudFront gets the object again from the origin server. The minimum expiration time CloudFront supports is 0 seconds for web distributions and 3600 seconds for RTMP distributions.

In this scenario, the main culprit is that the Cache-Control max-age directive is set to a low value, which is why the request is always directed to your origin server.

Hence, the correct answer is: **The Cache-Control max-age directive is set to zero.**

The option that says: An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server is incorrect because the issue also occurs even for the commonly requested objects. This means that these objects were successfully requested before but due to a zero Cache-Control max-age directive value, it causes this issue in CloudFront.

The option that says: **The file sizes of the cached objects are too large for CloudFront to handle** is incorrect because this is not related to the issue in caching.

The option that says: There are two primary origins configured in your Amazon CloudFront Origin Group is incorrect because you cannot set two origins in CloudFront in the first place. An origin group includes two origins which are the primary origin and the second origin that will be used for the actual failover. It also includes the failover criteria that you need to specify. In this scenario, the issue is more on the cache hit ratio and not on origin failovers.

Reference:

http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Expiration.html

Check out this Amazon CloudFront Cheat Sheet:

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

2. QUESTION

Category: CSAA – Design Secure Architectures

A company plans to conduct a network security audit. The web application is hosted on an Auto Scaling group of EC2 Instances with an Application Load Balancer in front to evenly distribute the incoming traffic. A Solutions Architect has been tasked to enhance the security posture of the company's cloud infrastructure and minimize the impact of DDoS attacks on its resources.

Which of the following is the most effective solution that should be implemented?

Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification.

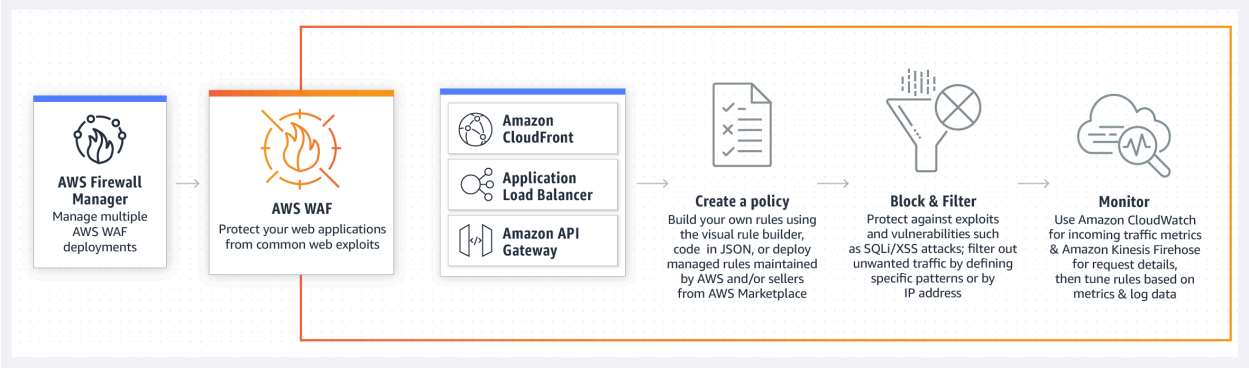
Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that processes the flow logs and invokes Amazon SNS for notification.

Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront. (Correct)

Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification.

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define. You can deploy AWS WAF on Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on EC2, or Amazon API Gateway for your APIs.

To detect and mitigate DDoS attacks, you can use AWS WAF in addition to AWS Shield. AWS WAF is a web application firewall that helps detect and mitigate web application layer DDoS attacks by inspecting traffic inline. Application layer DDoS attacks use well-formed but malicious requests to evade mitigation and consume application resources. You can define custom security rules that contain a set of conditions, rules, and actions to block attacking traffic. After you define web ACLs, you can apply them to CloudFront distributions, and web ACLs are evaluated in the priority order you specified when you configured them.



By using AWS WAF, you can configure web access control lists (Web ACLs) on your CloudFront distributions or Application Load Balancers to filter and block requests based on request signatures. Each Web ACL consists of rules that you can configure to string match or regex match one or more request attributes, such as the URI, query-string, HTTP method, or header key. In addition, by using AWS WAF's rate-based rules, you can automatically block the IP addresses of bad actors when requests matching a rule exceed a threshold that you define. Requests from offending client IP addresses will receive 403 Forbidden error responses and will remain blocked until request rates drop below the

threshold. This is useful for mitigating HTTP flood attacks that are disguised as regular web traffic.

