1. QUESTION

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

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

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

Correct

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

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

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

Key Features:

– Same low latency and high throughput performance of Standard

– Designed for durability of 99.999999999% of objects

– Designed for 99.9% availability over a given year

– Backed with the Amazon S3 Service Level Agreement for availability

– Supports SSL encryption of data in transit and at rest

– Lifecycle management for automatic migration of objects

Hence, the correct answers are:

– ***It is designed for data that is accessed less frequently.****

– ***It is designed for data that requires rapid access when needed.****

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

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

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

References:

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

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

Check out this Amazon S3 Cheat Sheet:

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

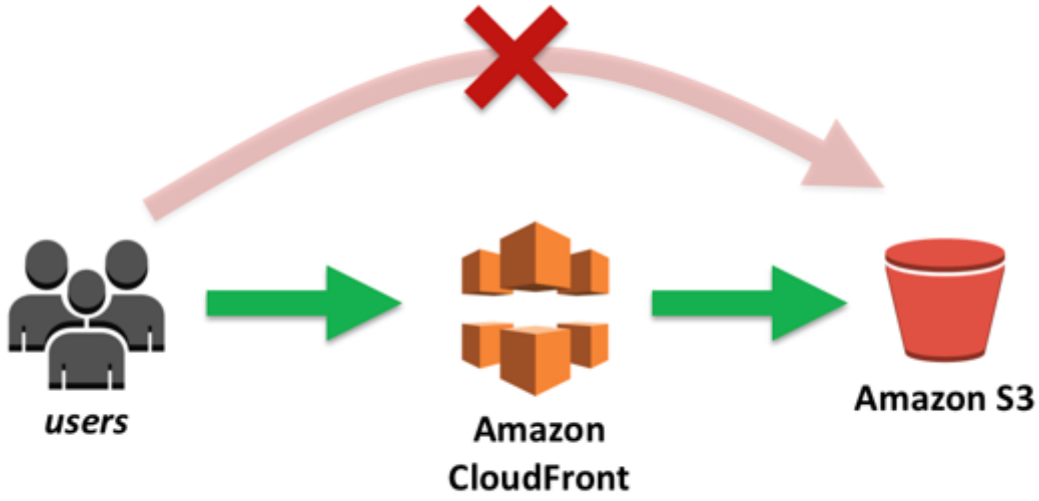
2. QUESTION

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

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

Correct

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



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

– Grant the CloudFront origin access identity the applicable permissions on the bucket.

– Deny access to anyone that you don't want to have access using Amazon S3 URLs.

Reference:

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

Check out this Amazon CloudFront Cheat Sheet:

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

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

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

Comparison of AWS Services Cheat Sheets:

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

3. **3. QUESTION**

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

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

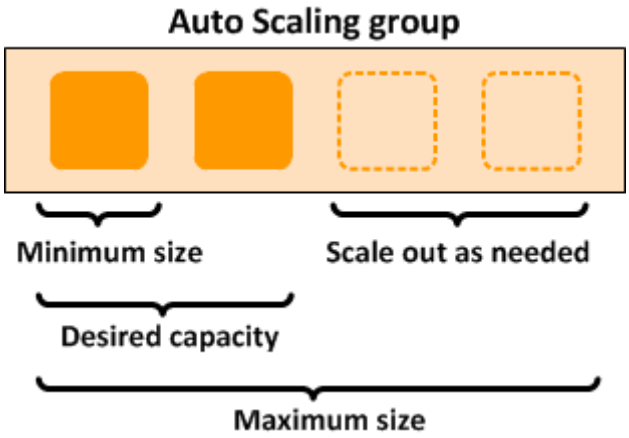
- Use simple scaling.

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

Incorrect

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

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



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

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

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

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

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

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

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

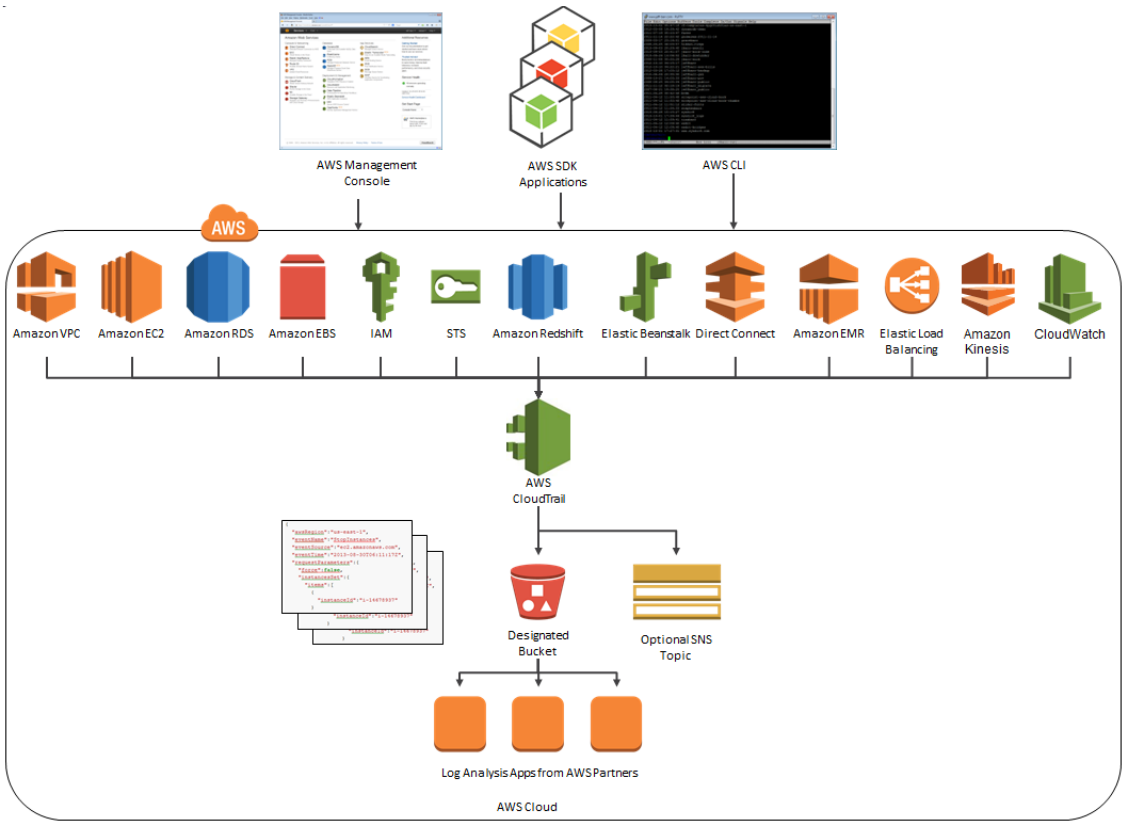
4. **4. QUESTION**

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

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

Incorrect

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



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

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

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

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

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

References:

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

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

Check out this AWS CloudTrail Cheat Sheet:

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

5. **5. QUESTION**

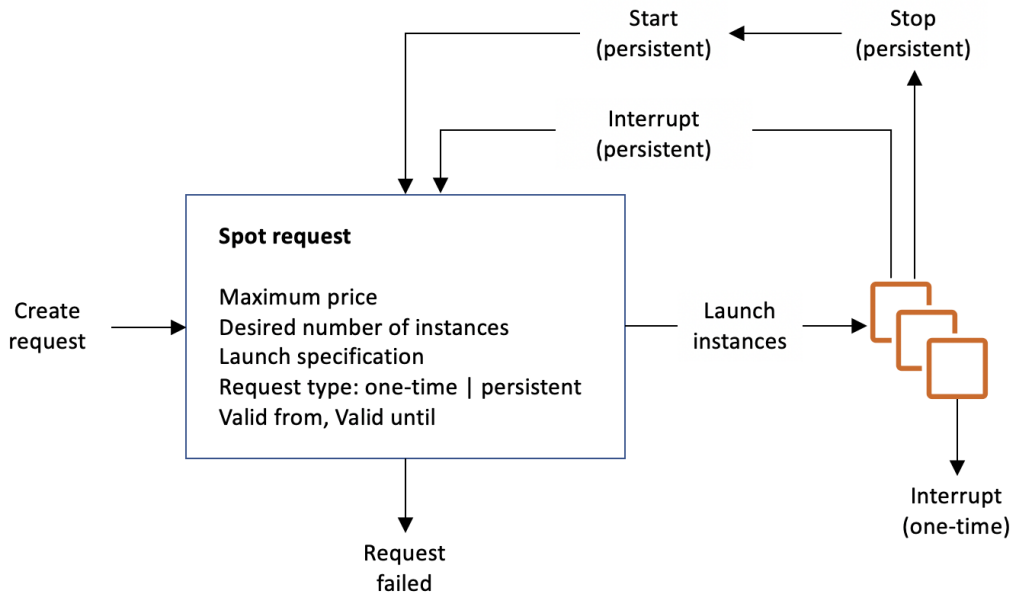
The media company that you are working for has a video transcoding application running on Amazon EC2. Each EC2 instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based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

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

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

Incorrect

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



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

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

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

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

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

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

References:

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

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

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

Check out this Amazon EC2 Cheat Sheet:

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

6. QUESTION

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

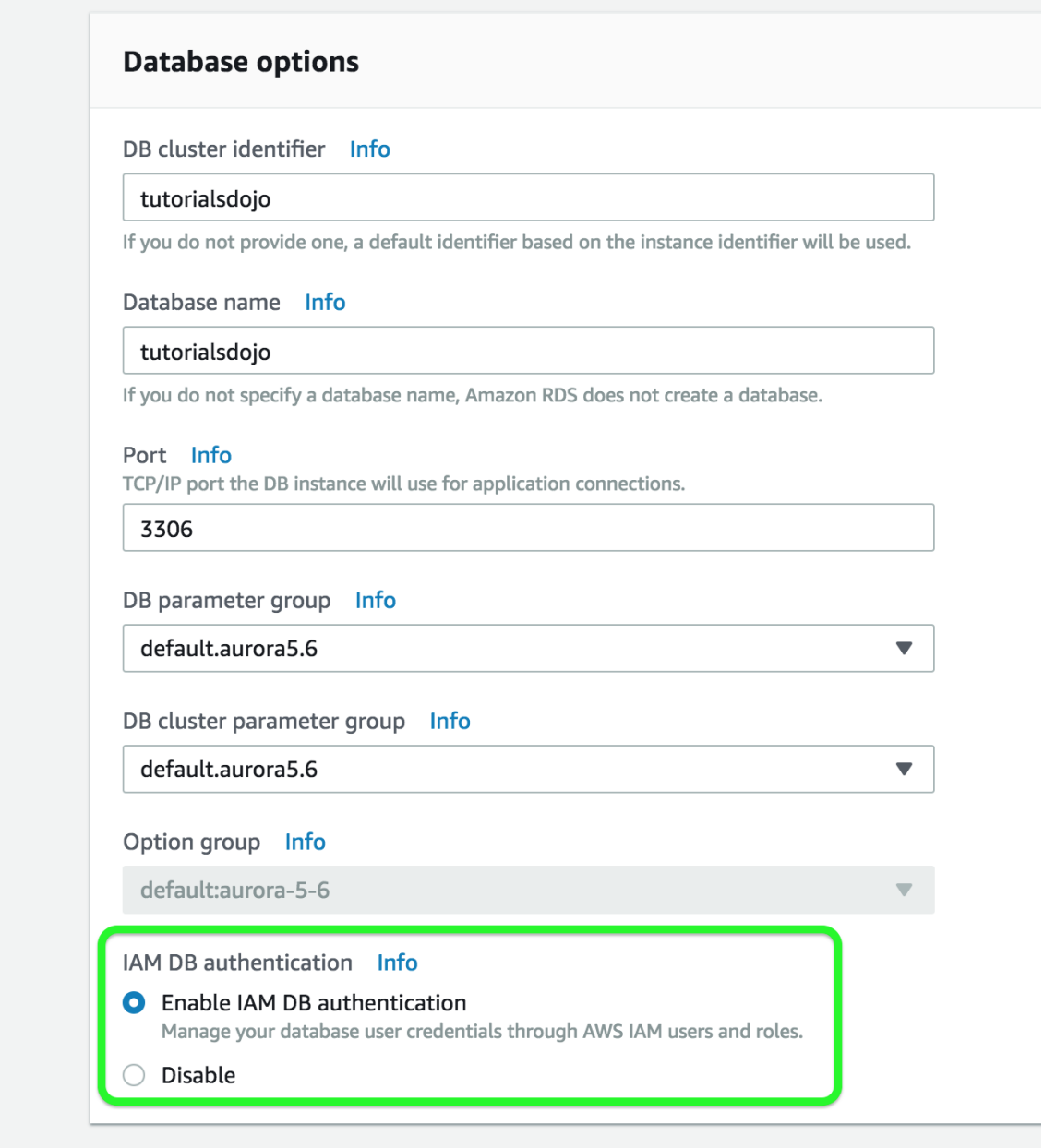
As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

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

Incorrect

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

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



IAM database authentication provides the following benefits:

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

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

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

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

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

Reference:

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

Check out this Amazon RDS cheat sheet:

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

6. **7. QUESTION**

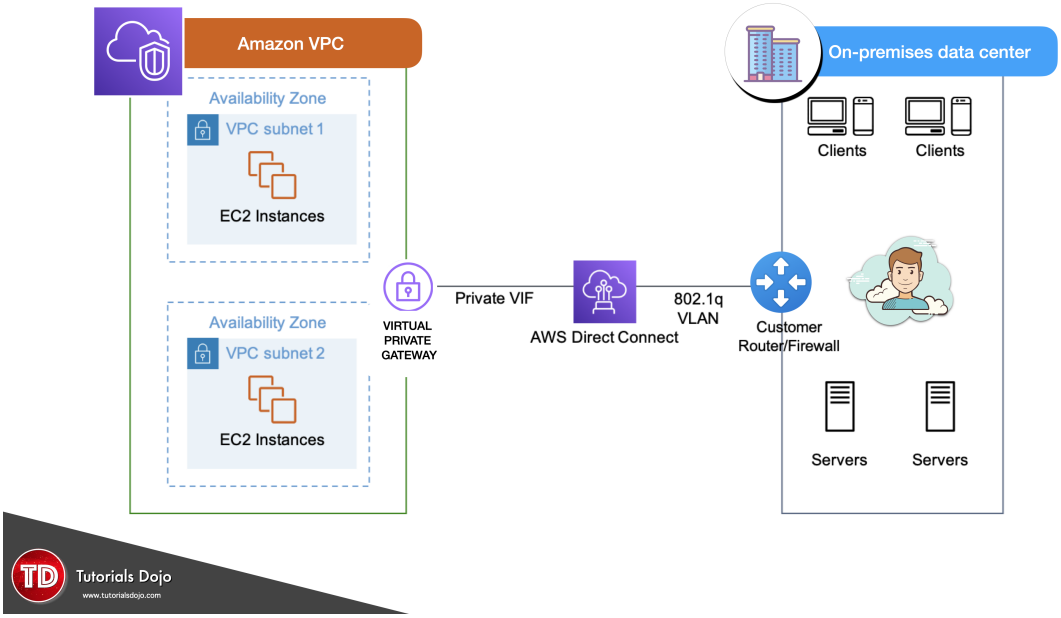
A company plans to implement a hybrid architecture. They need to create a dedicated connection from their Amazon Virtual Private Cloud (VPC) to their on-premises network. The connection must provide high bandwidth throughput and a more consistent network experience than Internet-based solutions.

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

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

Correct

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



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

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

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

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

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

References:

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

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

Check out this AWS Direct Connect Cheat Sheet:

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

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

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

Comparison of AWS Services Cheat Sheets:

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

8. QUESTION

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

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

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

Correct

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

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

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

– Change the versioning state of your bucket

– Permanently delete an object version

MFA Delete requires two forms of authentication together:

– Your security credentials

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

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

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

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

Reference:

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

Check out this Amazon S3 Cheat Sheet:

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

7. **9. QUESTION**

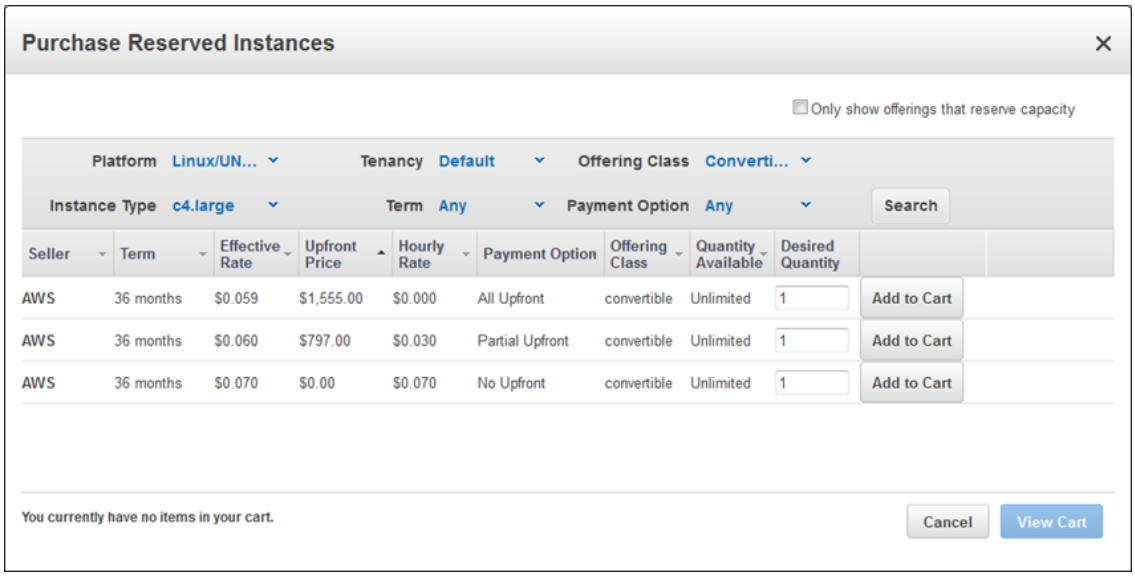
A company plans to set up a cloud infrastructure in AWS. In the planning, it was discussed that you need to deploy two EC2 instances that should continuously run for three years. The CPU utilization of the EC2 instances is also expected to be stable and predictable.

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

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

Correct

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



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

Reserved Instances are recommended for:

– Applications with steady state usage

– Applications that may require reserved capacity

– Customers that can commit to using EC2 over a 1 or 3 year term to reduce their total computing costs

References:

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

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

Check out this Amazon EC2 Cheat Sheet:

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

10. QUESTION

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

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

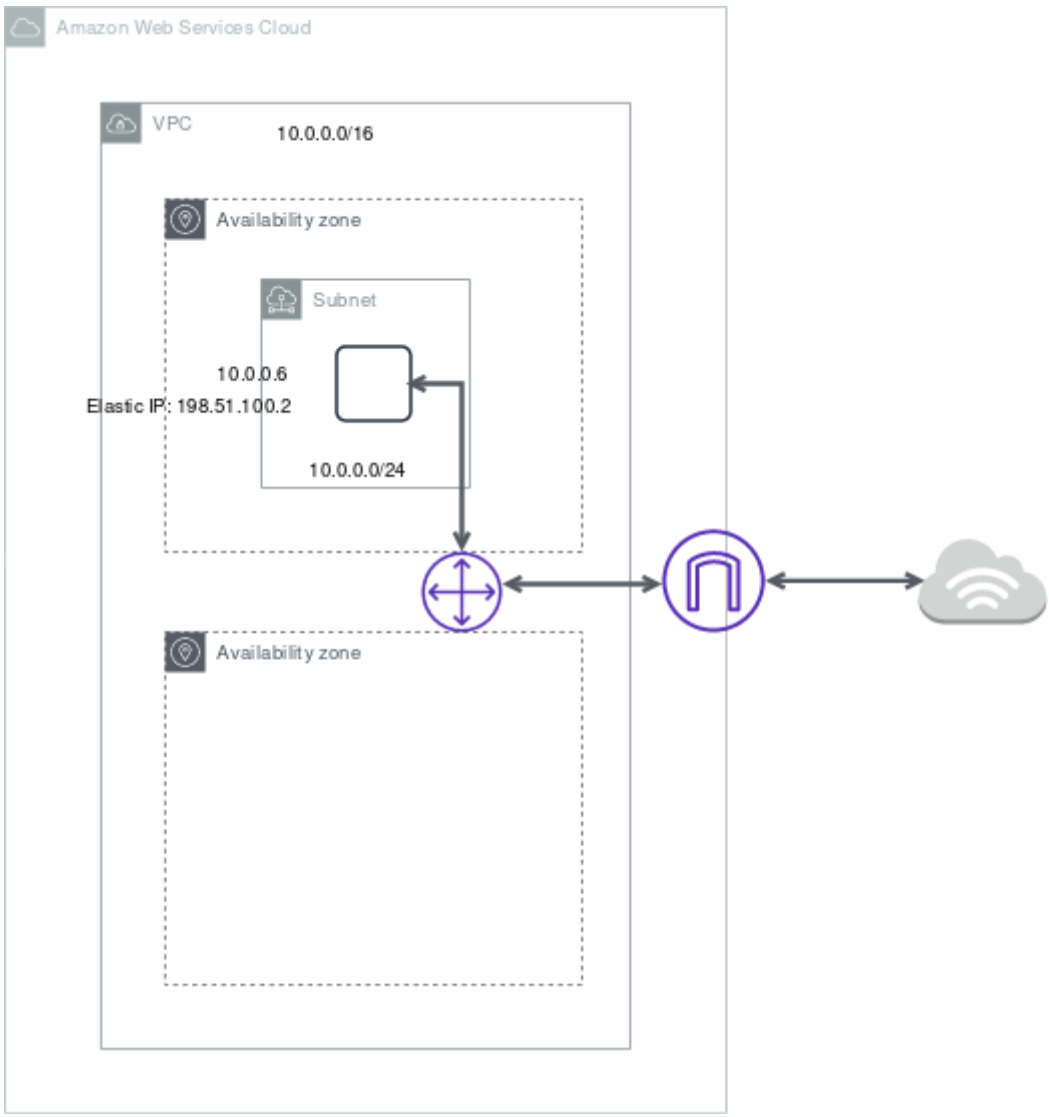
- The EC2 instance does not have a public IP address associated with it.
- The EC2 instance is running in an Availability Zone that is not connected to an Internet gateway.
- The route table is not properly configured to allow traffic to and from the Internet through the Internet gateway.
- The EC2 instance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

Incorrect

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

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

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



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

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

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

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

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

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

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

References:

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

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

Check out this Amazon VPC Cheat Sheet:

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

11. QUESTION

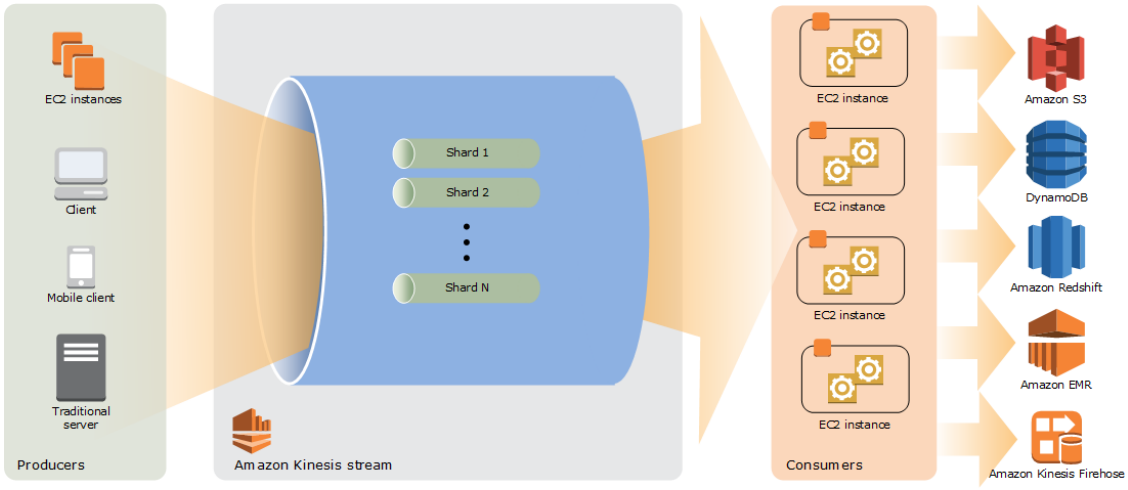
A startup is building IoT devices and monitoring applications. They are using IoT sensors to monitor the traffic in real-time by using an Amazon Kinesis Stream that is configured with default settings. It then sends the data to an Amazon S3 bucket every 3 days. When you checked the data in S3 on the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

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

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

Incorrect

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



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

The option that says: ***Amazon S3 bucket has encountered a data loss*** is incorrect because Amazon S3 rarely experiences data loss. Amazon has an SLA for S3 that it commits to its customers. Amazon S3 Standard, S3 Standard–IA, S3 One Zone-IA, and S3 Glacier are all designed to provide 99.999999999% durability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.000000001% of objects. Hence, Amazon S3 bucket data loss is highly unlikely.