It is recommended that you add web ACLs with rate-based rules as part of your AWS Shield Advanced protection. These rules can alert you to sudden spikes in traffic that might indicate a potential DDoS event. A rate-based rule counts the requests that arrive from any individual address in any five-minute period. If the number of requests exceeds the limit that you define, the rule can trigger an action such as sending you a notification.

Hence, the correct answer is: **Configure Amazon CloudFront distribution and set Application Load Balancer as the origin. Create a rate-based web ACL rule using AWS WAF and associate it with Amazon CloudFront.**

The option that says: **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use VPC Flow Logs to monitor abnormal traffic patterns. Set up a custom AWS Lambda function that processes the flow logs and invokes Amazon SNS for notification** is incorrect because this option only allows you to monitor the traffic that is reaching your instance. You can't use VPC Flow Logs to mitigate DDoS attacks.

The option that says: **Configure Amazon CloudFront distribution and set an Application Load Balancer as the origin. Create a security group rule and deny all the suspicious addresses. Use Amazon SNS for notification** is incorrect. To deny suspicious addresses, you must manually insert the IP addresses of these hosts. This is a manual task which is not a sustainable solution. Take note that attackers generate large volumes of packets or requests to overwhelm the target system. Using a security group in this scenario won't help you mitigate DDoS attacks.

The option that says: **Configure Amazon CloudFront distribution and set a Network Load Balancer as the origin. Use Amazon GuardDuty to block suspicious hosts based on its security findings. Set up a custom AWS Lambda function that processes the security logs and invokes Amazon SNS for notification** is incorrect because Amazon GuardDuty is just a threat detection service. You should use AWS WAF and create your own AWS WAF rate-based rules for mitigating HTTP flood attacks that are disguised as regular web traffic.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html

https://docs.aws.amazon.com/waf/latest/developerguide/ddos-get-started-rate-based-rules.html

https://d0.awsstatic.com/whitepapers/Security/DDoS_White_Paper.pdf

Check out this AWS WAF Cheat Sheet:

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

3. QUESTION

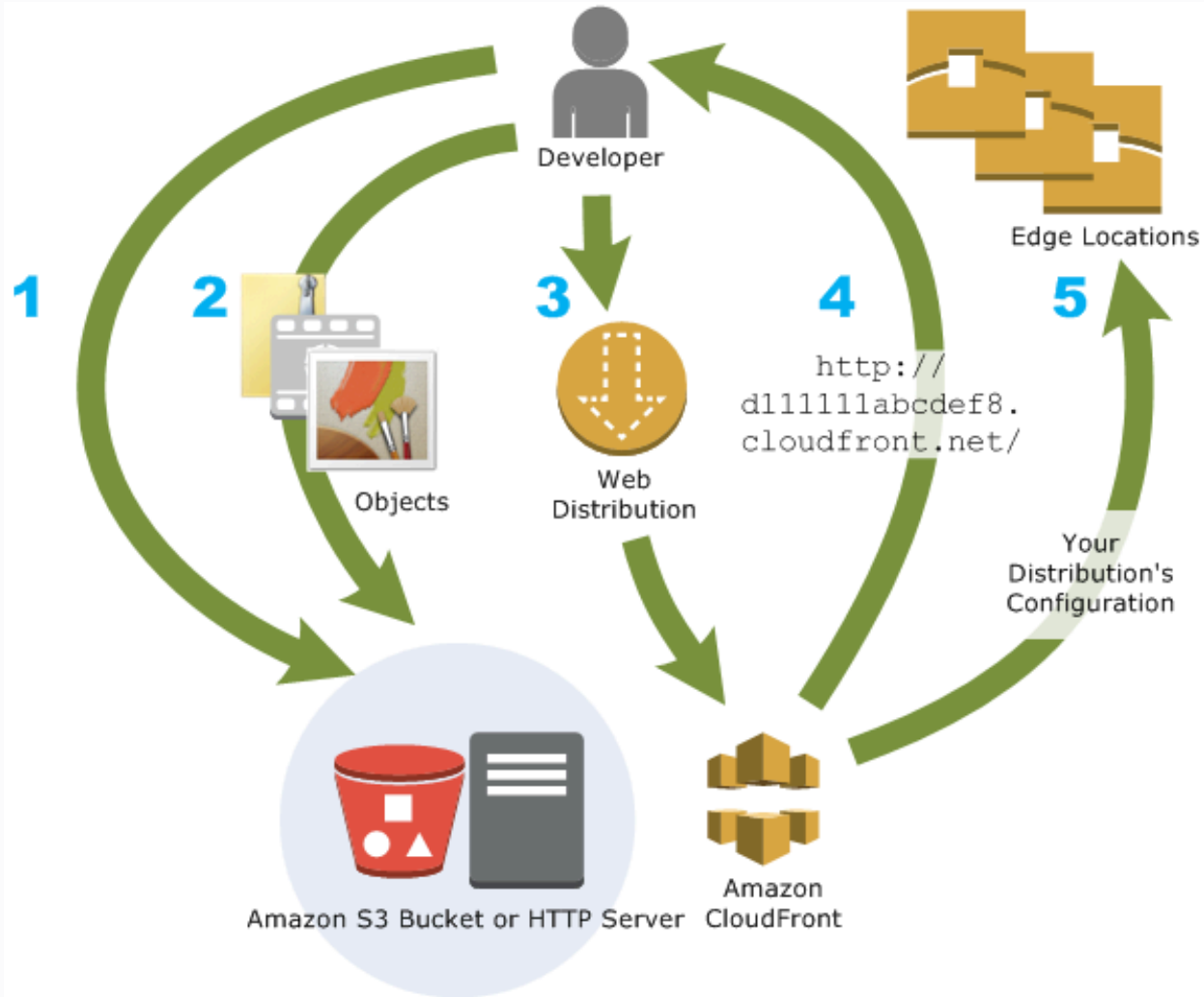
Category: CSAA – Design High-Performing Architectures