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

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

Reference:

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

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

Check out this Amazon Kinesis Cheat Sheet:

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

12. QUESTION

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

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

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

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

Incorrect

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

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

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

– Use an Amazon CloudFront service for distributing both static and dynamic content.

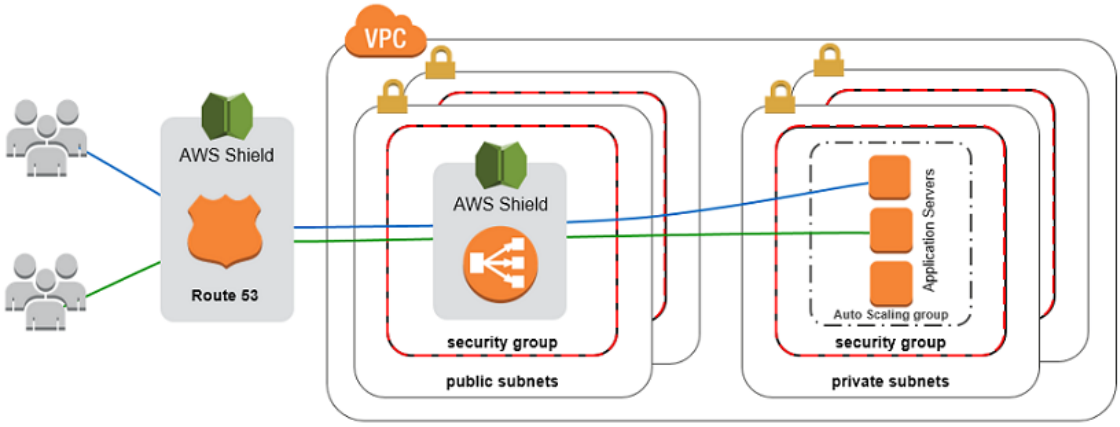
– Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by deploying it to a new private subnet.

– Set up alerts in Amazon CloudWatch to look for high `Network In` and CPU utilization metrics.

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

In addition, you can also use **AWS Shield** and **AWS WAF** to fortify your cloud network. AWS Shield is a managed DDoS protection service that is available in two tiers: Standard and Advanced. AWS Shield Standard applies always-on detection and inline mitigation techniques, such as deterministic packet filtering and priority-based traffic shaping, to minimize application downtime and latency.

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



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

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

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

– Use an Amazon CloudFront service for distributing both static and dynamic content.

– Use an Application Load Balancer with Auto Scaling groups for your EC2 instances. Prevent direct Internet traffic to your Amazon RDS database by deploying it to a new private subnet.

– Use AWS Shield and AWS WAF.

References:

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

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

Best practices on DDoS Attack Mitigation:

13. QUESTION

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

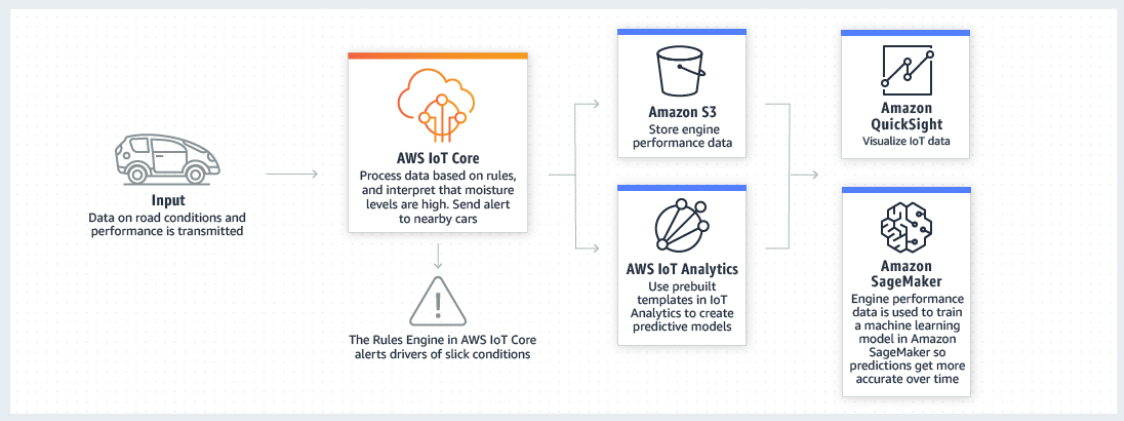
Which AWS service will allow the developers of the agency to connect the smart devices to the cloud-based applications?

- AWS CloudFormation

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

Incorrect

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



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

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

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

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

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

References:

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

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

14. QUESTION

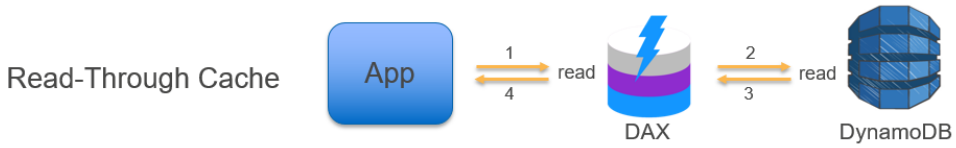
A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time.

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

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

    Incorrect

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

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

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

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

    References:

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

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

    Check out this Amazon DynamoDB Cheat Sheet:

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

15. QUESTION

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

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

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

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

Incorrect

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

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

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

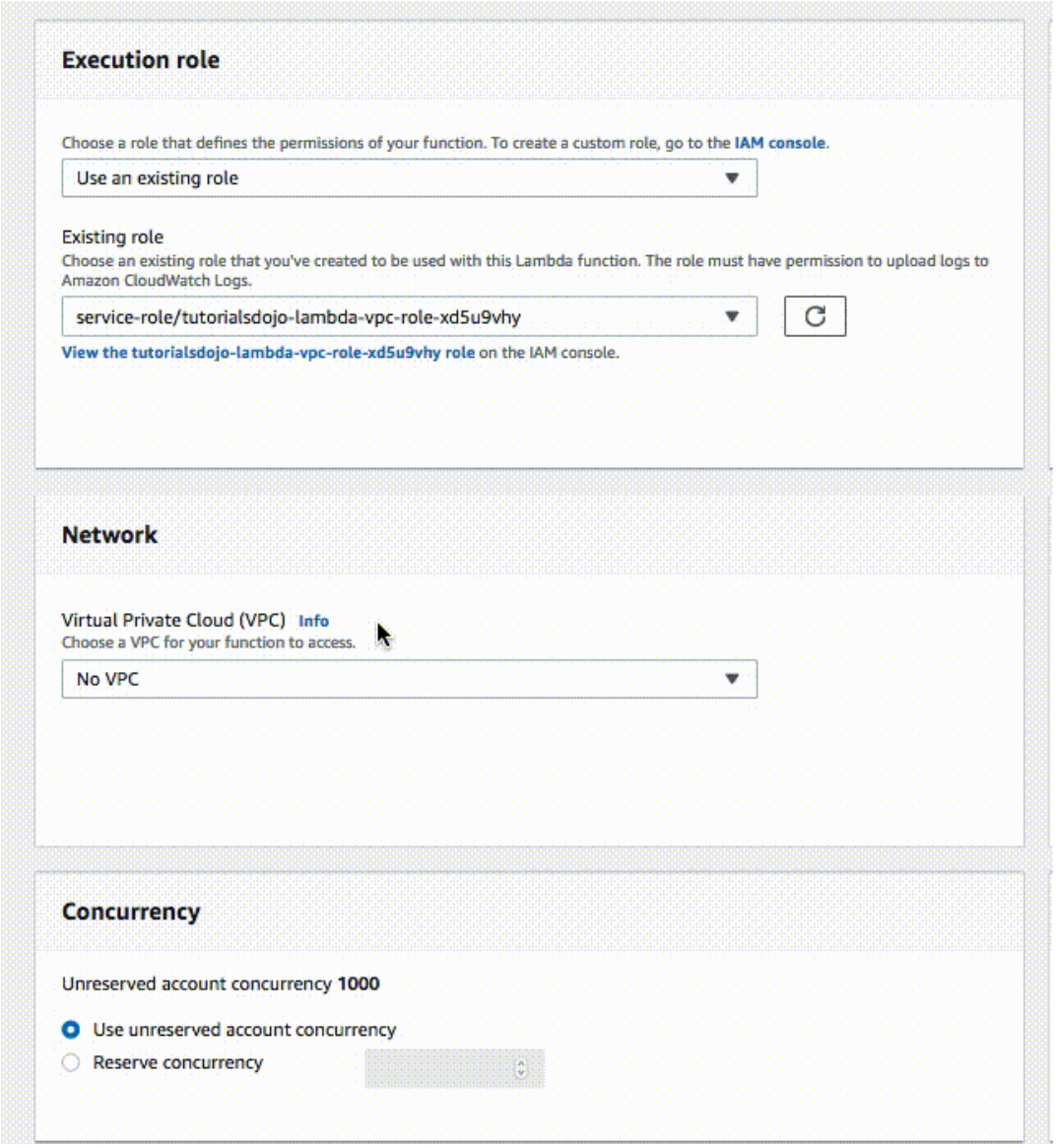
Your Lambda function automatically scales based on the number of events it processes. If your Lambda function accesses a VPC, you must make sure that your VPC has sufficient ENI capacity to support the scale requirements of your Lambda function. It is also recommended that you specify at least one subnet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

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

Hence, the correct answers for this scenario are:

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

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



The 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/

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

1. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Correct

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

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

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

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

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

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

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

References:

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

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

AWS Storage Gateway Overview:

Check out this AWS Storage Gateway Cheat Sheet:

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

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

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

2. QUESTION

Category: CSAA – Design High-Performing Architectures

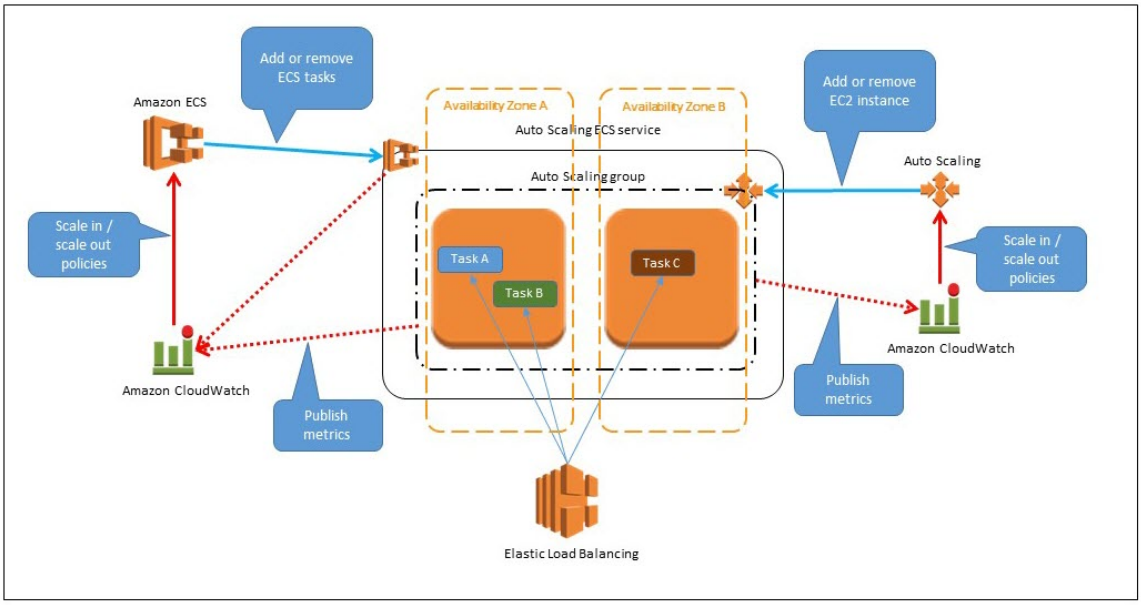
A loan processing application is hosted in a single On-Demand EC2 instance in your VPC. To improve the scalability of your application, you have to use Auto Scaling to automatically add new EC2 instances to handle a surge of incoming requests.

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

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

Incorrect

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



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

– The instance is in the `running` state.

– The AMI used to launch the instance must still exist.

– The instance is not a member of another Auto Scaling group.

– The instance is launched into one of the Availability Zones defined in your Auto Scaling group.

– If the Auto Scaling group has an attached load balancer, the instance and the load balancer must both be in EC2-Classic or the same VPC. If the Auto Scaling group has an attached target group, the instance and the load balancer must both be in the same VPC.

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

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

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

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

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

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

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

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

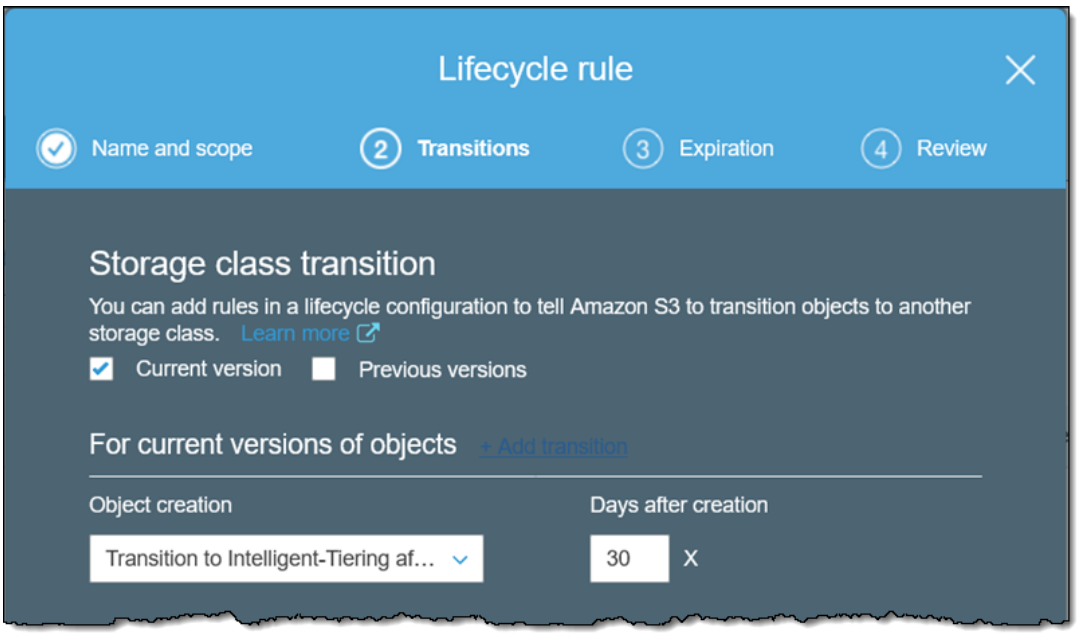
As a Solutions Architect, what should you do if the company needs to automatically transfer or archive media data from an S3 bucket to Glacier?

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

Correct

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

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



These actions can be classified as follows:

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

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

Reference:

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

Check out this Amazon S3 Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Resilient Architectures

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

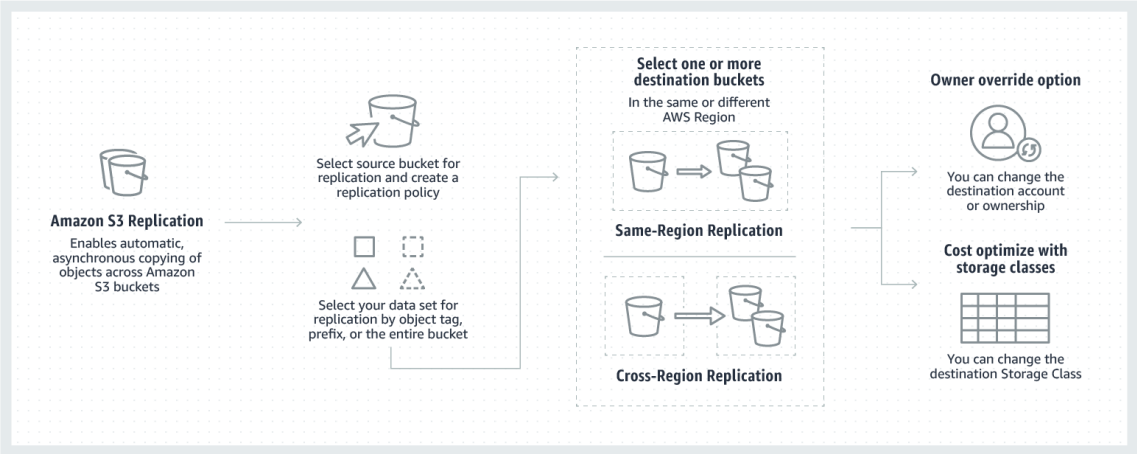
- In order to use the Cross-Region Replication feature in S3, you need to first enable versioning on the bucket.
- This is a premium feature which is only for AWS Enterprise accounts.
- The Cross-Region Replication feature is only available for Amazon S3 – One Zone-IA
- The Cross-Region Replication feature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

Correct

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

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

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



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

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

References:

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

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

Check out this Amazon S3 Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

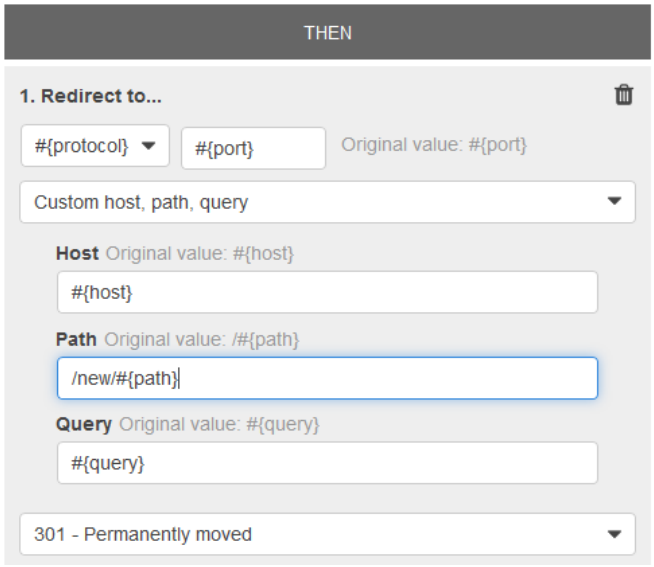
- Classic Load Balancer
- Network Load Balancer
- Application Load Balancer
- Either a Classic Load Balancer or a Network Load Balancer

Correct

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

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

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



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

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

References:

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

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

AWS Elastic Load Balancing Overview:

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

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

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

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

6. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

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

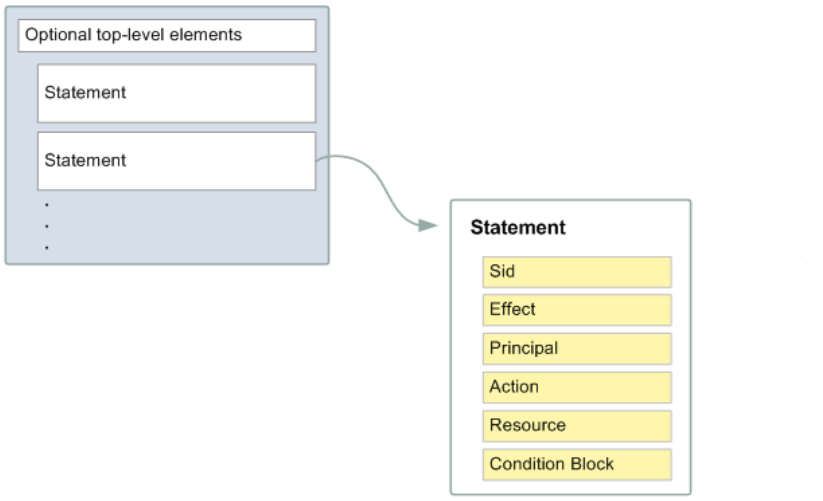
```
            "IpAddress": {
              "aws:SourceIp": "187.5.104.11/32"
            }
         }
      }
   ]
}
```

Which of the following options are allowed by this policy?