A company has a global news website hosted in a fleet of EC2 Instances. Lately, the load on the website has increased which resulted in slower response time for the site visitors. This issue impacts the revenue of the company as some readers tend to leave the site if it does not load after 10 seconds.

Which of the below services in AWS can be used to solve this problem? (Select TWO.)

- Deploy the website to all regions in different VPCs for faster processing.
- Use Amazon ElastiCache for the website's in-memory data store or cache. (Correct)
- For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions.
- Use Amazon CloudFront with website as the custom origin. (Correct)

The global news website has a problem with latency considering that there are a lot of readers of the site from all parts of the globe. In this scenario, you can use a content delivery network (CDN) which is a geographically distributed group of servers that work together to provide fast delivery of Internet content. And since this is a news website, most of its data are read-only, which can be cached to improve the read throughput and avoid repetitive requests from the server.



In AWS, Amazon CloudFront is the global content delivery network (CDN) service that you can use and for web caching, Amazon ElastiCache is the suitable service.

Hence, the correct answers are:

– Use Amazon CloudFront with website as the custom origin.

– Use Amazon ElastiCache for the website's in-memory data store or cache.

The option that says: **For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions** is incorrect as AWS Storage Gateway is used for storage.

Deploying the website to all regions in different VPCs for faster processing is incorrect as this would be costly and totally unnecessary considering that you can use Amazon CloudFront and ElastiCache to improve the performance of the website.

References:

https://aws.amazon.com/elasticache/

http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html

Check out this Amazon CloudFront Cheat Sheet:

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

4. 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?

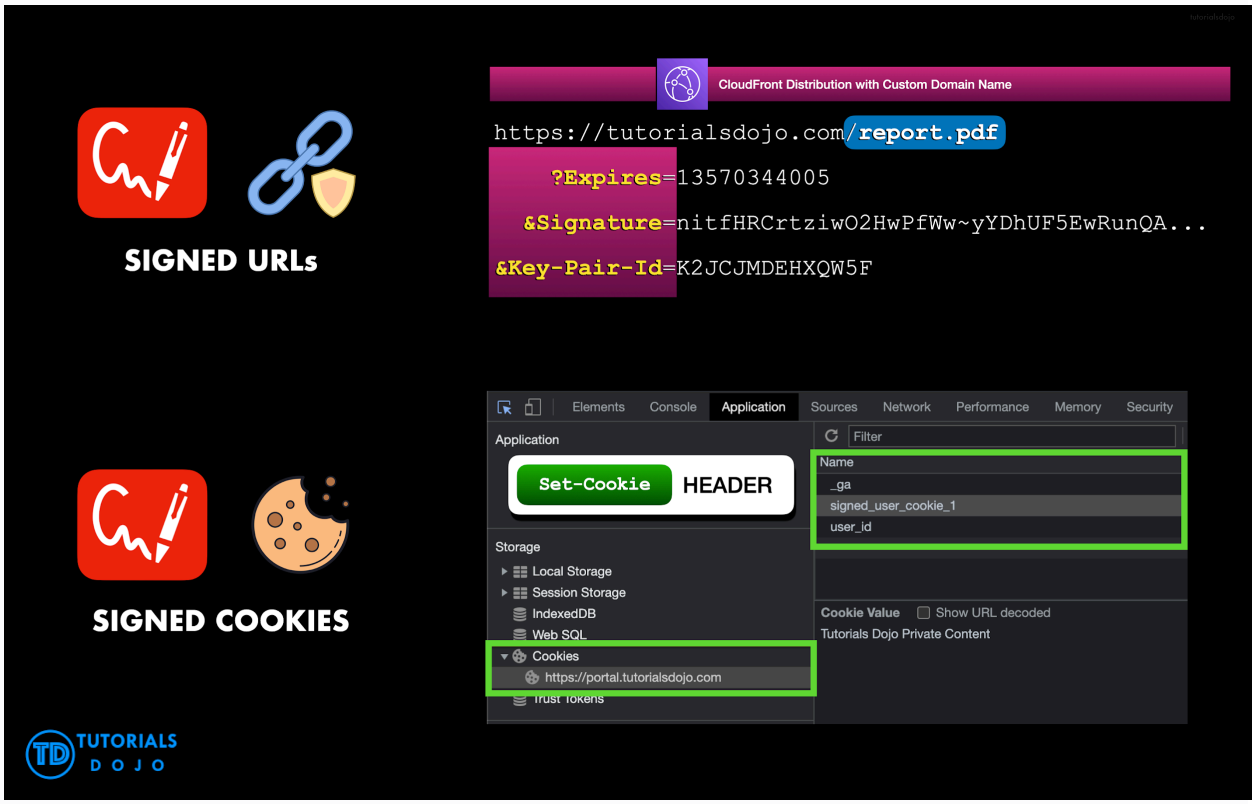
Create a Signed URL with a custom policy which only allows the members to see the private files.
Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members. Configure your CloudFront distribution to use Match Viewer as its Origin 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.
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/

5. QUESTION

Category: CSAA – Design Secure Architectures

A company has clients all across the globe that access product files stored in several S3 buckets, which are behind each of their own CloudFront web distributions. They currently want to deliver their content to a specific client, and they need to make sure that only that client can access the data. Currently, all of their clients can access their S3 buckets directly using an S3 URL or through their CloudFront distribution. The Solutions Architect must serve the private content via CloudFront only, to secure the distribution of files.

Which combination of actions should the Architect implement to meet the above requirements? (Select TWO.)

- Require the users to access the private content by using special CloudFront signed URLs or signed cookies. (Correct)
- Create a custom CloudFront function to check and ensure that only their clients can access the files.
- Enable the Origin Shield feature of the Amazon CloudFront distribution to protect the files from unauthorized access.
- Use S3 pre-signed URLs to ensure that only their client can access the files. Remove permission to use Amazon S3 URLs to read the files for anyone else.
- Restrict access to files in the origin by creating an origin access identity (OAI) and give it permission to read the files in the bucket. (Correct)

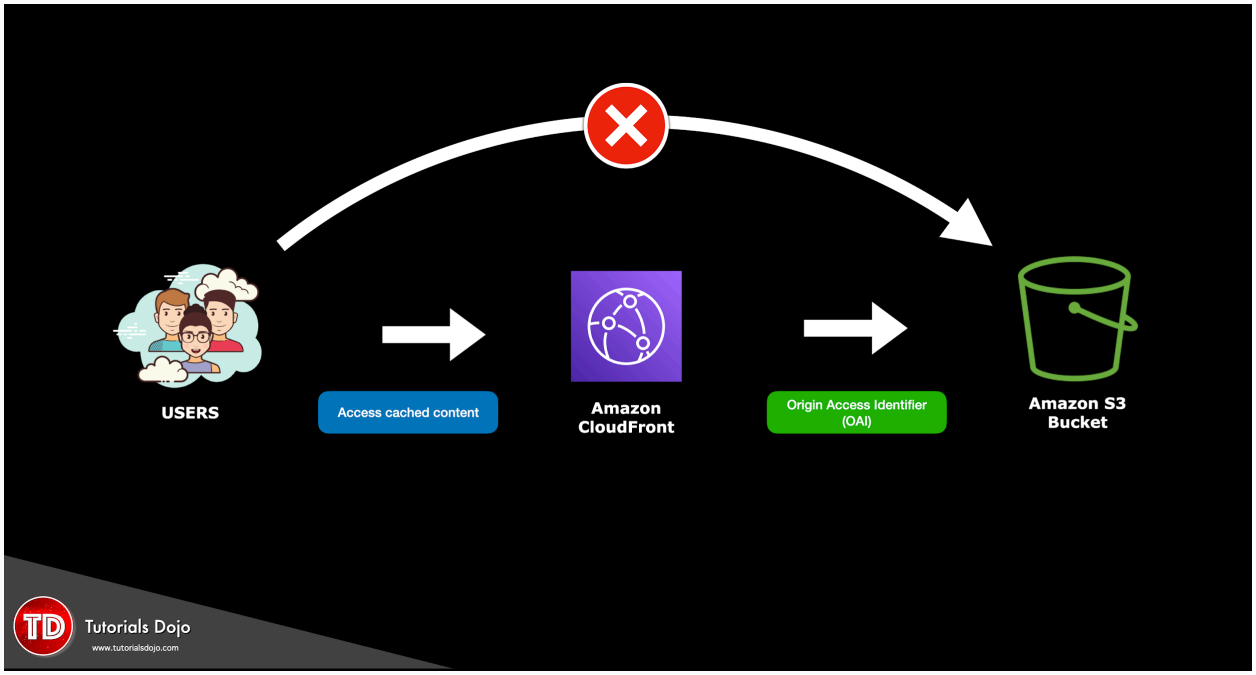
Many companies that distribute content over the Internet want to restrict access to documents, business data, media streams, or content that is intended for selected users, for