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

Correct

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



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

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

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

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

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

References:

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

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

Check out this AWS IAM Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

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

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

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

Incorrect

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

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

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

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

References:

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

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

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

Check out this Amazon EC2 Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Incorrect

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

FEATURES	SSD Solid State Drive	HDD Hard Disk Drive
Best for workloads with:	*small, random* I/O operations	*large, sequential* I/O operations
Can be used as a bootable volume?	Yes	No
Suitable Use Cases	- Best for transactional workloads - Critical business applications that require sustained IOPS performance - Large database workloads such as MongoDB, Oracle, Microsoft SQL Server and many others…	- Best for *large streaming workloads* requiring consistent, fast throughput at a low price - Big data, Data warehouses, Log processing - Throughput-oriented storage for large volumes of data that is *infrequently* accessed
Cost	moderate / high	low
Dominant Performance Attribute	IOPS	Throughput (MiB/s)

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

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

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

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

Reference:

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

Amazon EBS Overview – SSD vs HDD:

Check out this Amazon EBS Cheat Sheet:

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

9. QUESTION

Category: CSAA – Design Secure Applications and Architectures

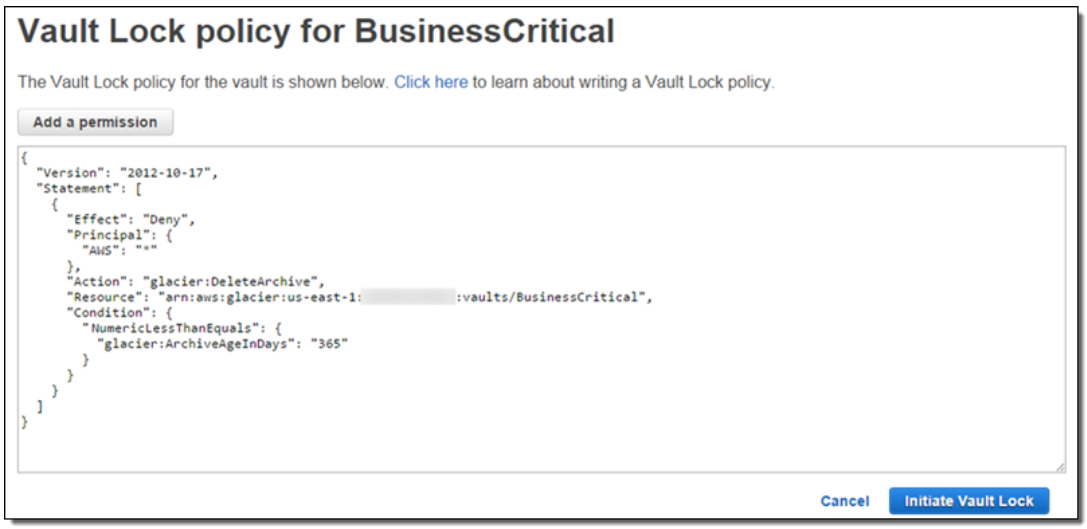
A Solutions Architect is designing a monitoring application which generates audit logs of all operational activities of the company's cloud infrastructure. Their IT Security and Compliance team mandates that the application retain the logs for 5 years before the data can be deleted.

How can the Architect meet the above requirement?

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

Correct

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



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

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

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

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

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

References:

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

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

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

Amazon S3 and S3 Glacier Overview:

Check out this Amazon S3 Glacier Cheat Sheet:

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

10. QUESTION

Category:CSAA – Design High-Performing Architectures

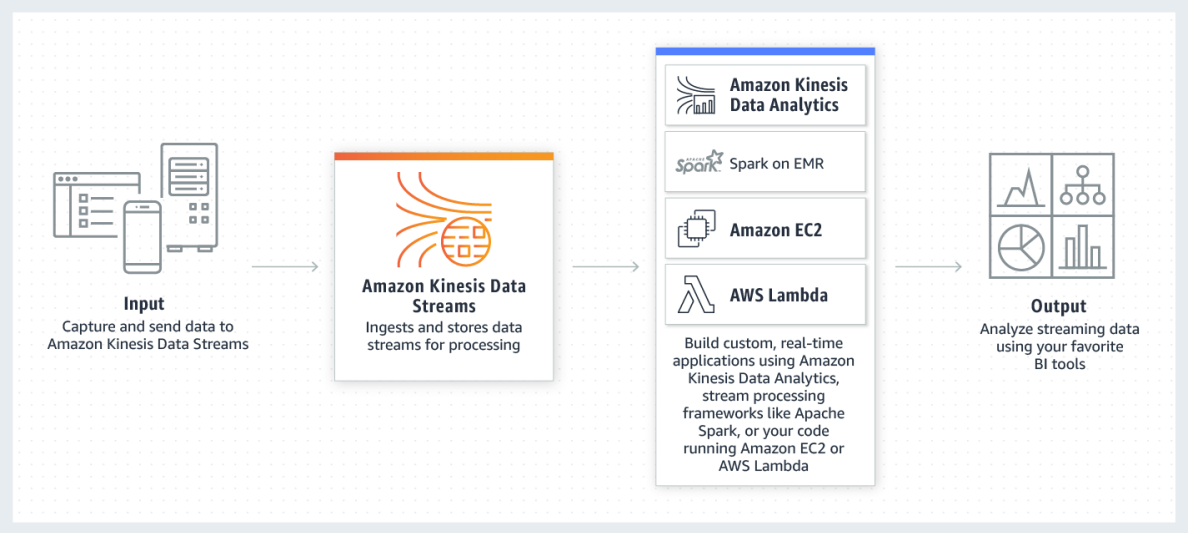
A company plans to develop a custom messaging service that will also be used to train their AI for an automatic response feature which they plan to implement in the future. Based on their research and tests, the service can receive up to thousands of messages a day, and all of these data are to be sent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

Which of the following options can satisfy the given requirement?

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

Incorrect

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



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

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

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

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

References:

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

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

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

Check out this Amazon Kinesis Cheat Sheet:

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

11. QUESTION

Category: CSAA – Design High-Performing Architectures

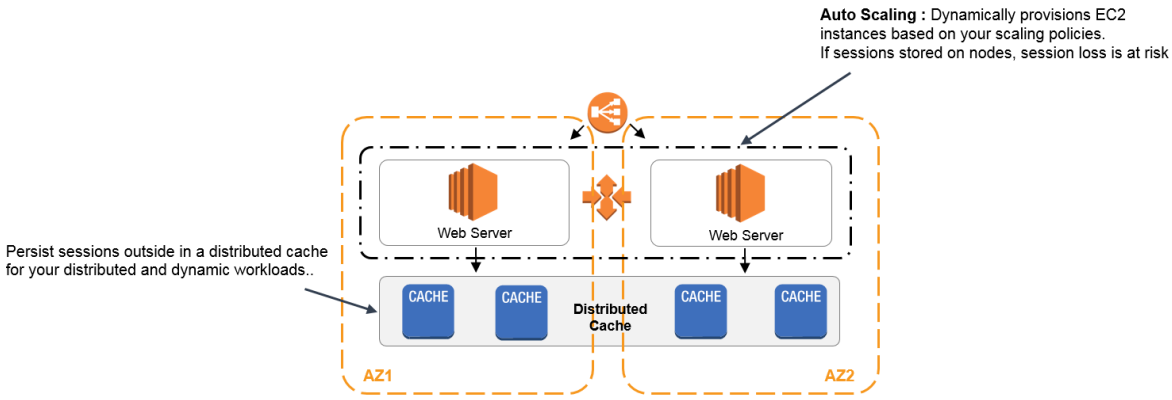
A company has a fleet of running Spot EC2 instances behind an Application Load Balancer. The incoming traffic comes from various users across multiple AWS regions and you would like to have the user's session shared among the fleet of instances. You are required to set up a distributed session management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

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

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

Incorrect

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



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

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

References:

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

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

Check out this Amazon Elasticache Cheat Sheet:

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

Redis (cluster mode enabled vs disabled) vs Memcached:

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

12. QUESTION

Category: CSAA – Design Resilient Architectures

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

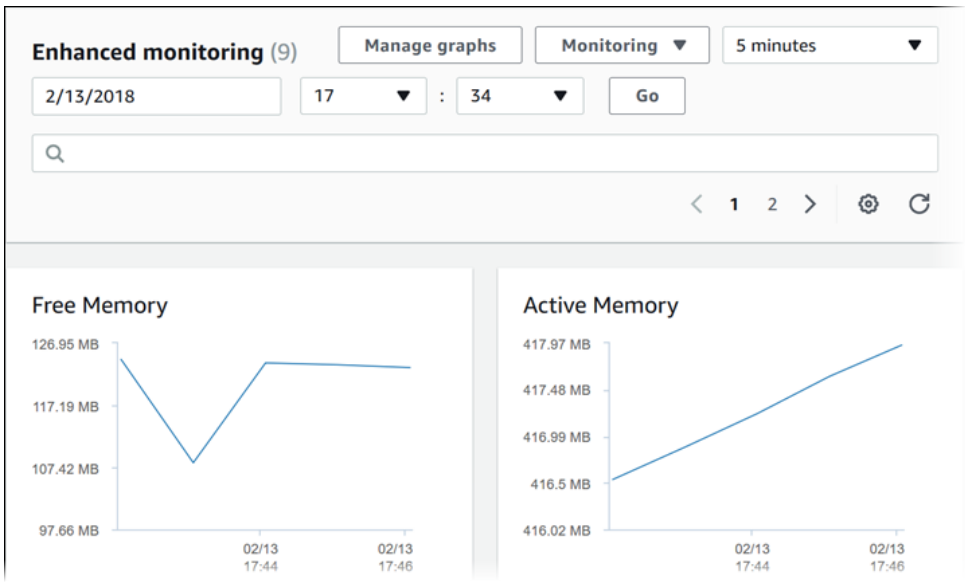
What are the enhanced monitoring metrics that Amazon CloudWatch gathers from Amazon RDS DB instances which provide more accurate information? (Select TWO.)

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

Incorrect

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

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



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

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

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

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

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

References:

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

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

Check out this Amazon CloudWatch Cheat Sheet:

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

Check out this Amazon RDS Cheat Sheet:

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

13. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

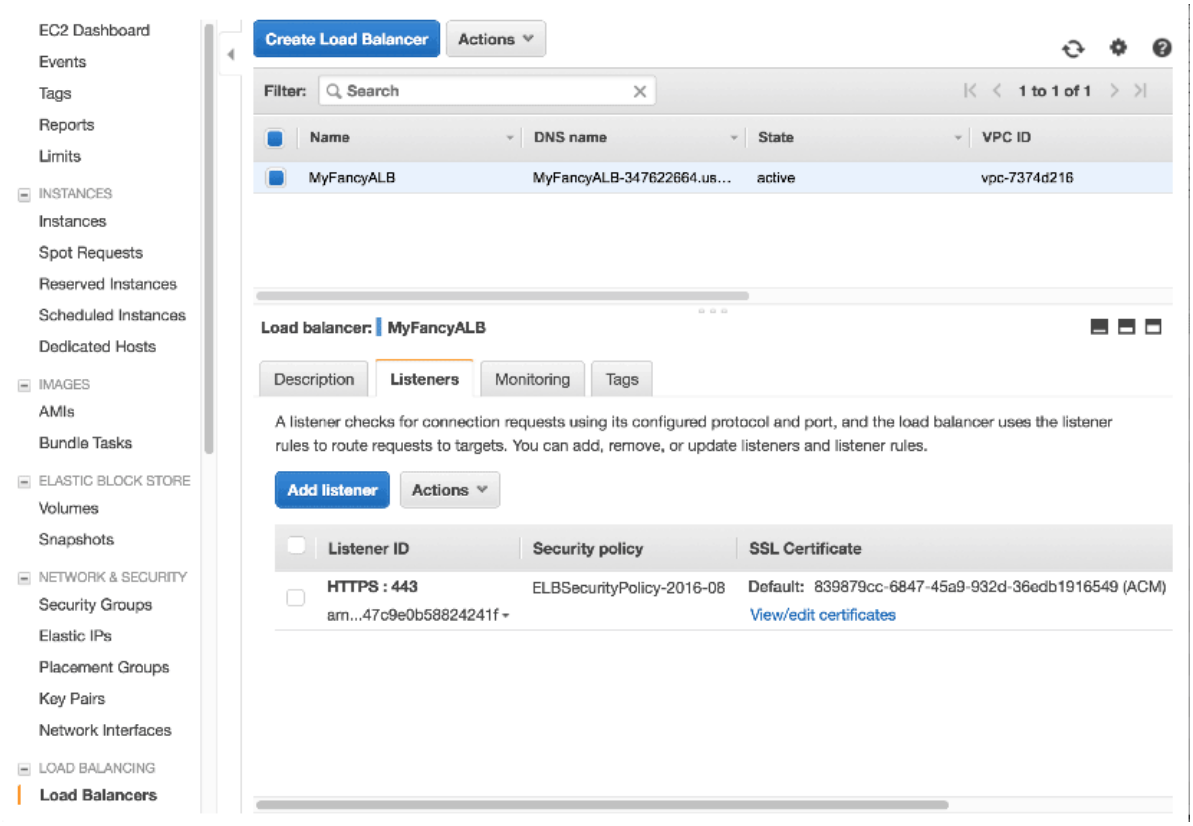
Which of the following is the most cost-effective solution to meet the above requirement?

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

Correct

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

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



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

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

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

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

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

References:

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

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

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

Check out this Amazon CloudFront Cheat Sheet:

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

SNI Custom SSL vs Dedicated IP Custom SSL:

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

Comparison of AWS Services Cheat Sheets:

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

14. QUESTION

Category: CSAA – Design Secure Applications and Architectures

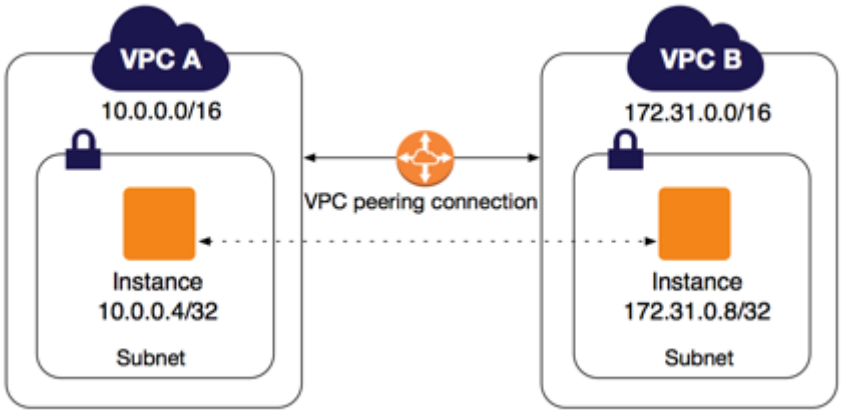
An operations team has an application running on EC2 instances inside two custom VPCs. The VPCs are located in the Ohio and N.Virginia Region respectively. The team wants to transfer data between the instances without traversing the public internet.

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

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

Incorrect

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



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

Hence, the correct answers are:

– Set up a VPC peering connection between the VPCs.

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

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

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

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

References:

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

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

Check out this Amazon VPC Cheat Sheet:

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

15. QUESTION

Category: CSAA – Design Secure Applications and Architectures

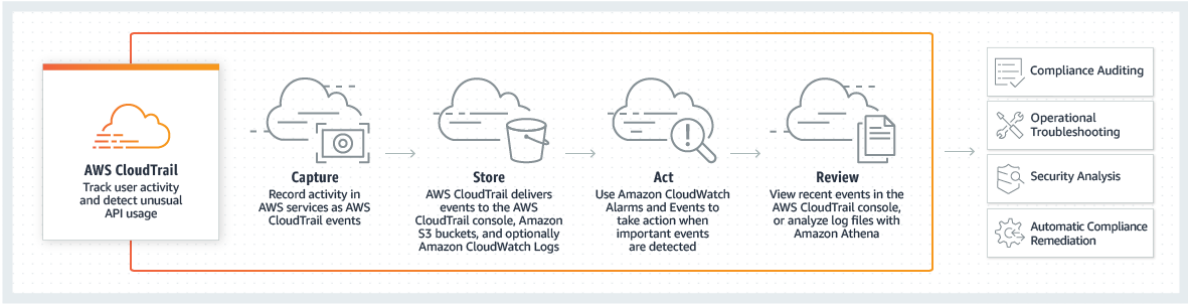
A startup has resources deployed on the AWS Cloud. It is now going through a set of scheduled audits by an external auditing firm for compliance.

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

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

Incorrect

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



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

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

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

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

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

Reference:

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

Check out this AWS CloudTrail Cheat Sheet:

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

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

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

1. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Correct

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

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

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

2. QUESTION

Category: CSAA – Design High-Performing Architectures

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

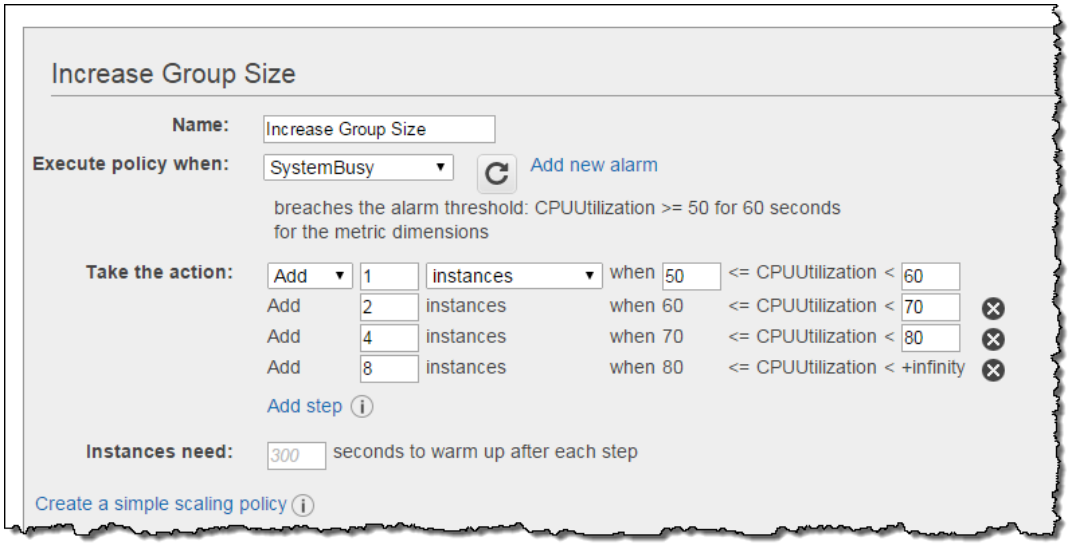
Which of the following is the most suitable type of scaling policy that you should use?

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

Correct

With step scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process as well as define how your scalable target should be scaled when a threshold is 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

3. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Correct

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

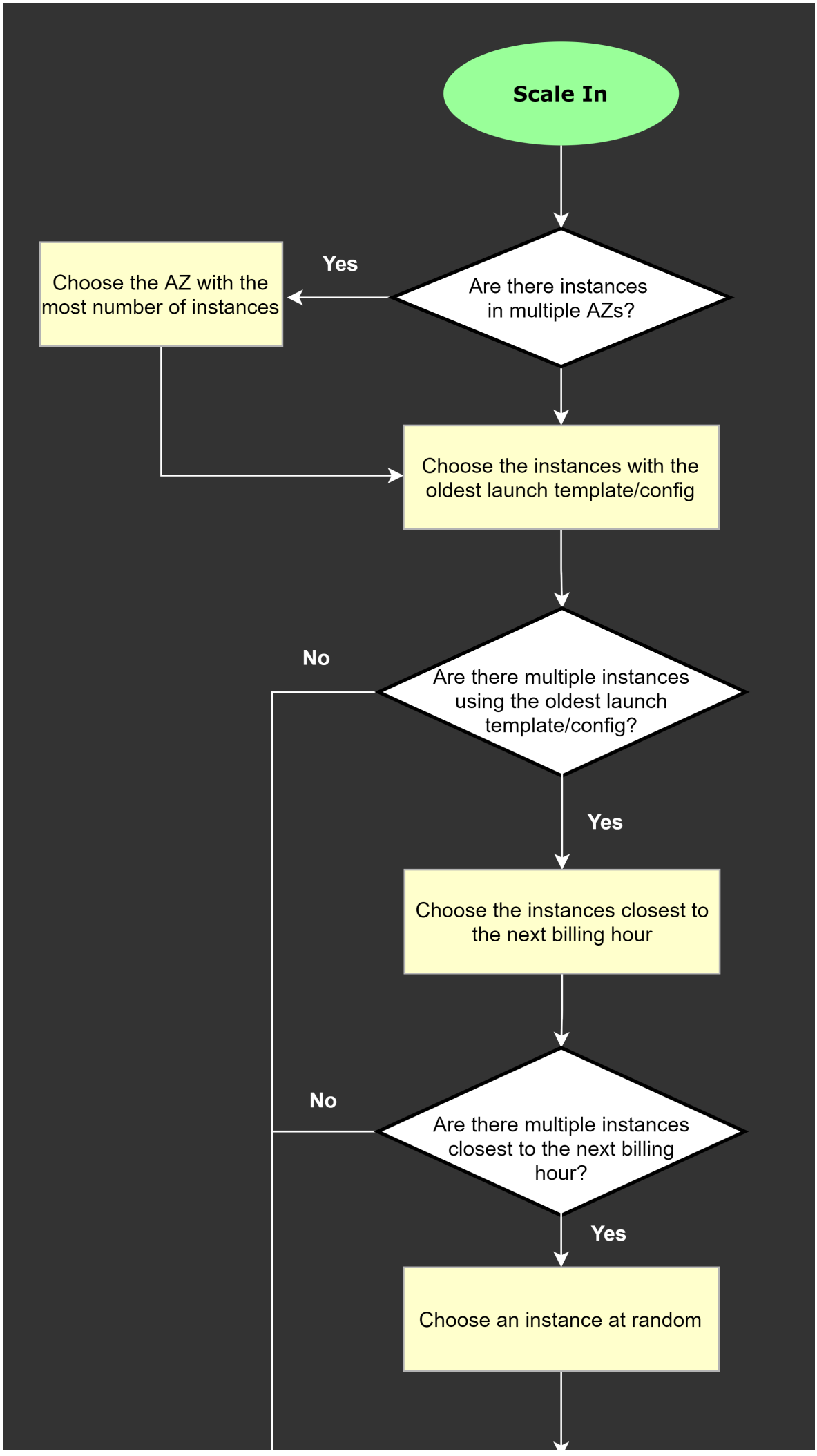
\1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that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

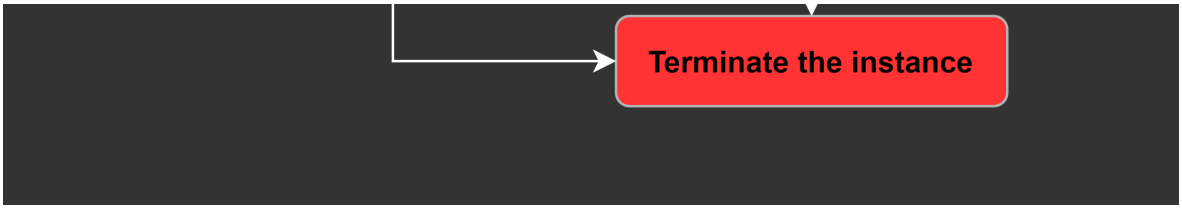
\2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.