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 Amazon S3 content by using CloudFront URLs, not Amazon S3 URLs. Requiring CloudFront URLs isn't necessary, but it is recommended to prevent users from bypassing the restrictions that you specify in signed URLs or signed cookies. You can do this by setting up an origin access identity (OAI) for your Amazon S3 bucket. You can also configure the custom headers for a private HTTP server or an Amazon S3 bucket configured as a website endpoint.

All objects and buckets by default are private. The pre-signed URLs are useful if you want your user/customer to be able to upload a specific object to your bucket, but you don't require them to have AWS security credentials or permissions.



You can generate a pre-signed URL programmatically using the AWS SDK for Java or the AWS SDK for .NET. If you are using Microsoft Visual Studio, you can also use AWS Explorer to generate a pre-signed object URL without writing any code. Anyone who receives a valid pre-signed URL can then programmatically upload an object.

Hence, the correct answers are:

– Restrict access to files in the origin by creating an origin access identity (OAI) and give it permission to read the files in the bucket.

– Require the users to access the private content by using special CloudFront signed URLs or signed cookies.

The option that says: **Create a custom CloudFront function to check and ensure that only their clients can access the files** is incorrect. CloudFront Functions are just lightweight functions in JavaScript for high-scale, latency-sensitive CDN customizations and not for enforcing security. A CloudFront Function runtime environment offers submillisecond startup times which allows your application to scale immediately to handle millions of requests per second. But again, this can't be used to restrict access to your files.

The option that says: **Enable the Origin Shield feature of the Amazon CloudFront distribution to protect the files from unauthorized access** is incorrect because this feature is not primarily used for security but for improving your origin's load times, improving origin availability, and reducing your overall operating costs in CloudFront.

The option that says: **Use S3 pre-signed URLs to ensure that only their client can access the files. Remove permission to use Amazon S3 URLs to read the files for anyone else** is incorrect. Although this could be a valid solution, it doesn't satisfy the requirement to serve the private content via CloudFront only to secure the distribution of files. A better solution is to set up an origin access identity (OAI) then use Signed URL or Signed Cookies in your CloudFront web distribution.

References:

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

https://docs.aws.amazon.com/AmazonS3/latest/dev/PresignedUrlUploadObject.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/

6. 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.

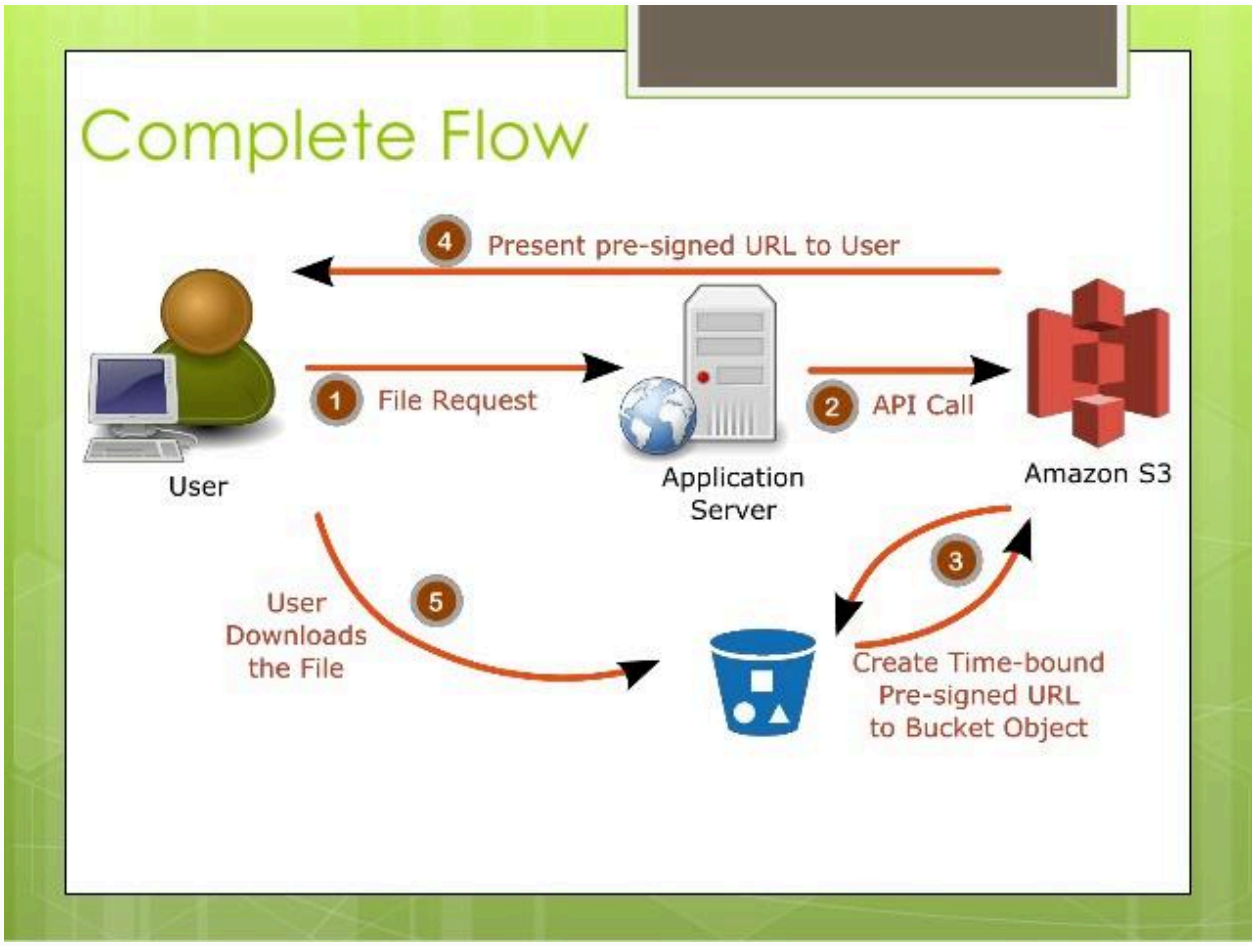
What is the MOST effective method to mitigate this issue?

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/

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

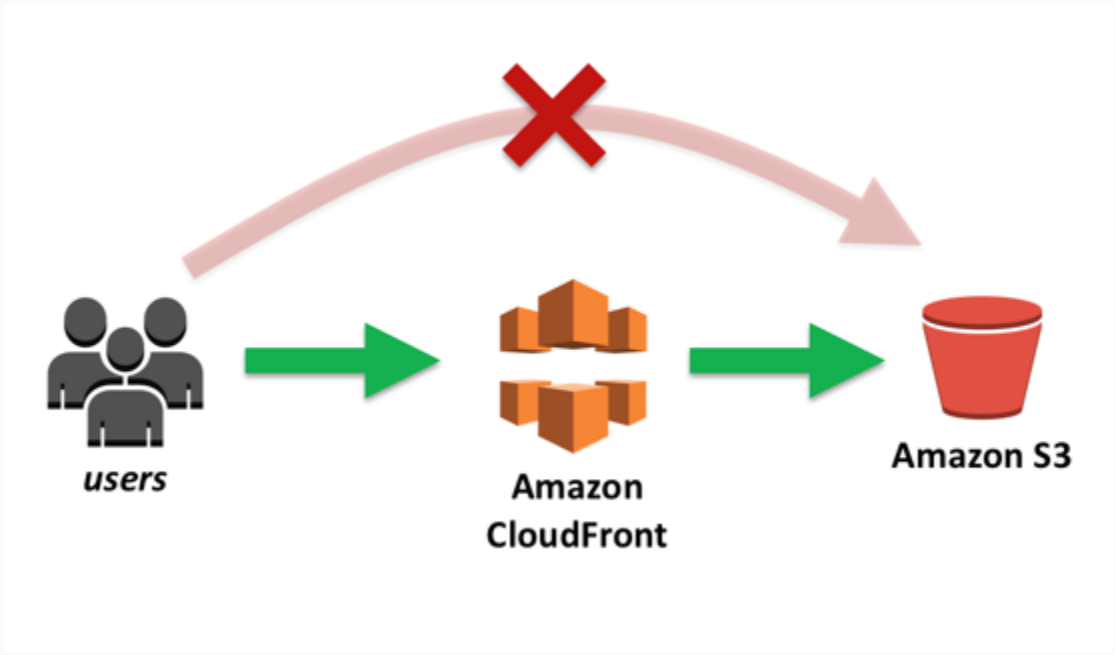
A digital media company shares static content to its premium users around the world and also to their partners who syndicate their media files. The company is looking for ways to reduce its server costs and securely deliver their data to their customers globally with low latency.