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

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

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


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



Reference:

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

Check out this AWS Auto Scaling Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Resilient Architectures

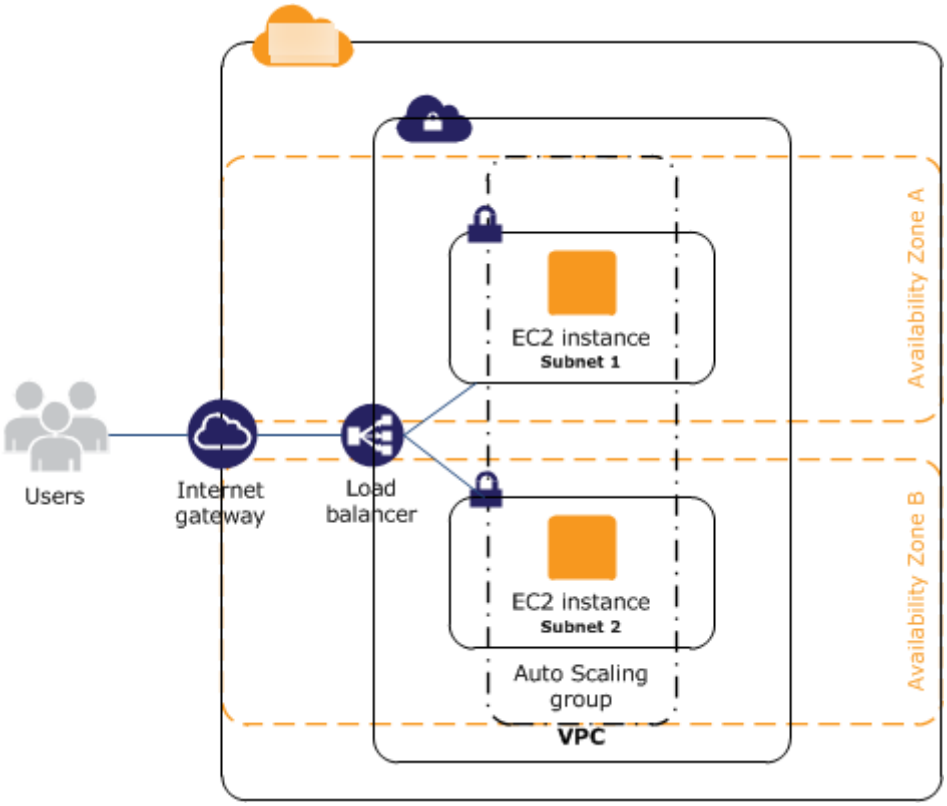
A major TV network has a web application running on eight Amazon T3 EC2 instances. The number of requests that the application processes are consistent and do not experience spikes. To ensure that eight instances are running at all times, the Solutions Architect should create an Auto Scaling group and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

Which of the following options can satisfy the given requirements?

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

Incorrect

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



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

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

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

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

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

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

References:

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

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

AWS Elastic Load Balancing Overview:

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

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

5. QUESTION

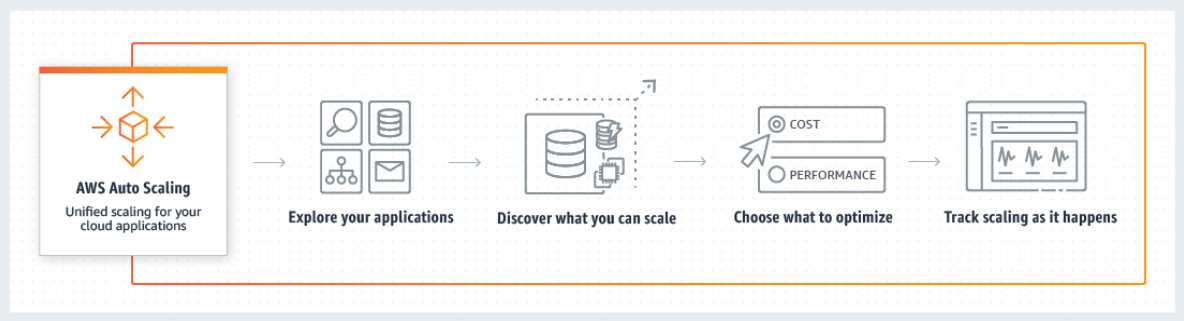
Category: CSAA – Design Resilient Architectures

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

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

Incorrect

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



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

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

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

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

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

References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Correct

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

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

The following options are incorrect:

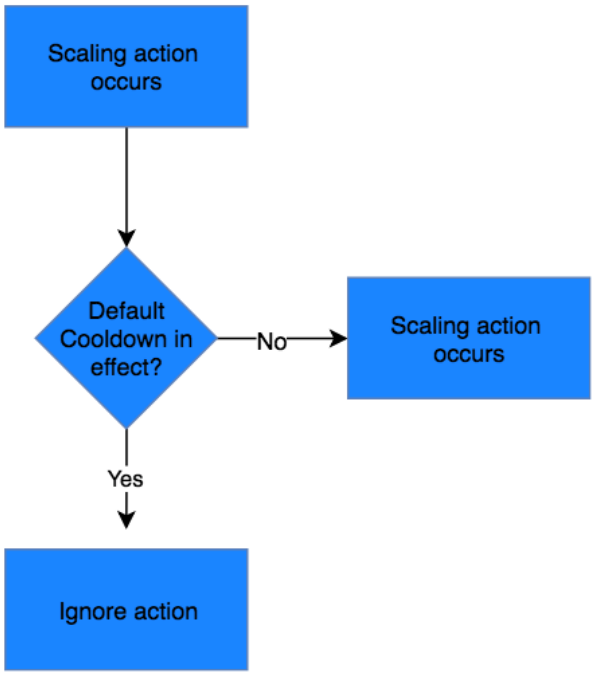
– It ensures that before the Auto Scaling group scales out, the EC2 instances have ample time to cooldown.

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

– Its default value is 600 seconds.

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

The figure below demonstrates the scaling cooldown:



Reference:

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

Check out this AWS Auto Scaling Cheat Sheet:

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

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

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

7. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Incorrect

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

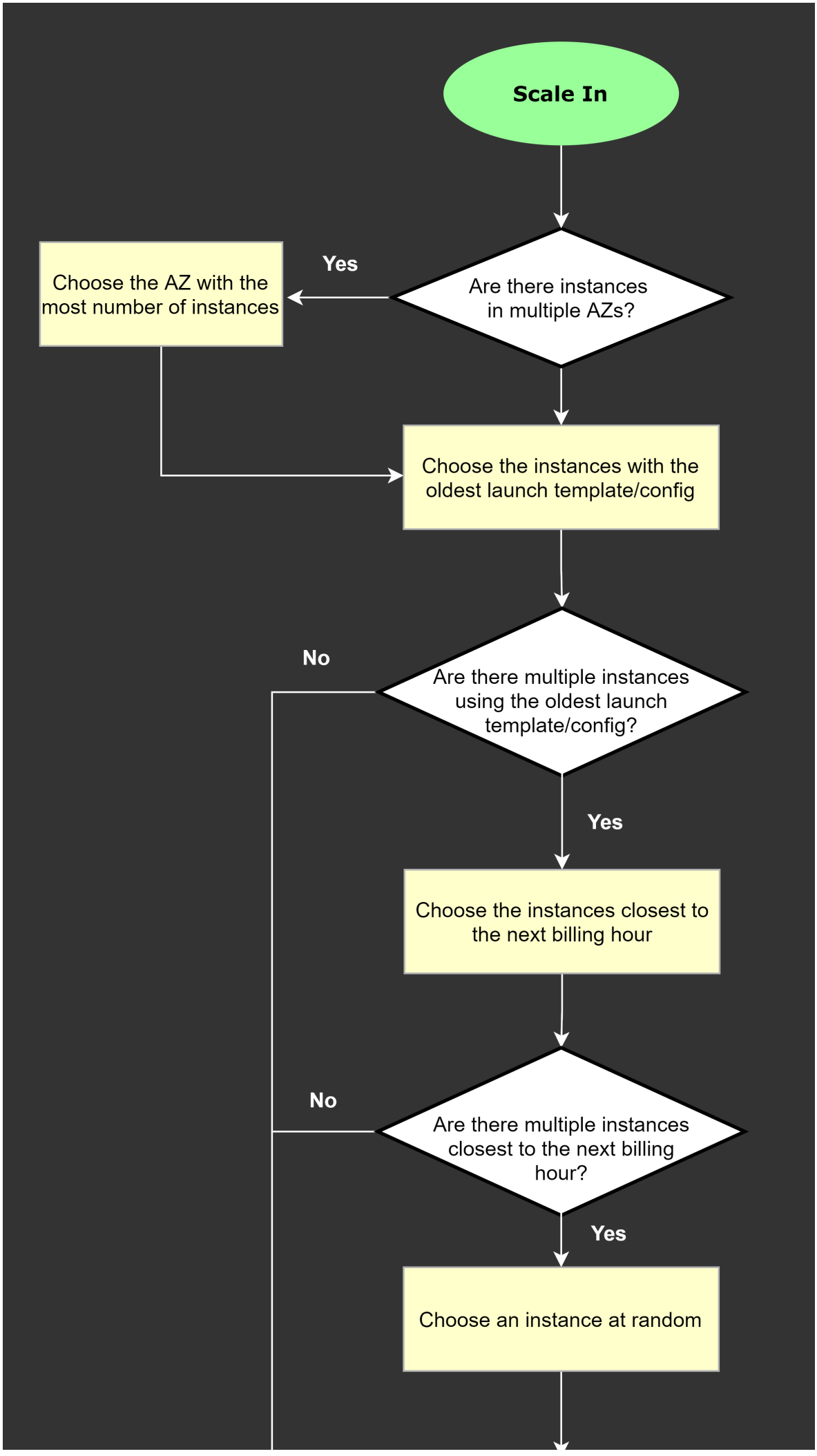
\1. If there are instances in multiple Availability Zones, choose the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, choose the Availability Zone with the instances that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

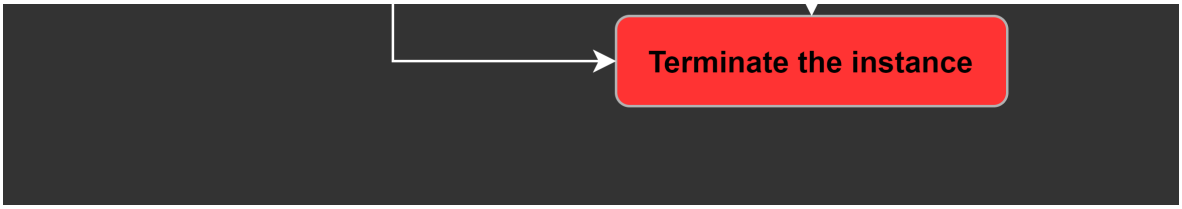
\2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.

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

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

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


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



References:

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

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

Check out this AWS Auto Scaling Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design High-Performing Architectures

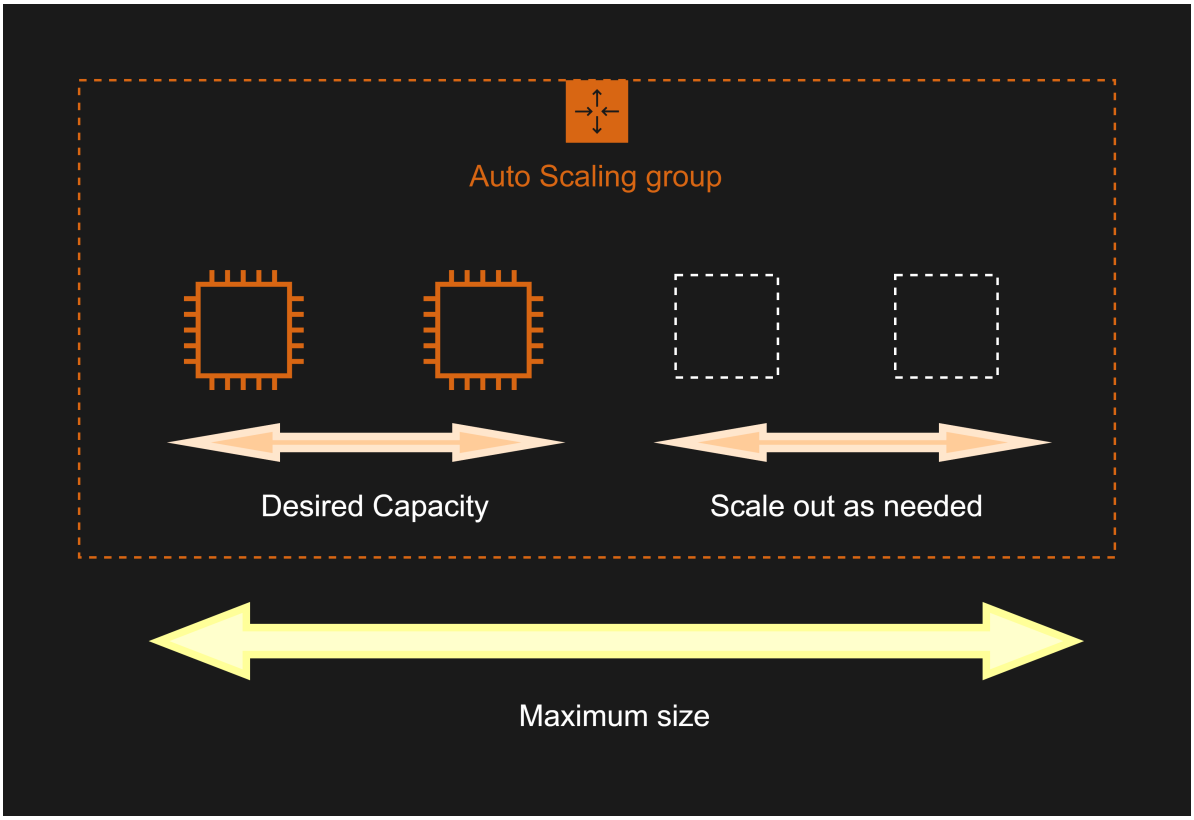
A tech company has a CRM application hosted on an Auto Scaling group of On-Demand EC2 instances. The application is extensively used during office hours from 9 in the morning till 5 in the afternoon. Their users are complaining that the performance of the application is slow during the start of the day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

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

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

Correct

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



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

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

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

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

Reference:

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

Check out this AWS Auto Scaling Cheat Sheet:

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

1. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Incorrect

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

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

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

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

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

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

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

– It normalizes data and stores it on multiple tables that require multiple queries to write to disk.

– It generally incurs the performance costs of an ACID-compliant transaction system.

– It uses expensive joins to reassemble required views of query results.

For DynamoDB, it scales well due to these reasons:

– Its **schema flexibility** lets DynamoDB store complex hierarchical data within a single item. DynamoDB is not a totally *schemaless* database since the very definition of a schema is just the model or structure of your data.

– Composite key design lets it store related items close together on the same table.

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

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

References:

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

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

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

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

Check out this Amazon DynamoDB Cheat Sheet:

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

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

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

2. QUESTION

Category: CSAA – Design High-Performing Architectures

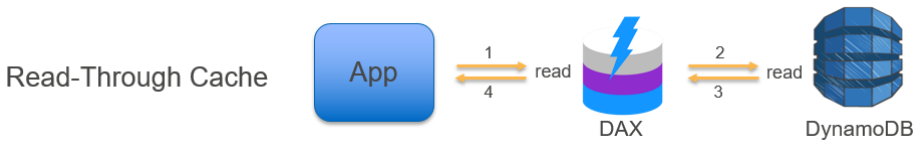
A popular mobile game uses CloudFront, Lambda, and DynamoDB for its backend services. The player data is persisted on a DynamoDB table and the static assets are distributed by CloudFront. However, there are a lot of complaints that saving and retrieving player information is taking a lot of time.

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

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

Correct

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



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

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

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

References:

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

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

Check out this Amazon DynamoDB Cheat Sheet:

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

3. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

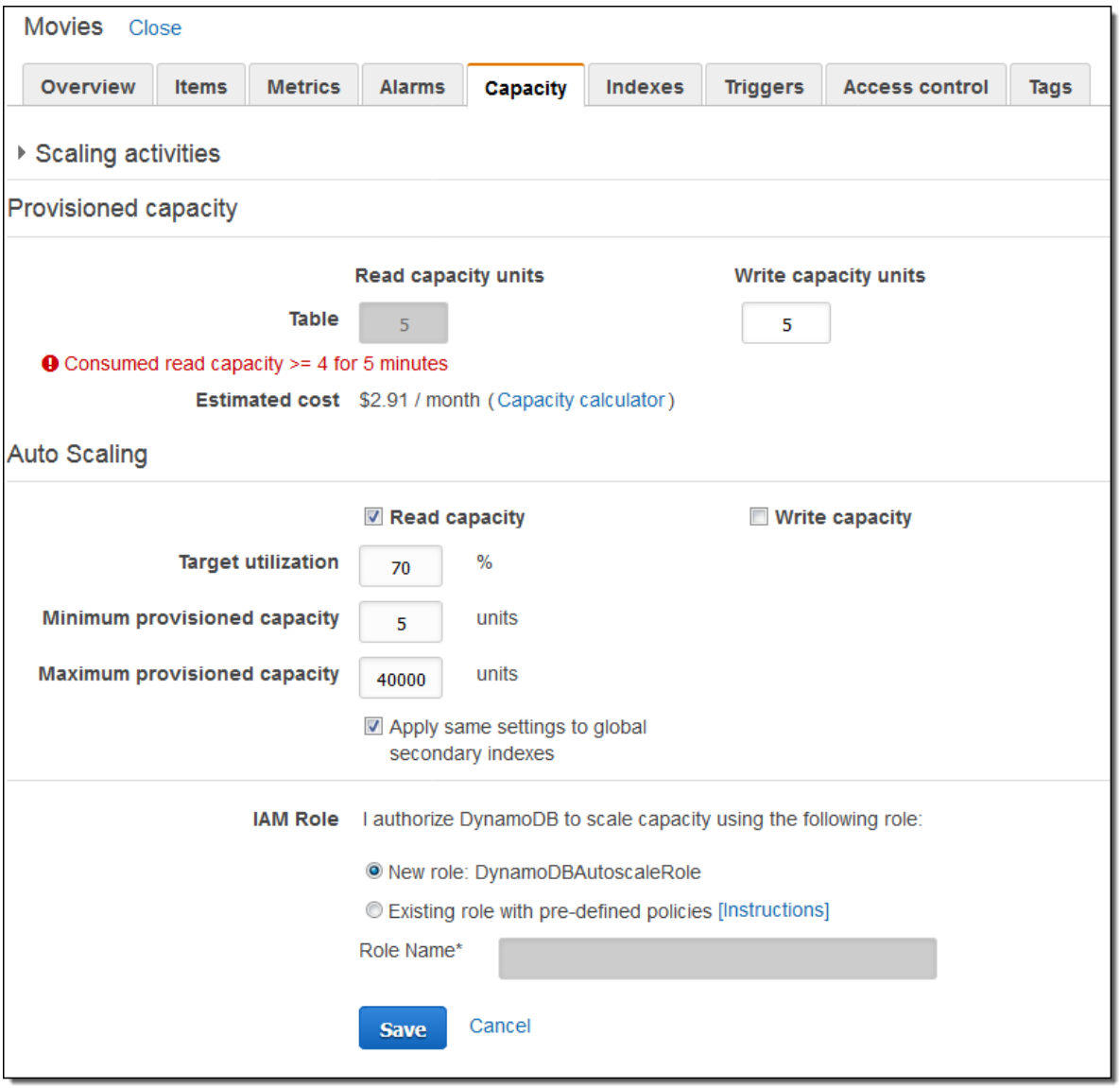
Incorrect

The correct answers are the options that say:

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

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

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



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

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

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

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

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

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

References:

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

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

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

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

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

- Amazon Neptune
- DynamoDB
- SimpleDB
- Amazon Aurora

Correct

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

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

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

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

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

Reference:

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

Check out this Amazon DynamoDB Cheat Sheet:

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

Amazon DynamoDB Overview:

https://youtu.be/3ZOyUNIeorU

5. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Correct

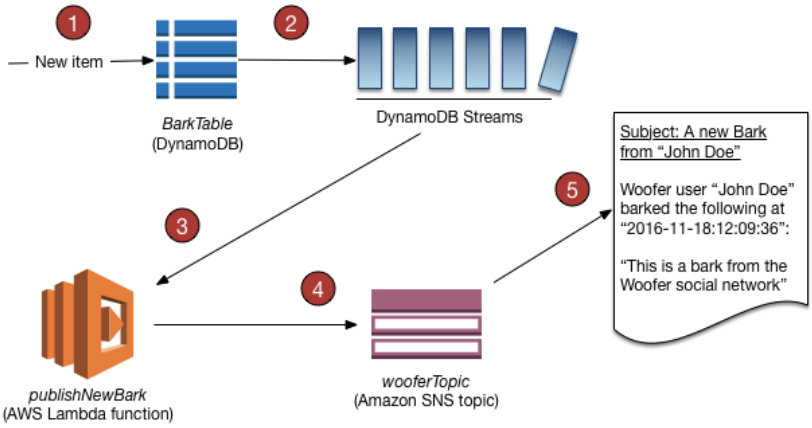
A **DynamoDB stream** is an ordered flow of information about changes to items in an Amazon DynamoDB table. When you enable a stream on a table, DynamoDB captures information about every modification to data items in the table.

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

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

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

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



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

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

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

References:

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

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

Check out this Amazon DynamoDB Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Correct

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

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

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

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

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

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

References:

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

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

Check out this Amazon DynamoDB Cheat Sheet:

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

Amazon DynamoDB Overview:

https://youtu.be/3ZOyUNIeorU

7. QUESTION

Category: CSAA – Design High-Performing Architectures

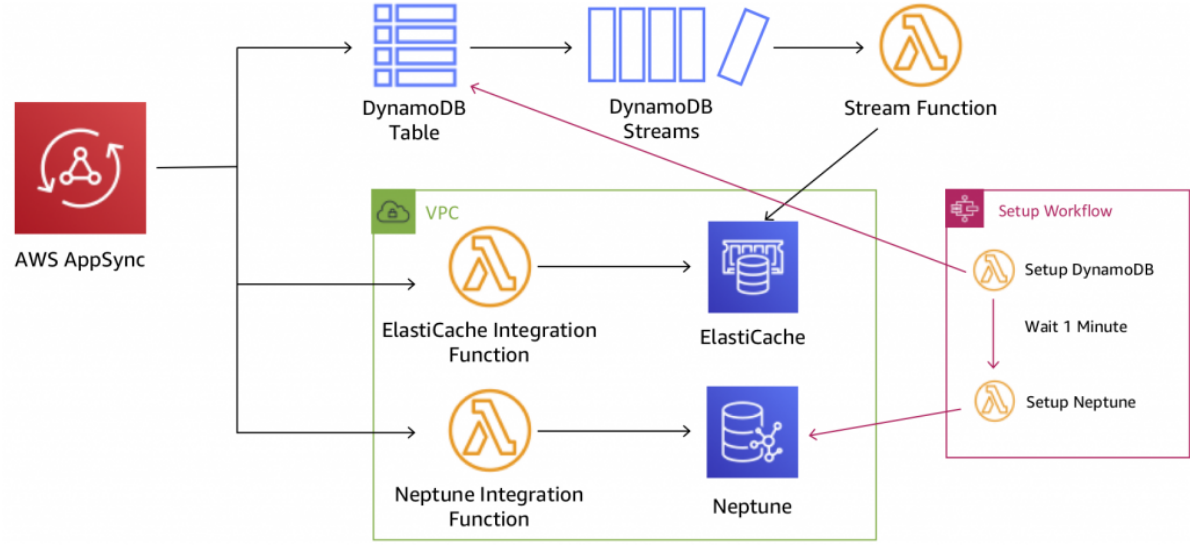
A Solutions Architect needs to deploy a mobile application that can collect votes for a popular singing competition. Millions of users from around the world will submit votes using their mobile phones. These votes must be collected and stored in a highly scalable and highly available data store which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

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

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

Correct

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



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

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

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

References:

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

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

Amazon DynamoDB Overview:

Check out this Amazon DynamoDB Cheat Sheet:

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

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

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

8. QUESTION

Category: CSAA – Design High-Performing Architectures

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

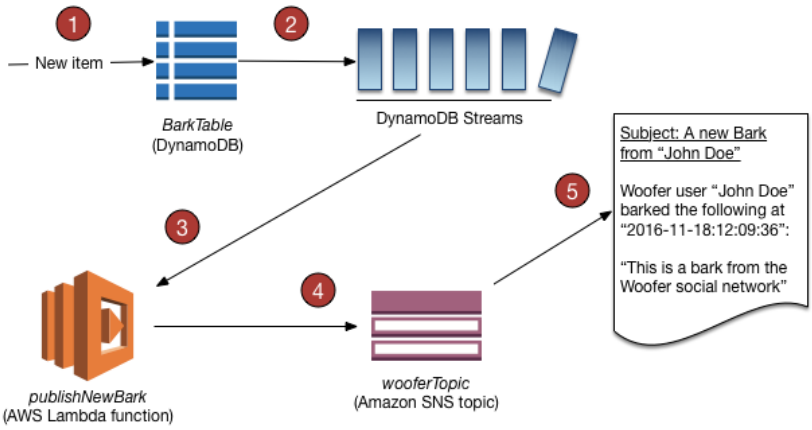
What solution can be easily implemented to alert the Lambda function of new entries while requiring minimal configuration change to your architecture?

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

Correct

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

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



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

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

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

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

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

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

the capability to detect new entries in a DynamoDB table.

References:

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

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

Check out this Amazon DynamoDB cheat sheet:

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

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

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

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

Incorrect

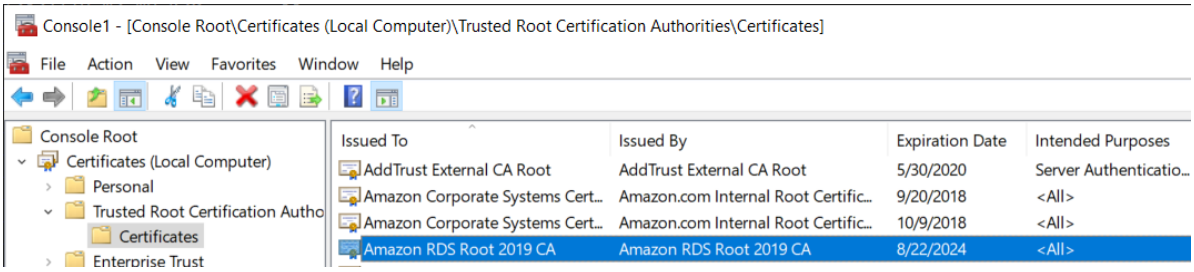
You can use Secure Sockets Layer (SSL) to encrypt connections between your client applications and your Amazon RDS DB instances running Microsoft SQL Server. SSL support is available in all AWS regions for all supported SQL Server editions.

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

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

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

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



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

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

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

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

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

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

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

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

References:

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

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

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

Check out this Amazon RDS Cheat Sheet:

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

2. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Incorrect

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

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

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

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

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

group entries needed to enable the secure channel.

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

Hence, the correct answers are:

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

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

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

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

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

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

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

https://youtu.be/j5da6d2TIPc

3. QUESTION

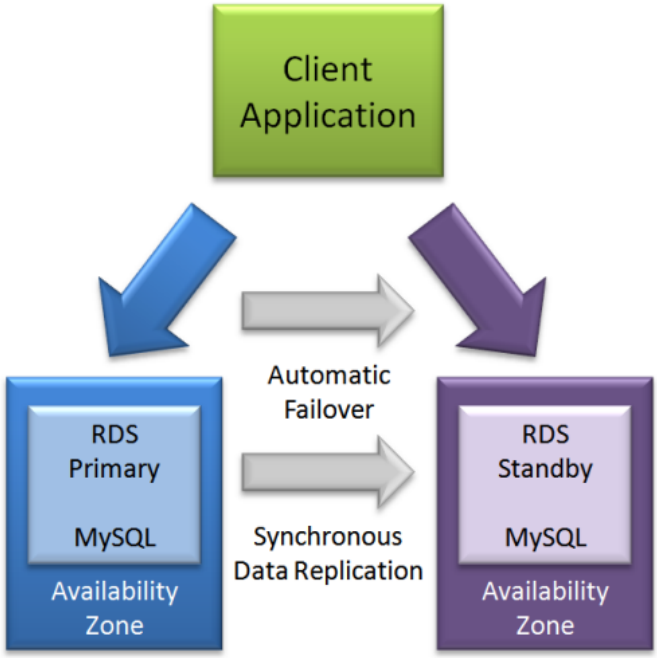
Category: CSAA – Design Resilient Architectures

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

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

Correct

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



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

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

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

References:

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

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

Amazon RDS Overview:*

Check out this Amazon RDS Cheat Sheet:

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

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

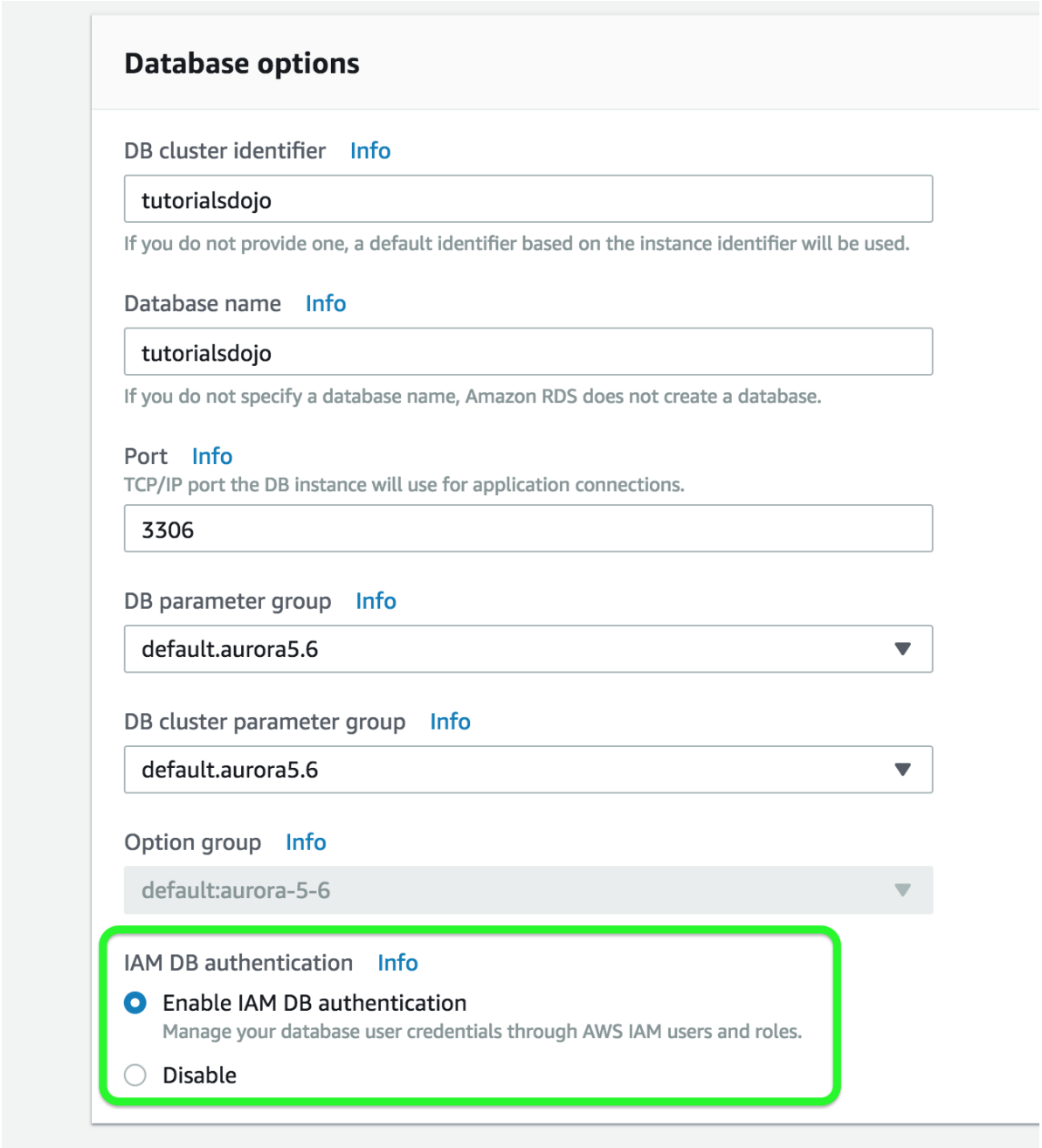
As the Solutions Architect of the company, which of the following should you do to meet the above requirement?

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

Correct

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

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



IAM database authentication provides the following benefits:

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

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

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

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

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

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

Reference:

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

Check out this Amazon RDS cheat sheet:

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

5. QUESTION

Category: CSAA – Design High-Performing Architectures

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

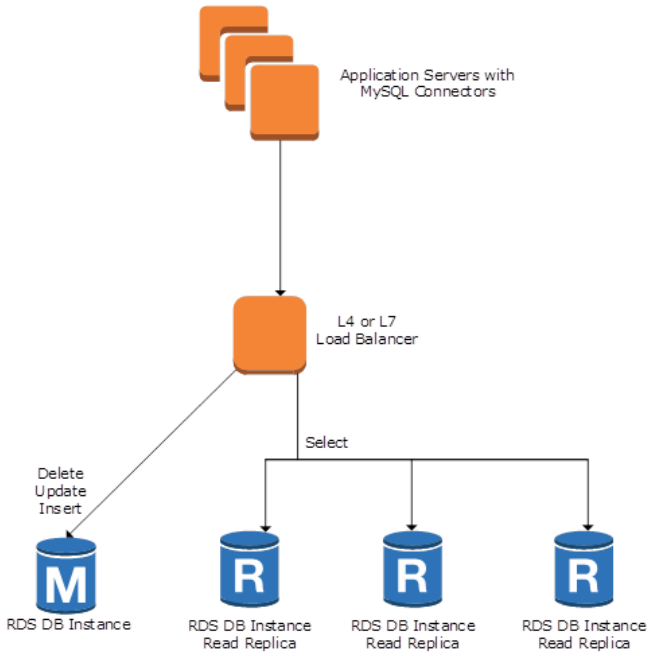
In this scenario, which of the following is the best option to use to increase the read throughput on the MySQL database?

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

Correct

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

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



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

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

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

References:

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

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

Amazon RDS Overview:

Check out this Amazon RDS Cheat Sheet:

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

6. QUESTION

Category: CSAA – Design Resilient Architectures

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

Which of the following performs synchronous data replication in RDS?

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

Correct

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

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

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

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

Reference:

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

Amazon RDS Overview:

Check out this Amazon RDS Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

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

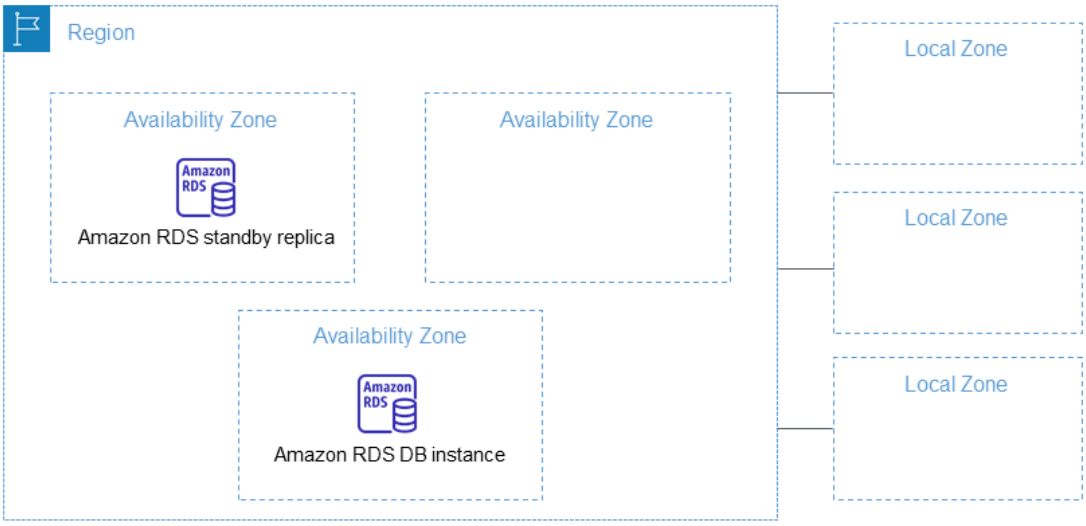
- Storage failure on secondary DB instance
- In the event of Read Replica failure
- Compute unit failure on secondary DB instance

Correct

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

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

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



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

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

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

Hence, the correct answers are:

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

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

– Storage failure on secondary DB instance

– In the event of Read Replica failure

– Compute unit failure on secondary DB instance

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design Resilient Architectures

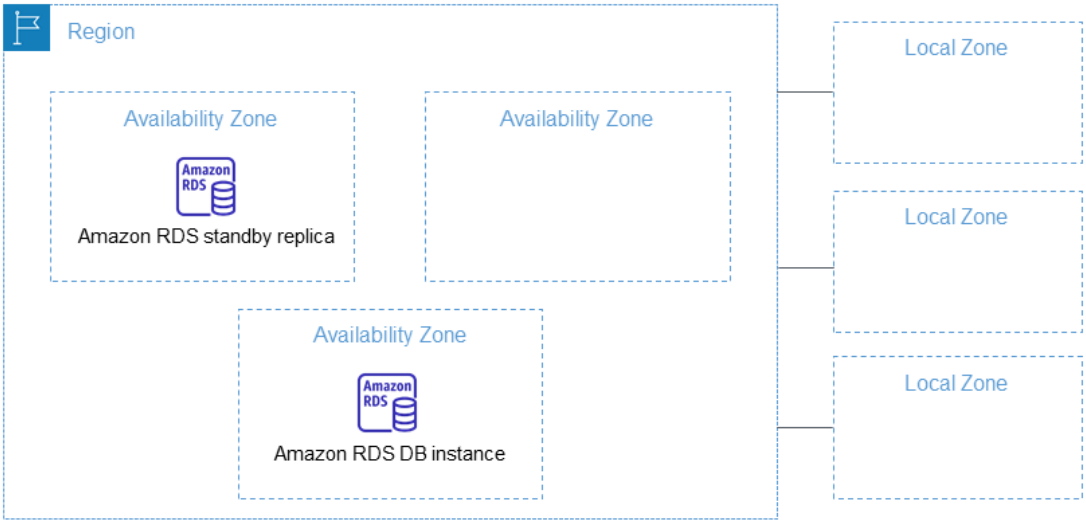
A Forex trading platform, which frequently processes and stores global financial data every minute, is hosted in your on-premises data center and uses an Oracle database. Due to a recent cooling problem in their data center, the company urgently needs to migrate their infrastructure to AWS to improve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

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

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

Correct

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



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

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

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

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

References:

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

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

Check out this Amazon RDS Cheat Sheet:

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

1. QUESTION

Category: CSAA – Design High-Performing Architectures

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

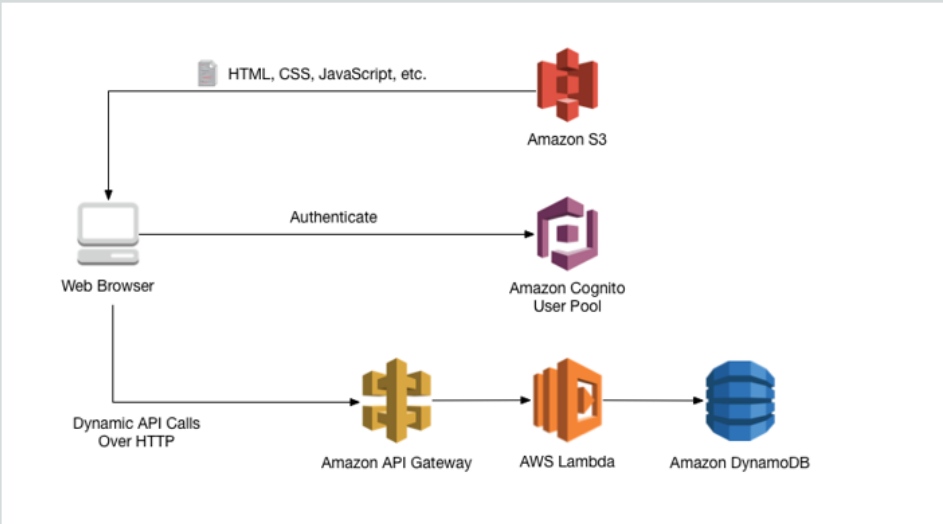
Which of the following should you recommend as the most cost-effective and scalable solution to meet the above requirement?

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

Correct

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

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



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

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

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

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

References:

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

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

Check out this AWS Lambda Cheat Sheet:

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

EC2 Container Service (ECS) vs Lambda:

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

2. QUESTION

Category: CSAA – Design Resilient Architectures

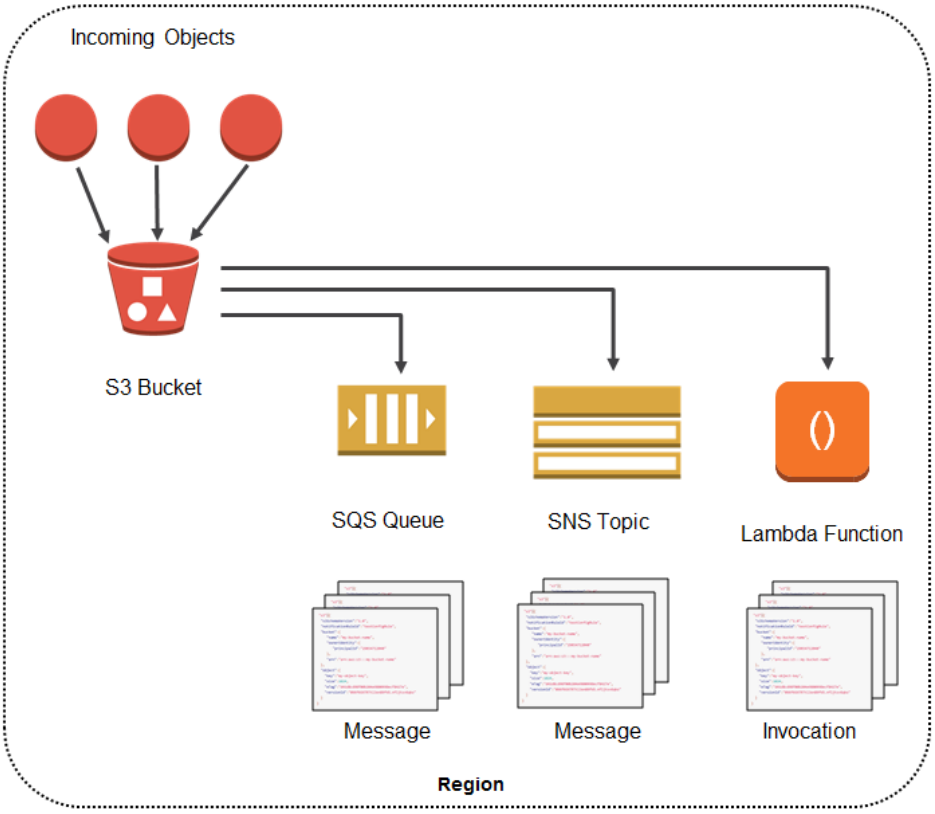
A Data Engineer is working for a litigation firm for their case history application. The engineer needs to keep track of all the cases that the firm has handled. The static assets like .jpg, .png, and .pdf files are stored in S3 for cost efficiency and high durability. As these files are critical to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

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

- Lambda function
- SWF
- SES
- SQS
- Kinesis

Incorrect

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



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

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

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

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

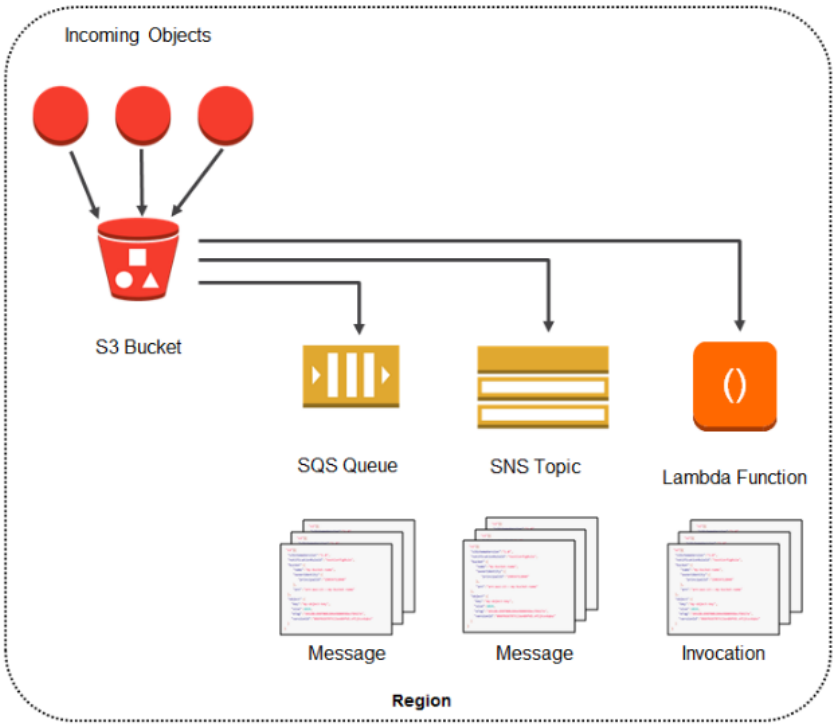
Kinesis is incorrect because this is used to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information, and not used for event notifications. You have to use SNS, SQS or Lambda.