Which combination of services should be used to provide the MOST suitable and cost-effective architecture? (Select TWO.)

- Amazon CloudFront (Correct)
- Amazon S3 (Correct)
- AWS Fargate
- AWS Global Accelerator
- AWS Lambda

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services. CloudFront works seamlessly with services, including AWS Shield for DDoS mitigation, Amazon S3, Elastic Load Balancing or Amazon EC2 as origins for your applications, and Lambda@Edge to run custom code closer to customers' users and to customize the user experience. Lastly, if you use AWS origins such as Amazon S3, Amazon EC2 or Elastic Load Balancing, you don't pay for any data transferred between these services and CloudFront.



Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers an extremely durable, highly available, and infinitely scalable data storage infrastructure at very low costs.

AWS Global Accelerator and Amazon CloudFront are separate services that use the AWS global network and its edge locations around the world. CloudFront improves performance for both cacheable content (such as images and videos) and dynamic content (such as API acceleration and dynamic site delivery). Global Accelerator improves performance for a wide range of applications over TCP or UDP by proxying packets at the edge to applications running in one or more AWS Regions. Global Accelerator is a good fit for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Both services integrate with AWS Shield for DDoS protection.

Hence, the correct options are **Amazon CloudFront** and **Amazon S3.**

AWS Fargate is incorrect because this service is just a serverless compute engine for containers that work with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Although this service is more cost-effective than its

server-based counterpart, Amazon S3 still costs way less than Fargate, especially for storing static content.

AWS Lambda is incorrect because this simply lets you run your code serverless without provisioning or managing servers. Although this is also a cost-effective service since you have to pay only for the compute time you consume, you can't use this to store static content or as a Content Delivery Network (CDN). A better combination is Amazon CloudFront and Amazon S3.

AWS Global Accelerator is incorrect because this service is more suitable for non-HTTP use cases, such as gaming (UDP), IoT (MQTT), or Voice over IP, as well as for HTTP use cases that specifically require static IP addresses or deterministic, fast regional failover. Moreover, there is no direct way that you can integrate AWS Global Accelerator with Amazon S3. It's more suitable to use Amazon CloudFront instead in this scenario.

References:

https://aws.amazon.com/premiumsupport/knowledge-center/cloudfront-serve-static-website/

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

https://aws.amazon.com/global-accelerator/faqs/

8. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A solutions architect is instructed to host a website that consists of HTML, CSS, and some Javascript files. The web pages will display several high-resolution images. The website should have optimal loading times and be able to respond to high request rates.

Which of the following architectures can provide the most cost-effective and fastest loading experience?

Host the website in an AWS Elastic Beanstalk environment. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.