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

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

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

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



Reference:

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

Check out this Amazon S3 Cheat Sheet:

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

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

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

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

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

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

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

Correct

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

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

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

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

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

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

References:

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

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

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

AWS Lambda Overview – Serverless Computing in AWS:

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

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

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

Correct

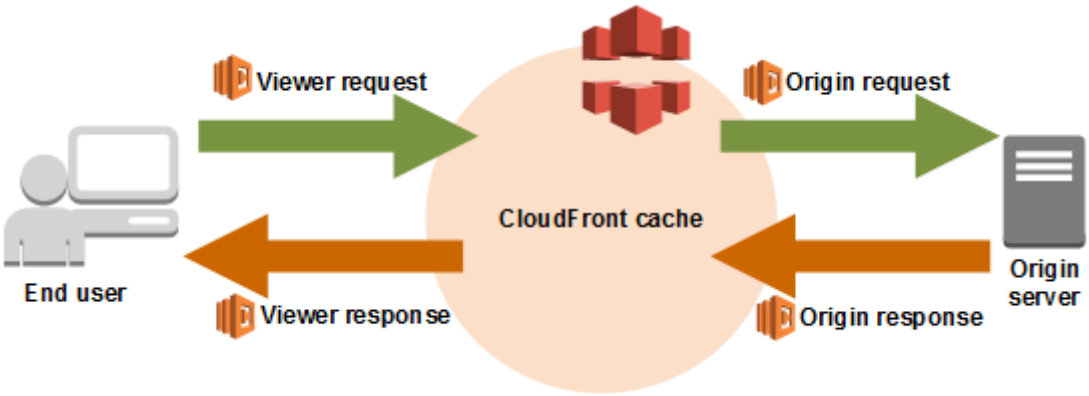
Lambda@Edge lets you run Lambda functions to customize the content that CloudFront delivers, executing the functions in AWS locations closer to the viewer. The functions run in response to CloudFront events, without provisioning or managing servers. You can use Lambda functions to change CloudFront requests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

– After CloudFront receives a request from a viewer (viewer request)

– Before CloudFront forwards the request to the origin (origin request)

– After CloudFront receives the response from the origin (origin response)

– Before CloudFront forwards the response to the viewer (viewer response)



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

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

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

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

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

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

References:

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

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

Check out these Amazon CloudFront and AWS Lambda Cheat Sheets:

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

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

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

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

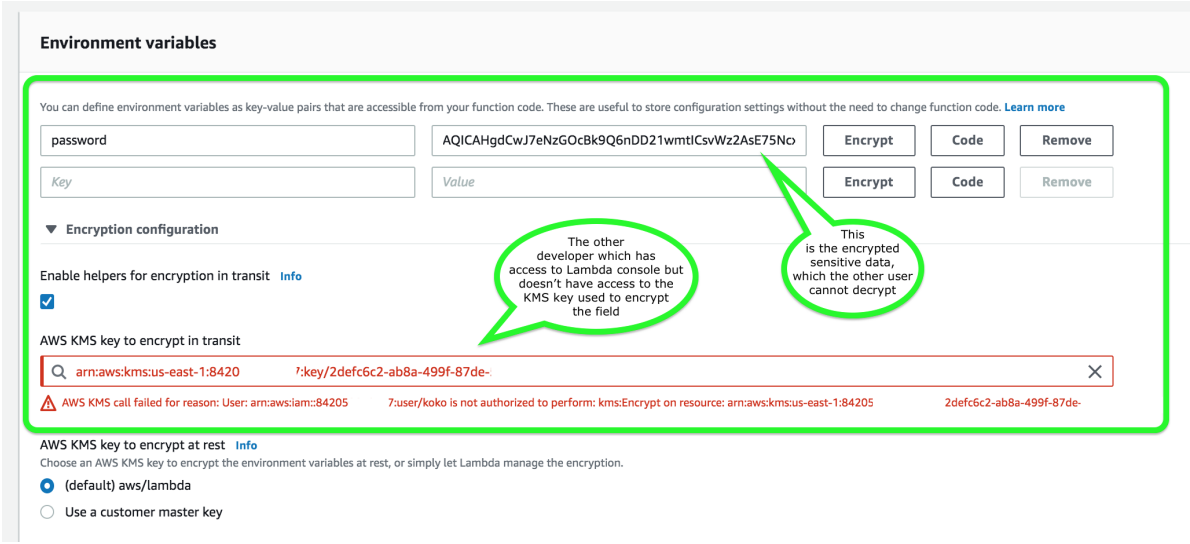
Considering that the Lambda function is storing sensitive database and API credentials, how can this information be secured to prevent other developers in the team, or anyone, from seeing these credentials in plain text? Select the best option that provides maximum security.

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

Incorrect

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

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



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

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

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

References:

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

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

Check out this AWS Lambda Cheat Sheet:

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

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

6. QUESTION

Category: CSAA – Design High-Performing Architectures

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

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

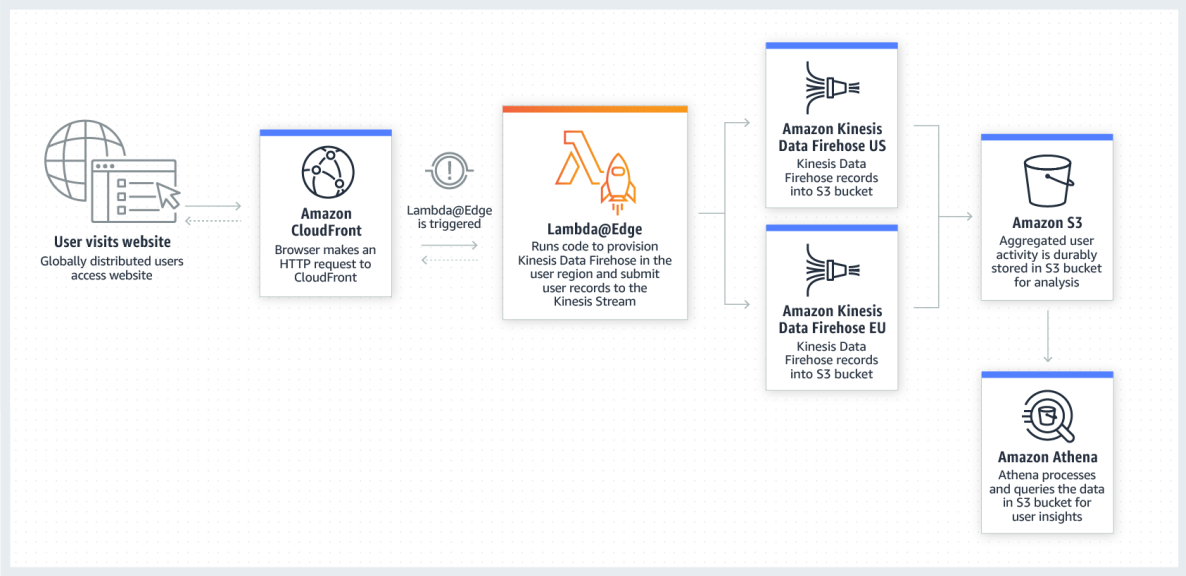
- Integrate CloudFront with Lambda@Edge in order to process the data in close geographical proximity to users and respond to user requests at low latencies. Process real-time streaming data using Kinesis and durably store the results to an Amazon S3 bucket.
- Use a CloudFront web distribution and Route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

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

Correct

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

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



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

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

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

enables you to easily analyze data in Amazon S3 using standard SQL. Kinesis should be used to process the streaming data in real-time.

References:

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

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

7. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Correct

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

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

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

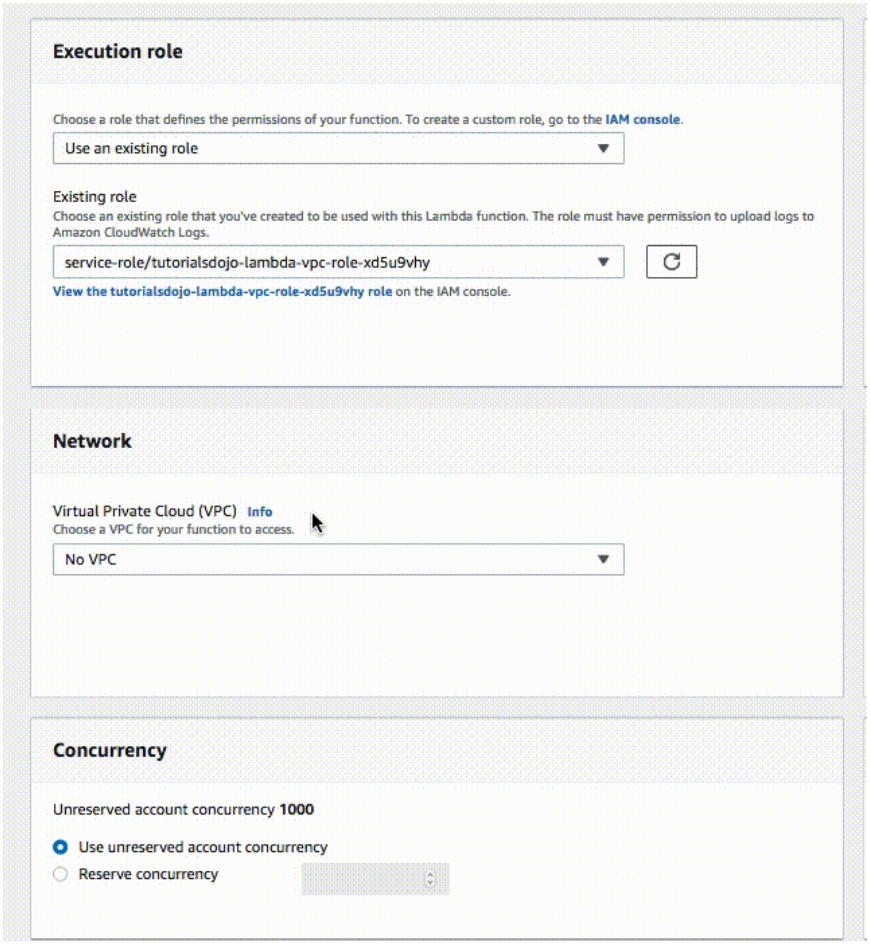
Your Lambda function automatically scales based on the number of events it processes. If your Lambda function accesses a VPC, you must make sure that your VPC has sufficient ENI capacity to support the scale requirements of your Lambda function. It is also recommended that you specify at least one subnet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

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

Hence, the correct answers for this scenario are:

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

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



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

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

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

References:

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

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

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

Check out this AWS Lambda Cheat Sheet:

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

8. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

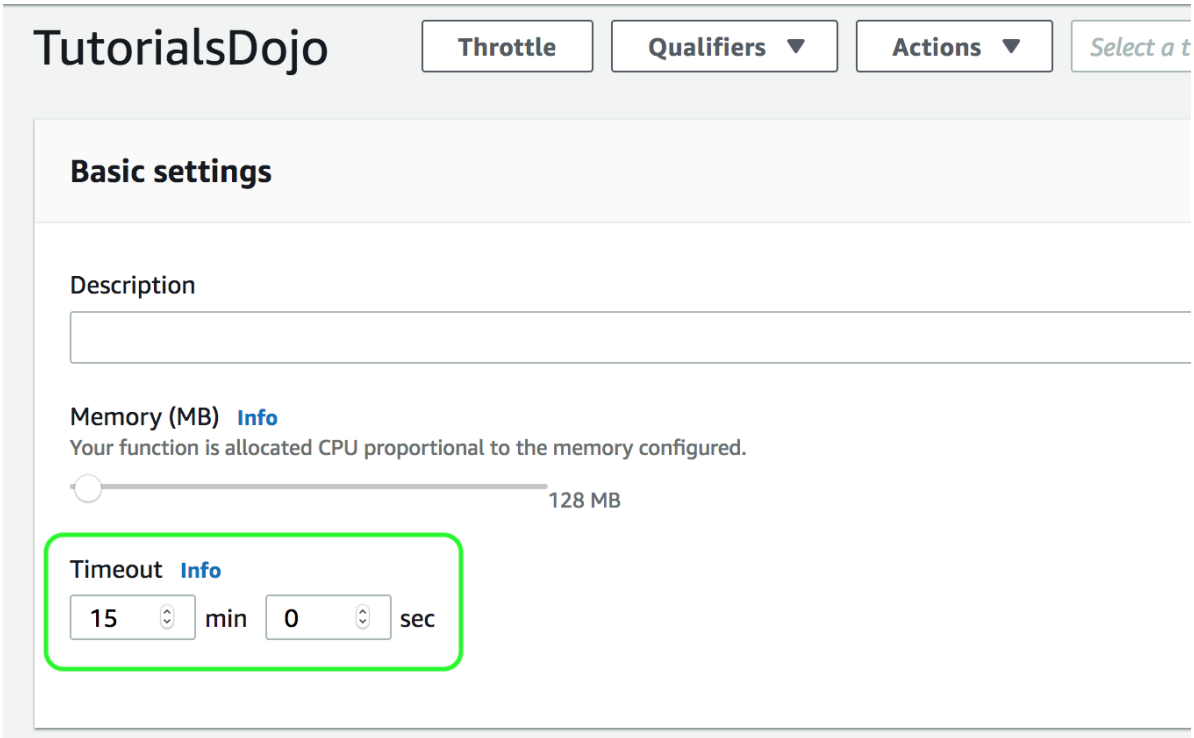
- The failed Lambda Invocations contain a `ServiceException` error which means that the AWS Lambda service encountered an internal error.
- The concurrent execution limit has been reached.
- The Lambda function contains a recursive code and has been running for over 15 minutes.
- The failed Lambda functions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

Correct

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

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

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



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

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

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

The option that says: ***The failed Lambda Invocations contain a `ServiceException` error which means that the AWS Lambda service encountered an internal error**** is incorrect because although this is a valid root cause, it is unlikely to have several **ServiceException** errors throughout the day unless there is an outage or disruption in AWS. Since the scenario says that the Lambda function runs for about 10 to 15 minutes, the maximum execution duration is the most likely cause of the issue and not the AWS Lambda service encountering an internal error.

References:

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

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

AWS Lambda Overview – Serverless Computing in AWS:

Check out this AWS Lambda Cheat Sheet:

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

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application is using CloudFront to distribute their images, videos, and other static contents stored in their S3 bucket to its users around the world. The company has recently introduced a new member-only access to some of its high quality media files. There is a requirement to provide access to multiple private media files only to their paying subscribers without having to change their current URLs.

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

- Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member.
- Create a Signed URL with a custom policy which only allows the members to see the private files.
- Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members.
- Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required `Set-Cookie` headers to the viewer which will unlock the content only to them.

Incorrect

CloudFront signed URLs and signed cookies provide the same basic functionality: they allow you to control who can access your content. If you want to serve private content through CloudFront and you're trying to decide whether to use signed URLs or signed cookies, consider the following:

Use **signed URLs** for the following cases:

– You want to use an RTMP distribution. Signed cookies aren't supported for RTMP distributions.

– You want to restrict access to individual files, for example, an installation download for your application.

– Your users are using a client (for example, a custom HTTP client) that doesn't support cookies.

Use **signed cookies** for the following cases:

– You want to provide access to multiple restricted files, for example, all of the files for a video in HLS format or all of the files in the subscribers' area of a website.

– You don't want to change your current URLs.

Hence, the correct answer for this scenario is the option that says: ***Use Signed Cookies to control who can access the private files in your CloudFront distribution by modifying your application to determine whether a user should have access to your content. For members, send the required*** `Set-Cookie` ***headers to the viewer which will unlock the content only to them.***

The option that says: ***Configure your CloudFront distribution to use Match Viewer as its Origin Protocol Policy which will automatically match the user request. This will allow access to the private content if the request is a paying member and deny it if it is not a member*** is incorrect because a Match Viewer is an Origin Protocol Policy which configures CloudFront to communicate with your origin using HTTP or HTTPS, depending on the protocol of the viewer request. CloudFront caches the object only once even if viewers make requests using both HTTP and HTTPS protocols.

The option that says: ***Create a Signed URL with a custom policy which only allows the members to see the private files*** is incorrect because Signed URLs are primarily used for providing access to individual files, as shown on the above explanation. In addition, the scenario explicitly says that they don't want to change their current URLs which is why implementing Signed Cookies is more suitable than Signed URL.

The option that says: ***Configure your CloudFront distribution to use Field-Level Encryption to protect your private data and only allow access to members**** is incorrect because Field-Level Encryption only allows you to securely upload user-submitted sensitive information to your web servers. It does not provide access to download multiple private files.

Reference:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-choosing-signed-urls-cookies.html

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/private-content-signed-cookies.html

Check out this Amazon CloudFront Cheat Sheet:

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

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A travel photo sharing website is using Amazon S3 to serve high-quality photos to visitors of your website. After a few days, you found out that there are other travel websites linking and using your photos. This resulted in financial losses for your business.

What is the MOST effective method to mitigate this issue?

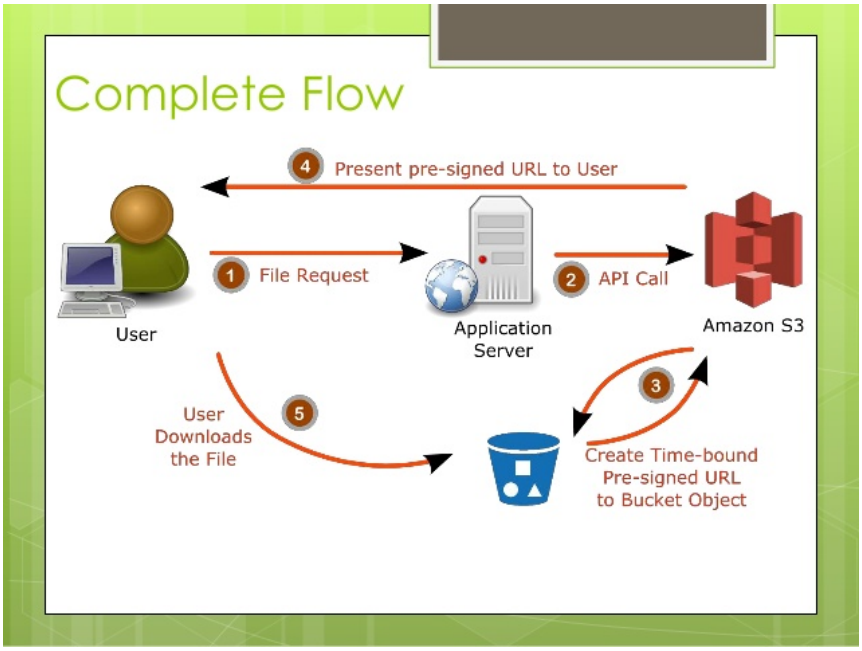
- Store and privately serve the high-quality photos on Amazon WorkDocs instead.
- Configure your S3 bucket to remove public read access and use pre-signed URLs with expiry dates.
- Use CloudFront distributions for your photos.
- Block the IP addresses of the offending websites using NACL.

Incorrect

In Amazon S3, all objects are private by default. Only the object owner has permission to access these objects. However, the object owner can optionally share objects with others by creating a pre-signed URL, using their own security credentials, to grant time-limited permission to download the objects.

When you create a pre-signed URL for your object, you must provide your security credentials, specify a bucket name, an object key, specify the HTTP method (GET to download the object) and expiration date and time. The pre-signed URLs are valid only for the specified duration.

Anyone who receives the pre-signed URL can then access the object. For example, if you have a video in your bucket and both the bucket and the object are private, you can share the video with others by generating a pre-signed URL.



Using CloudFront distributions for your photos is incorrect. CloudFront is a content delivery network service that speeds up delivery of content to your customers.

Blocking the IP addresses of the offending websites using NACL is also incorrect. Blocking IP address using NACLs is not a very efficient method because a quick change in IP address would easily bypass this configuration.

Storing and privately serving the high-quality photos on Amazon WorkDocs instead is incorrect as WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service. It is not a suitable service for storing static content. Amazon WorkDocs is more often used to easily create, edit, and share documents for collaboration and not for serving object data like Amazon S3.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/ShareObjectPreSignedURL.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/ObjectOperations.html

Check out this Amazon CloudFront Cheat Sheet:

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

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

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

Comparison of AWS Services Cheat Sheets:

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

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A website that consists of HTML, CSS, and other client-side Javascript will be hosted on the AWS environment. Several high-resolution images will be displayed on the webpage. The website and the photos should have the optimal loading response times as possible, and should also be able to scale to high request rates.

Which of the following architectures can provide the most cost-effective and fastest loading experience?

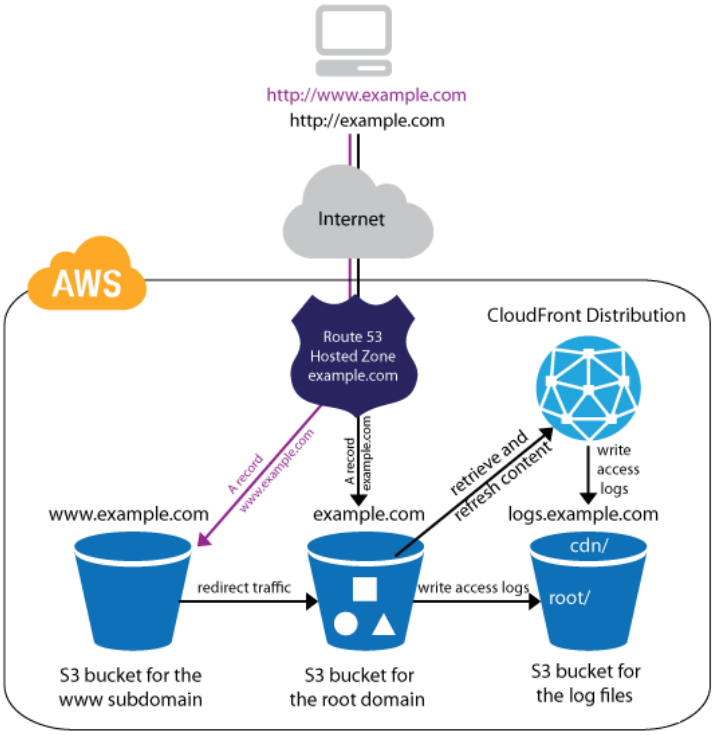
- Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator.

- Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.
- Create a Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.
- Create a Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users.

Correct

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. Additionally, You can use Amazon S3 to host a static website. On a static website, individual webpages include static content. Amazon S3 is **highly scalable and you only pay for what you use**, you can start small and grow your application as you wish, with no compromise on performance or reliability.

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds. CloudFront can be integrated with Amazon S3 for fast delivery of data originating from an S3 bucket to your end-users. By design, delivering data out of CloudFront can be more cost-effective than delivering it from S3 directly to your users.



The scenario given is about storing and hosting images and a static website respectively. Since we are just dealing with static content, we can leverage the web hosting feature of S3. Then we can improve the architecture further by integrating it with CloudFront. This way, users will be able to load both the web pages and images faster than if we are serving them from a standard webserver.

Hence, the correct answer is: ***Upload the HTML, CSS, Javascript, and the images in a single bucket. Then enable website hosting. Create a CloudFront distribution and point the domain on the S3 website endpoint.***

The option that says: ***Create an Nginx web server in an EC2 instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in a S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users*** is incorrect. Creating your own web server just to host a static website in AWS is a costly solution. Web Servers on an EC2 instance is usually used for hosting

dynamic web applications. Since static websites contain web pages with fixed content, we should use S3 website hosting instead.

The option that says: ***Launch an Auto Scaling Group using an AMI that has a pre-configured Apache web server, then configure the scaling policy accordingly. Store the images in an Elastic Block Store. Then, point your instance's endpoint to AWS Global Accelerator**** is incorrect. This is how we serve static websites in the old days. Now, with the help of S3 website hosting, we can host our static contents from a durable, high-availability, and highly scalable environment without managing any servers. Hosting static websites in S3 is cheaper than hosting it in an EC2 instance. In addition, Using ASG for scaling instances that host a static website is an over-engineered solution that carries unnecessary costs. S3 automatically scales to high requests and you only pay for what you use.

The option that says: ***Create an Nginx web server in an Amazon LightSail instance to host the HTML, CSS, and Javascript files then enable caching. Upload the images in an S3 bucket. Use CloudFront as a CDN to deliver the images closer to your end-users**** is incorrect because although LightSail is cheaper than EC2, creating your own LightSail web server for hosting static websites is still a relatively expensive solution when compared to hosting it on S3. In addition, S3 automatically scales to high request rates.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html

https://aws.amazon.com/blogs/networking-and-content-delivery/amazon-s3-amazon-cloudfront-a-match-made-in-the-cloud/

Check out these Amazon S3 and CloudFront Cheat Sheets:

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

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

4. QUESTION

Category: CSAA – Design High-Performing Architectures

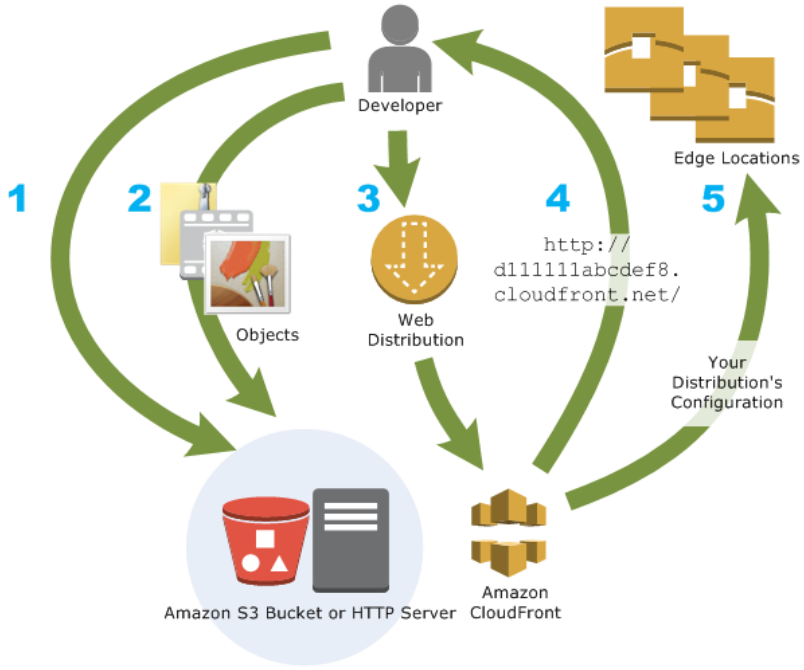
A company has a global news website hosted in a fleet of EC2 Instances. Lately, the load on the website has increased which resulted in slower response time for the site visitors. This issue impacts the revenue of the company as some readers tend to leave the site if it does not load after 10 seconds.

Which of the below services in AWS can be used to solve this problem? (Select TWO.)

- Deploy the website to all regions in different VPCs for faster processing.
- Use Amazon ElastiCache for the website's in-memory data store or cache.
- For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions.
- Use Amazon CloudFront with website as the custom origin.

Correct

The global news website has a problem with latency considering that there are a lot of readers of the site from all parts of the globe. In this scenario, you can use a content delivery network (CDN) which is a geographically distributed group of servers that work together to provide fast delivery of Internet content. And since this is a news website, most of its data are read-only, which can be cached to improve the read throughput and avoid repetitive requests from the server.



In AWS, Amazon CloudFront is the global content delivery network (CDN) service that you can use and for web caching, Amazon ElastiCache is the suitable service.

Hence, the correct answers are:

– Use Amazon CloudFront with website as the custom origin.

– Use Amazon ElastiCache for the website's in-memory data store or cache.

The option that says: ***For better read throughput, use AWS Storage Gateway to distribute the content across multiple regions*** is incorrect as AWS Storage Gateway is used for storage.

Deploying the website to all regions in different VPCs for faster processing is incorrect as this would be costly and totally unnecessary considering that you can use Amazon CloudFront and ElastiCache to improve the performance of the website.

References:

https://aws.amazon.com/elasticache/

http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html

Check out this Amazon CloudFront Cheat Sheet:

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

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A web application, which is used by your clients around the world, is hosted in an Auto Scaling group of EC2 instances behind a Classic Load Balancer. You need to secure your application by allowing multiple domains to serve SSL traffic over the same IP address.

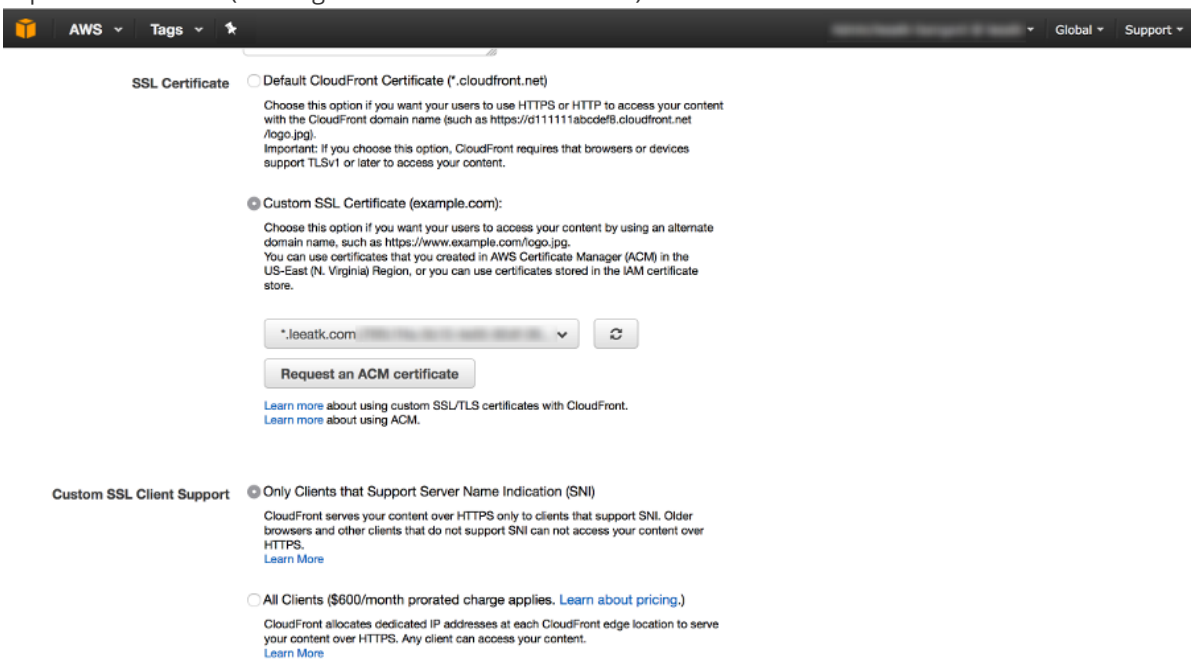
Which of the following should you do to meet the above requirement?

- Generate an SSL certificate with AWS Certificate Manager and create a CloudFront web distribution. Associate the certificate with your web distribution and enable the support for Server Name Indication (SNI).
- It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS.
- Use Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic.
- Use an Elastic IP and upload multiple 3rd party certificates in your Classic Load Balancer using the AWS Certificate Manager.

Incorrect

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

Amazon CloudFront delivers your content from each edge location and offers the same security as the Dedicated IP Custom SSL feature. SNI Custom SSL works with most modern browsers, including Chrome version 6 and later (running on Windows XP and later or OS X 10.5.7 and later), Safari version 3 and later (running on Windows Vista and later or Mac OS X 10.5.6. and later), Firefox 2.0 and later, and Internet Explorer 7 and later (running on Windows Vista and later).



Some users may not be able to access your content because some older browsers do not support SNI and will not be able to establish a connection with CloudFront to load the HTTPS version of your content. If you need to support non-SNI compliant browsers for HTTPS content, it is recommended to use the Dedicated IP Custom SSL feature.

Using Server Name Indication (SNI) on your Classic Load Balancer by adding multiple SSL certificates to allow multiple domains to serve SSL traffic is incorrect because a Classic Load Balancer does not support Server Name Indication (SNI). You have to use an Application Load Balancer instead or a CloudFront web distribution to allow the SNI feature.

Using an Elastic IP and uploading multiple 3rd party certificates in your Classic Load Balancer using the AWS Certificate Manager is incorrect because just like in the above, a Classic Load Balancer does not support Server Name Indication (SNI) and the use of an Elastic IP is not a suitable solution to allow multiple domains to serve SSL traffic. You have to use Server Name Indication (SNI).

The option that says: ***It is not possible to allow multiple domains to serve SSL traffic over the same IP address in AWS*** is incorrect because AWS does support the use of Server Name Indication (SNI).

References:

https://aws.amazon.com/about-aws/whats-new/2014/03/05/amazon-cloudront-announces-sni-custom-ssl/

https://aws.amazon.com/blogs/security/how-to-help-achieve-mobile-app-transport-security-compliance-by-using-amazon-cloudfront-and-aws-certificate-manager/

Check out this Amazon CloudFront Cheat Sheet:

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

SNI Custom SSL vs Dedicated IP Custom SSL:

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

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A global news network created a CloudFront distribution for their web application. However, you noticed that the application's origin server is being hit for each request instead of the AWS Edge locations, which serve the cached objects. The issue occurs even for the commonly requested objects.

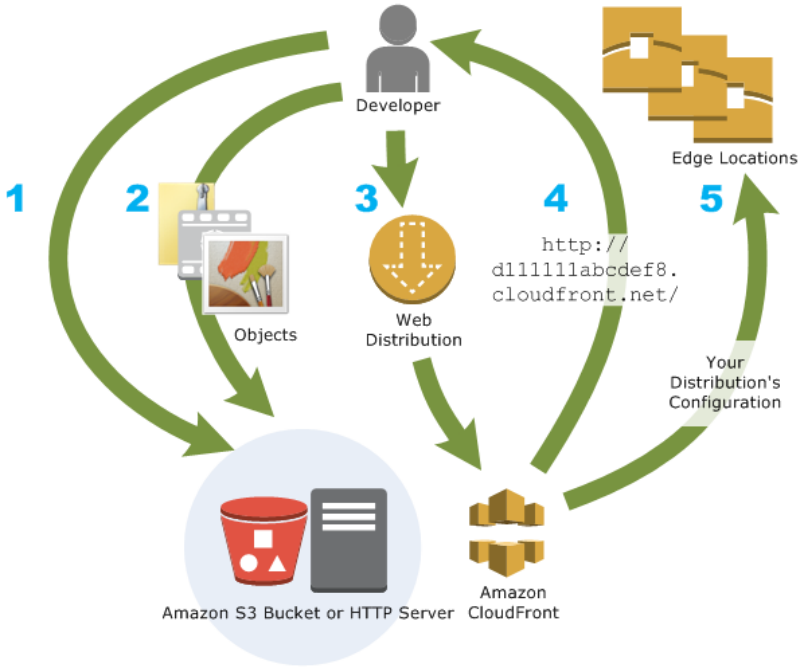
What could be a possible cause of this issue?

- The file sizes of the cached objects are too large for CloudFront to handle.
- You did not add an SSL certificate.
- An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server.
- The Cache-Control max-age directive is set to zero.

Correct

You can control how long your objects stay in a CloudFront cache before CloudFront forwards another request to your origin. Reducing the duration allows you to serve dynamic content. Increasing the duration means your users get better performance because your objects are more likely to be served directly from the edge cache. A longer duration also reduces the load on your origin.

Typically, CloudFront serves an object from an edge location until the cache duration that you specified passes — that is, until the object expires. After it expires, the next time the edge location gets a user request for the object, CloudFront forwards the request to the origin server to verify that the cache contains the latest version of the object.



The `Cache-Control` and `Expires` headers control how long objects stay in the cache. The `Cache-Control max-age` directive lets you specify how long (in seconds) you want an object to remain in the cache before CloudFront gets the object again from the origin server. The minimum expiration time CloudFront supports is 0 seconds for web distributions and 3600 seconds for RTMP distributions.

In this scenario, the main culprit is that the Cache-Control max-age directive is set to a low value, which is why the request is always directed to your origin server.

Hence, the correct answer is: ***The Cache-Control max-age directive is set to zero.***

The option that says: ***An object is only cached by CloudFront once a successful request has been made hence, the objects were not requested before, which is why the request is still directed to the origin server**** is incorrect because the issue also occurs even for the commonly requested objects. This means that these objects were successfully requested before but due to a zero Cache-Control max-age directive value, it causes this issue in CloudFront.

The options that say: ***The file sizes of the cached objects are too large for CloudFront to handle**** and ***You did not add an SSL certificate**** are incorrect because they are not related to the issue in caching.

Reference:

http://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Expiration.html

Check out this Amazon CloudFront Cheat Sheet:

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

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

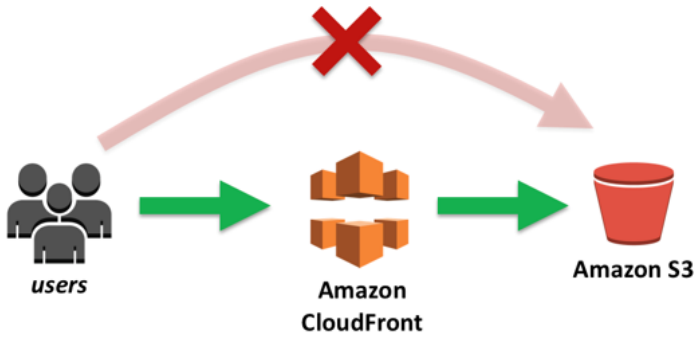
A Solutions Architect is working for a large global media company with multiple office locations all around the world. The Architect is instructed to build a system to distribute training videos to all employees.

Using CloudFront, what method would be used to serve content that is stored in S3, but not publicly accessible from S3 directly?

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

Correct

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



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

– Grant the CloudFront origin access identity the applicable permissions on the bucket.

– Deny access to anyone that you don't want to have access using Amazon S3 URLs.

Reference:

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

Check out this Amazon CloudFront Cheat Sheet:

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

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

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

Comparison of AWS Services Cheat Sheets:

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

1. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A Solutions Architect is working for a financial company. The manager wants to have the ability to automatically transfer obsolete data from their S3 bucket to a low-cost storage system in AWS.

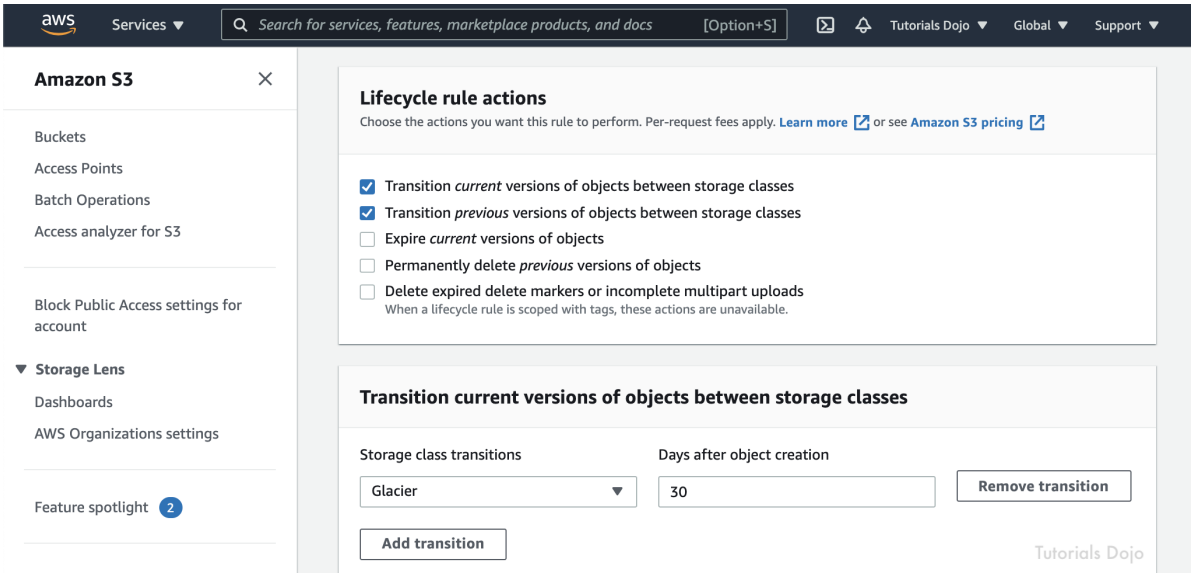
What is the best solution that the Architect can provide to them?

- Use Amazon SQS.
- Use CloudEndure Migration.
- Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier.
- Use Lifecycle Policies in S3 to move obsolete data to Glacier.

Correct

In this scenario, you can use lifecycle policies in S3 to automatically move obsolete data to Glacier.

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



These actions can be classified as follows:

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

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

The option that says: ***Use an EC2 instance and a scheduled job to transfer the obsolete data from their S3 location to Amazon S3 Glacier*** is incorrect because you don't need to create a scheduled job in EC2 as you can simply use the lifecycle policy in S3.

The option that says: ***Use Amazon SQS*** is incorrect as SQS is not a storage service. Amazon SQS is primarily used to decouple your applications by queueing the incoming requests of your application.

The option that says: ***Use CloudEndure Migration*** is incorrect because this service is just a highly automated lift-and-shift (rehost) solution that simplifies, expedites, and reduces the cost of migrating applications to AWS. You cannot use this to automatically transition your S3 objects to a cheaper storage class.

References:

http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

https://aws.amazon.com/blogs/aws/archive-s3-to-glacier/

Check out this Amazon S3 Cheat Sheet:

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

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

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

2. QUESTION

Category: CSAA – Design Resilient Architectures

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

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

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

Correct

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

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

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

– Change the versioning state of your bucket

– Permanently delete an object version

MFA Delete requires two forms of authentication together:

– Your security credentials

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

Providing access to S3 data strictly through pre-signed URL only is incorrect since a pre-signed URL gives access to the object identified in the URL. Pre-signed URLs are useful when customers perform an object upload to your S3 bucket, but does not help in preventing accidental deletes.

Disallowing S3 Delete using an IAM bucket policy is incorrect since you still want users to be able to delete objects in the bucket, and you just want to prevent accidental deletions. Disallowing S3 Delete using an IAM bucket policy will restrict all delete operations to your bucket.

Enabling Amazon S3 Intelligent-Tiering is incorrect since S3 intelligent tiering does not help in this situation.

Reference:

https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

A start-up company that offers an intuitive financial data analytics service has consulted you about their AWS architecture. They have a fleet of Amazon EC2 worker instances that process financial data and then outputs reports which are used by their clients. You must store the generated report files in a durable storage. The number of files to be stored can grow over time as the start-up company is expanding rapidly overseas and hence, they also need a way to distribute the reports faster to clients located across the globe.