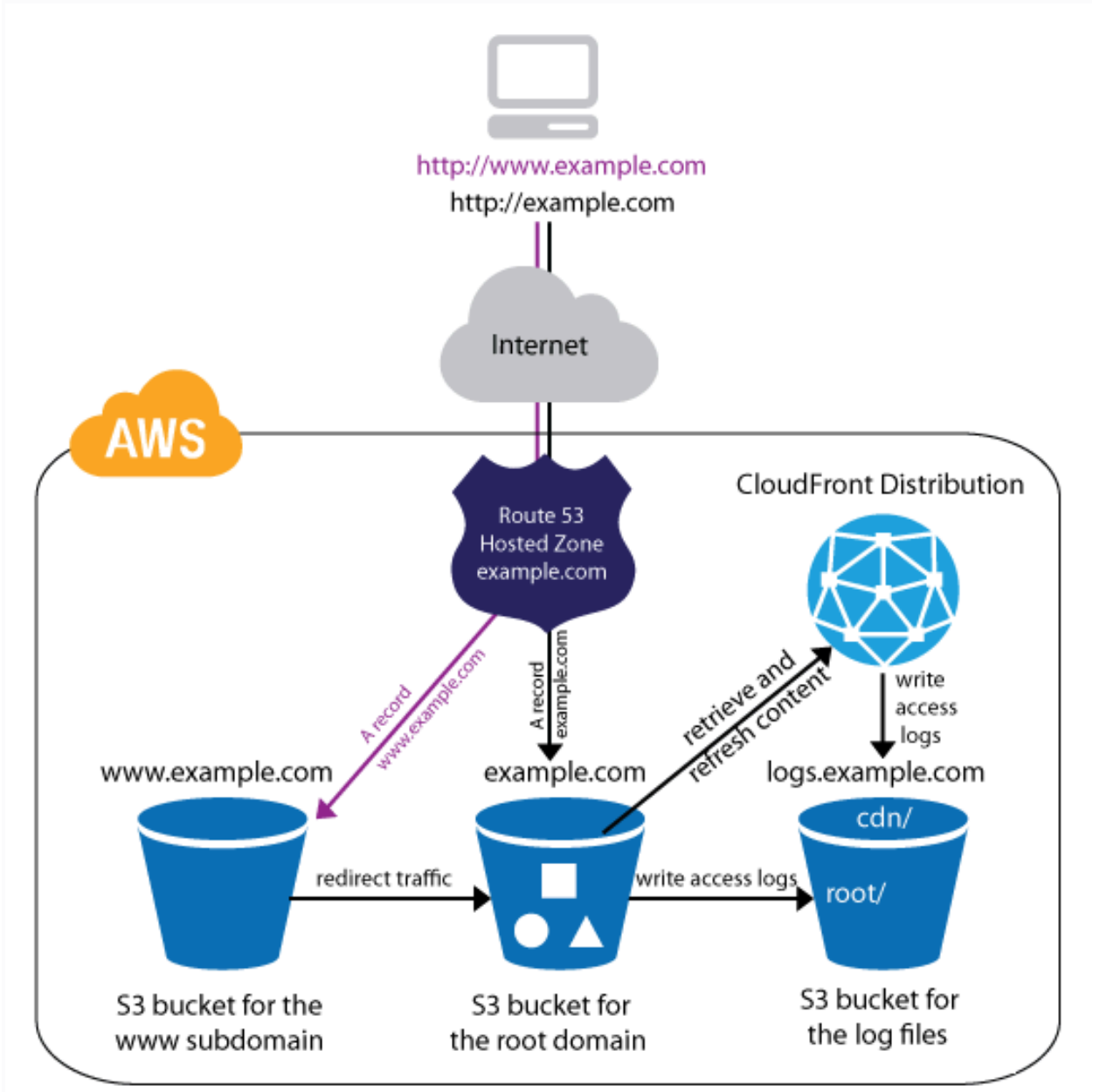
Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint. (Correct)

Host the website using an Nginx server in an EC2 instance. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to end-users.

Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator.

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Additionally, You can use Amazon S3 to host a static website. On a static website, individual webpages include static content. Amazon S3 is highly scalable and you only pay for what you use, you can start small and grow your application as you wish, with no compromise on performance or reliability.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. CloudFront can be integrated with Amazon S3 for fast delivery of data originating from an S3 bucket to your end-users. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.



In the scenario, Since we are only dealing with static content, we can leverage the web hosting feature of S3. Then we can improve the architecture further by integrating it with CloudFront. This way, users will be able to load both the web pages and images faster than if we hosted them on a webserver that we built from scratch.

Hence, the correct answer is: **Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.**

The option that says: **Host the website using an Nginx server in an EC2 instance. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images**

closer to end-users is incorrect. Creating your own web server to host a static website in AWS is a costly solution. Web Servers on an EC2 instance are usually used for hosting applications that require server-side processing (connecting to a database, data validation, etc.). Since static websites contain web pages with fixed content, we should use S3 website hosting instead.

The option that says: Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator is incorrect. This is how we serve static websites in the old days. Now, with the help of S3 website hosting, we can host our static contents from a durable, high-availability, and highly scalable environment without managing any servers. Hosting static websites in S3 is cheaper than hosting it in an EC2 instance. In addition, Using ASG for scaling instances that host a static website is an over-engineered solution that carries unnecessary costs. S3 automatically scales to high requests and you only pay for what you use.

The option that says: **Host the website in an AWS Elastic Beanstalk environment. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users** is incorrect. AWS Elastic Beanstalk simply sets up the infrastructure (EC2 instance, load balancer, auto-scaling group) for your application. It's a more expensive and a bit of an overkill solution for hosting a bunch of client-side files.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

Check out these Amazon S3 and CloudFront Cheat Sheets:

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

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