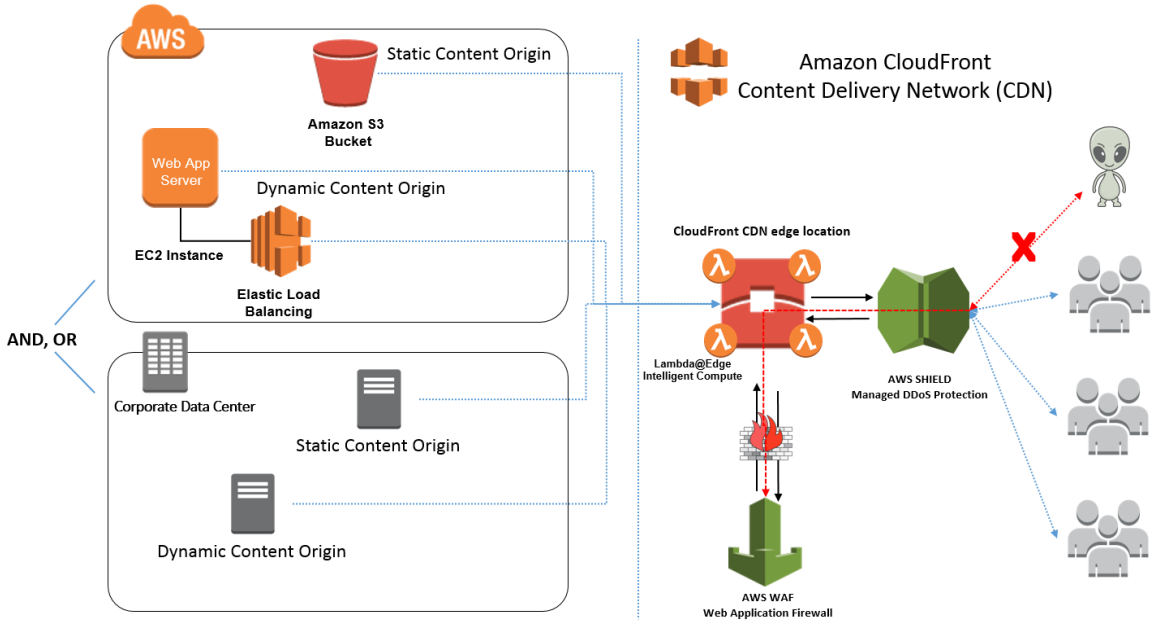
Which of the following is a cost-efficient and scalable storage option that you should use for this scenario?

- Use multiple EC2 instance stores for data storage and ElastiCache as the CDN.
- Use Amazon S3 Glacier as the data storage and ElastiCache as the CDN.
- Use Amazon Redshift as the data storage and CloudFront as the CDN.
- Use Amazon S3 as the data storage and CloudFront as the CDN.

Correct

A Content Delivery Network (CDN) is a critical component of nearly any modern web application. It used to be that CDN merely improved the delivery of content by replicating commonly requested files (static content) across a globally distributed set of caching servers. However, CDNs have become much more useful over time.

For caching, a CDN will reduce the load on an application origin and improve the experience of the requestor by delivering a local copy of the content from a nearby cache edge, or Point of Presence (PoP). The application origin is off the hook for opening the connection and delivering the content directly as the CDN takes care of the heavy lifting. The end result is that the application origins don't need to scale to meet demands for static content.



Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment. CloudFront is integrated with AWS – both physical locations that are directly connected to the AWS global infrastructure, as well as other AWS services.

Amazon S3 offers a highly durable, scalable, and secure destination for backing up and archiving your critical data. This is the correct option as the start-up company is looking for a durable storage to store the audio and text files. In addition, ElastiCache is only used for caching and not specifically as a Global Content Delivery Network (CDN).

Using Amazon Redshift as the data storage and CloudFront as the CDN is incorrect as Amazon Redshift is usually used as a Data Warehouse.

Using Amazon S3 Glacier as the data storage and ElastiCache as the CDN is incorrect as Amazon S3 Glacier is usually used for data archives.

Using multiple EC2 instance stores for data storage and ElastiCache as the CDN is incorrect as data stored in an instance store is not durable.

References:

https://aws.amazon.com/s3/

https://aws.amazon.com/caching/cdn/

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

4. QUESTION

Category: CSAA – Design High-Performing Architectures

A Solutions Architect created a new Standard-class S3 bucket to store financial reports that are not frequently accessed but should immediately be available when an auditor requests them. To save costs, the Architect changed the storage class of the S3 bucket from Standard to Infrequent Access storage class.

In Amazon S3 Standard – Infrequent Access storage class, which of the following statements are true? (Select TWO.)

- It provides high latency and low throughput performance.
- It automatically moves data to the most cost-effective access tier without any operational overhead.
- It is designed for data that requires rapid access when needed.
- It is designed for data that is accessed less frequently.
- Ideal to use for data archiving.

Correct

Amazon S3 Standard – Infrequent Access (Standard – IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Standard – IA offers the high durability, throughput, and low latency of Amazon S3 Standard, with a low per GB storage price and per GB retrieval fee.

	S3 Standard	S3 Standard-Infrequent Access (IA)	S3 One Zone-Infrequent Access (IA)	S3 Intelligent Tiering
Features	General-purpose storage of frequently accessed data	For long-lived, rapid but less frequently accessed data; data is stored redundantly in multiple AZs	For long-lived, rapid but less frequently accessed data; data is stored redundantly in only one AZ of your choice	For long-lived data that have unpredictable access patterns
Durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Availability	99.99%	99.9%	99.5%	99.9%
Availability SLA	99.9%	99%	99%	99%
Number of Availability Zones	At least 3	At least 3	Only 1	At least 3
Minimum capacity charge per object	N/A	128KB	128KB	N/A
Minimum storage duration charge	N/A	30 days	30 days	30 days
Inserting data	Directly PUT into S3 Standard	Directly PUT into S3 Standard-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 Standard-IA storage class.	Directly PUT into S3 One Zone-IA or set Lifecycle policies to transition objects from the S3 Standard to the S3 One Zone-IA storage class.	Directly PUT into S3 Intelligent-Tiering or set Lifecycle policies to transition objects from the S3 Standard to the S3 Intelligent-Tiering storage class.
Retrieval fee	N/A	per GB retrieved	per GB retrieved	N/A
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds
Storage transition	S3 Standard to all other S3 storage types including Glacier	S3 Standard-IA to S3 One Zone-IA or S3 Glacier	S3 One Zone-IA to S3 Glacier	S3 Intelligent to S3 One Zone-IA or S3 Glacier
Use Cases	Cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.	Ideally suited for long-term file storage, older sync and share storage, and other aging data.	For infrequently-accessed storage, like backup copies, disaster recovery copies, or other easily recreatable data.	Data with unknown or changing access patterns, optimize storage costs automatically, and unpredictable workloads

This combination of low cost and high performance make Standard – IA ideal for long-term storage, backups, and as a data store for disaster recovery. The Standard – IA storage class is set at the object level and can exist in the same bucket as Standard, allowing you to use lifecycle policies to automatically transition objects between storage classes without any application changes.

Key Features:

– Same low latency and high throughput performance of Standard

– Designed for durability of 99.999999999% of objects

– Designed for 99.9% availability over a given year

– Backed with the Amazon S3 Service Level Agreement for availability

– Supports SSL encryption of data in transit and at rest

– Lifecycle management for automatic migration of objects

Hence, the correct answers are:

– ***It is designed for data that is accessed less frequently.****

– ***It is designed for data that requires rapid access when needed.****

The option that says: ***It automatically moves data to the most cost-effective access tier without any operational overhead**** is incorrect as it actually refers to Amazon S3 – Intelligent Tiering, which is the only cloud storage class that delivers automatic cost savings by moving objects between different access tiers when access patterns change.

The option that says: ***It provides high latency and low throughput performance**** is incorrect as it should be "low latency" and "high throughput" instead. S3 automatically scales performance to meet user demands.

The option that says: ***Ideal to use for data archiving**** is incorrect because this statement refers to Amazon S3 Glacier. Glacier is a secure, durable, and extremely low-cost cloud storage service for data archiving and long-term backup.

References:

https://aws.amazon.com/s3/storage-classes/

https://aws.amazon.com/s3/faqs

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

5. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An online medical system hosted in AWS stores sensitive Personally Identifiable Information (PII) of the users in an Amazon S3 bucket. Both the master keys and the unencrypted data should never be sent to AWS to comply with the strict compliance and regulatory requirements of the company.

Which S3 encryption technique should the Architect use?

- Use S3 client-side encryption with a client-side master key.
- Use S3 client-side encryption with a KMS-managed customer master key.
- Use S3 server-side encryption with a KMS managed key.
- Use S3 server-side encryption with customer provided key.

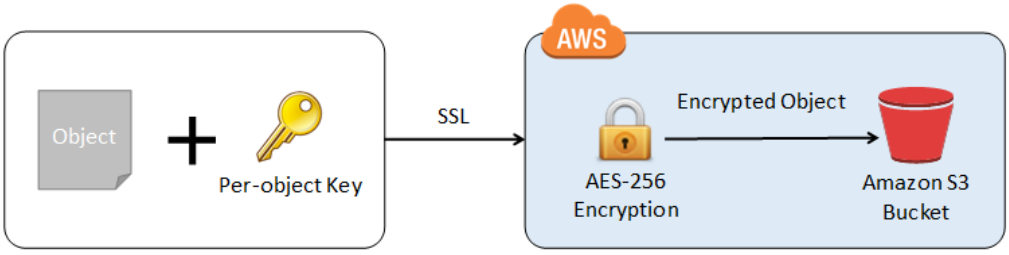
Correct

Client-side encryption is the act of encrypting data before sending it to Amazon S3. To enable client-side encryption, you have the following options:

– Use an AWS KMS-managed customer master key.

– Use a client-side master key.

When using an AWS KMS-managed customer master key to enable client-side data encryption, you provide an AWS KMS customer master key ID (CMK ID) to AWS. On the other hand, when you use client-side master key for client-side data encryption, **your client-side master keys and your unencrypted data are never sent to AWS**. It's important that you safely manage your encryption keys because if you lose them, you can't decrypt your data.



This is how client-side encryption using client-side master key works:

When uploading an object – You provide a client-side master key to the Amazon S3 encryption client. The client uses the master key only to encrypt the data encryption key that it generates randomly. The process works like this:

\1. The Amazon S3 encryption client generates a one-time-use symmetric key (also known as a data encryption key or data key) locally. It uses the data key to encrypt the data of a single Amazon S3 object. The client generates a separate data key for each object.

\2. The client encrypts the data encryption key using the master key that you provide. The client uploads the encrypted data key and its material description as part of the object metadata. The client uses the material description to determine which client-side master key to use for decryption.

\3. The client uploads the encrypted data to Amazon S3 and saves the encrypted data key as object metadata (`x-amz-meta-x-amz-key`) in Amazon S3.

When downloading an object – The client downloads the encrypted object from Amazon S3. Using the material description from the object's metadata, the client determines which master key to use to decrypt the data key. The client uses that master key to decrypt the data key and then uses the data key to decrypt the object.

Hence, the correct answer is to ***use S3 client-side encryption with a client-side master key***.

Using S3 client-side encryption with a KMS-managed customer master key is incorrect because in client-side encryption with a KMS-managed customer master key, you provide an AWS KMS customer master key ID (CMK ID) to AWS. The scenario clearly indicates that both the master keys and the unencrypted data should never be sent to AWS.

Using S3 server-side encryption with a KMS managed key is incorrect because the scenario mentioned that the unencrypted data should never be sent to AWS, which means that you have to use client-side encryption in order to encrypt the data first before sending to AWS. In this way, you can ensure that there is no unencrypted data being uploaded to AWS. In addition, the master key used by Server-Side Encryption with AWS KMS–Managed Keys (SSE-KMS) is uploaded and managed by AWS, which directly violates the requirement of not uploading the master key.

Using S3 server-side encryption with customer provided key is incorrect because just as mentioned above, you have to use client-side encryption in this scenario instead of server-side encryption. For the S3 server-side encryption with customer-provided key (SSE-C), you actually provide the encryption key as part of your request to upload the object to S3. Using this key, Amazon S3 manages both the encryption (as it writes to disks) and decryption (when you access your objects).

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

6. QUESTION

Category: CSAA – Design High-Performing Architectures

A company collects atmospheric data such as temperature, air pressure, and humidity from different countries. Each site location is equipped with various weather instruments and a high-speed Internet connection. The average collected data in each location is around 500 GB and will be analyzed by a weather forecasting application hosted in Northern Virginia. As the Solutions Architect, you need to aggregate all the data in the fastest way.

Which of the following options can satisfy the given requirement?

- Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.
- Use AWS Snowball Edge to transfer large amounts of data.
- Set up a Site-to-Site VPN connection.
- Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket.

Incorrect

Amazon S3 is object storage built to store and retrieve any amount of data from anywhere on the Internet. It's a simple storage service that offers industry-leading durability, availability, performance, security, and virtually unlimited scalability at very low costs. Amazon S3 is also designed to be highly flexible. Store any type and amount of data that you want; read the same piece of data a million times or only for emergency disaster recovery; build a simple FTP application or a sophisticated web application.

Amazon S3 Transfer Acceleration

Speed Comparison

Upload speed comparison in the selected region
(Based on the location of bucket: jbarr-public)

N. Virginia
(US-EAST-1) 539% faster

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed

Upload complete

This speed checker uses multipart uploads to transfer a file from your browser to various Amazon S3 regions with and without Amazon S3 Transfer Acceleration. It compares the speed results and shows the percentage difference for every region.

Note: In general, the farther away you are from an Amazon S3 region, the higher the speed improvement you can expect from using Amazon S3 Transfer Acceleration. If you see similar speed results with and without the acceleration, your upload bandwidth or a system constraint might be limiting your speed.

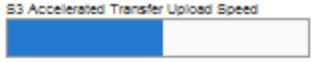
Upload speed comparison in other regions

N. California
(US-WEST-1) 73% faster

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed



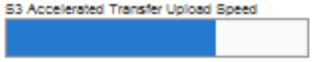
Upload complete

Oregon
(US-WEST-2) 17% slower

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed



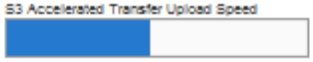
Upload complete

Ireland
(EU-WEST-1) 919% faster

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed



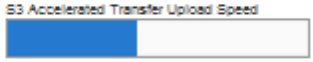
Upload complete

Frankfurt
(EU-CENTRAL-1) 928% faster

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed



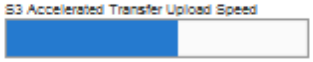
Upload complete

Tokyo
(AP-NORTHEAST-1) 680% faster

S3 Direct Upload Speed

Upload complete

S3 Accelerated Transfer Upload Speed



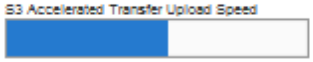
Upload complete

Seoul
(AP-NORTHEAST-2) 822% faster

S3 Direct Upload Speed

Upload complete

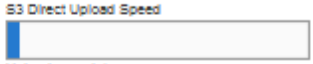
S3 Accelerated Transfer Upload Speed



Upload complete

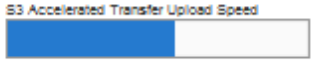
Singapore
(AP-SOUTHEAST-1) 1261% faster

S3 Direct Upload Speed



Upload complete

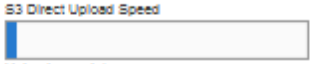
S3 Accelerated Transfer Upload Speed



Upload complete

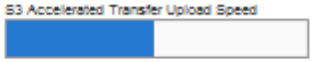
Sydney
(AP-SOUTHEAST-2) 1226% faster

S3 Direct Upload Speed



Upload complete

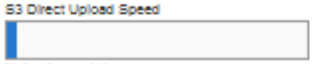
S3 Accelerated Transfer Upload Speed



Upload complete

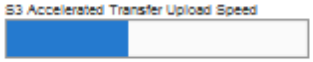
São Paulo
(SA-EAST-1) 1000% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Since the weather forecasting application is located in N.Virginia, you need to transfer all the data in the same AWS Region. With Amazon S3 Transfer Acceleration, you can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Multipart upload allows you to upload a single object as a set of parts. After all the parts of your object are uploaded, Amazon S3 then presents the data as a single object. This approach is the fastest way to aggregate all the data.

Hence, the correct answer is: ***Enable Transfer Acceleration in the destination bucket and upload the collected data using Multipart Upload.***

The option that says: ***Upload the data to the closest S3 bucket. Set up a cross-region replication and copy the objects to the destination bucket*** is incorrect because replicating the objects to the destination bucket takes about 15 minutes. Take note that the requirement in the scenario is to aggregate the data in the fastest way.

The option that says: ***Use AWS Snowball Edge to transfer large amounts of data*** is incorrect because the end-to-end time to transfer up to 80 TB of data into AWS Snowball Edge is approximately one week.

The option that says: ***Set up a Site-to-Site VPN connection*** is incorrect because setting up a VPN connection is not needed in this scenario. Site-to-Site VPN is just used for establishing secure connections between an on-premises network and Amazon VPC. Also, this approach is not the fastest way to transfer your data. You must use Amazon S3 Transfer Acceleration.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/replication.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

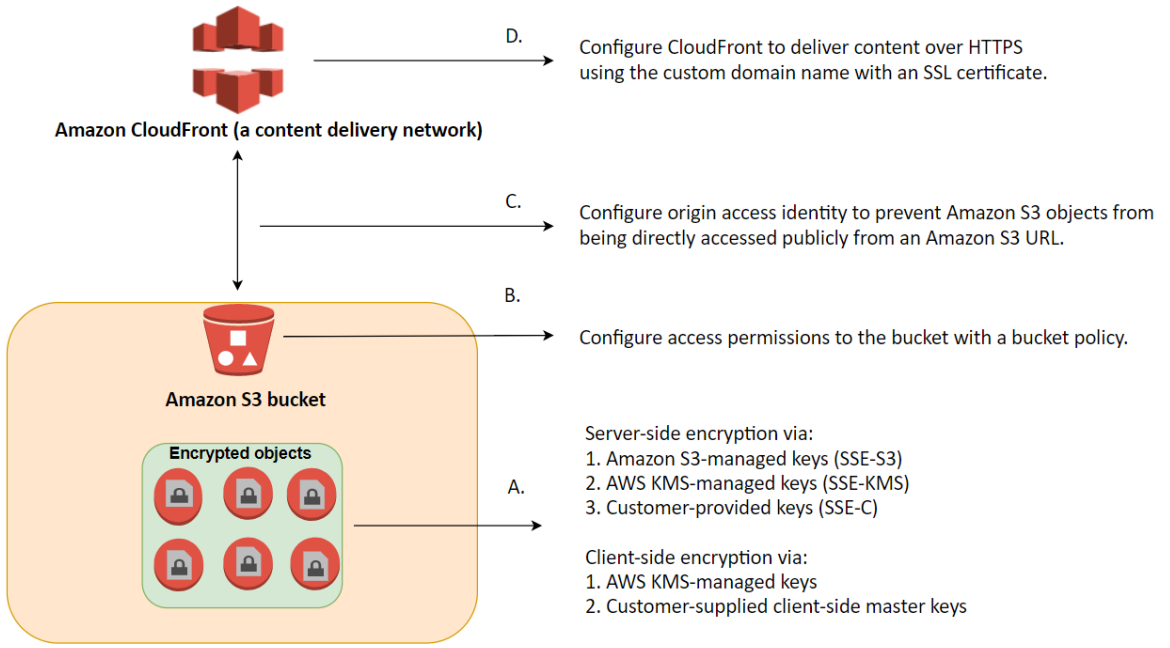
For data privacy, a healthcare company has been asked to comply with the Health Insurance Portability and Accountability Act (HIPAA). The company stores all its backups on an Amazon S3 bucket. It is required that data stored on the S3 bucket must be encrypted.

What is the best option to do this? (Select TWO.)

- Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption.
- Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys.
- Enable Server-Side Encryption on an S3 bucket to make use of AES-128 encryption.
- Store the data on EBS volumes with encryption enabled instead of using Amazon S3.
- Store the data in encrypted EBS snapshots.

Correct

Server-side encryption is about data encryption at rest—that is, Amazon S3 encrypts your data at the object level as it writes it to disks in its data centers and decrypts it for you when you access it. As long as you authenticate your request and you have access permissions, there is no difference in the way you access encrypted or unencrypted objects. For example, if you share your objects using a pre-signed URL, that URL works the same way for both encrypted and unencrypted objects.



You have three mutually exclusive options depending on how you choose to manage the encryption keys:

1. Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)
2. Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)
3. Use Server-Side Encryption with Customer-Provided Keys (SSE-C)

The options that say: ***Before sending the data to Amazon S3 over HTTPS, encrypt the data locally first using your own encryption keys*** and ***Enable Server-Side Encryption on an S3 bucket to make use of AES-256 encryption*** are correct because these options are using client-side encryption and Amazon S3-Managed Keys (SSE-S3) respectively. *Client-side encryption* is the act of encrypting data before sending it to Amazon S3 while SSE-S3 uses AES-256 encryption.

Storing the data on EBS volumes with encryption enabled instead of using Amazon S3 and ***storing the data in encrypted EBS snapshots*** are incorrect because both options use EBS encryption and not S3.

Enabling Server-Side Encryption on an S3 bucket to make use of AES-128 encryption is incorrect as S3 doesn't provide AES-128 encryption, only AES-256.

References:

http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingClientSideEncryption.html

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

8. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

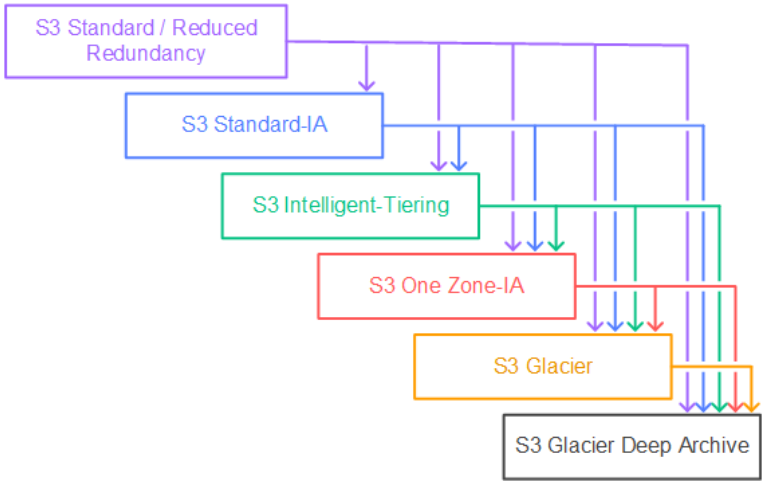
There are a few, easily reproducible but confidential files that your client wants to store in AWS without worrying about storage capacity. For the first month, all of these files will be accessed frequently but after that, they will rarely be accessed at all. The old files will only be accessed by developers so there is no set retrieval time requirement. However, the files under a specific `tdojo-finance` prefix in the S3 bucket will be used for post-processing that requires millisecond retrieval time.

Given these conditions, which of the following options would be the most cost-effective solution for your client's storage needs?

- Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the bucket to Intelligent-Tiering using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the `tdojo-finance` prefix to S3-IA while the remaining go to Glacier using lifecycle policy.
- Store the files in S3 then after a month, change the storage class of the bucket to S3-IA using lifecycle policy.

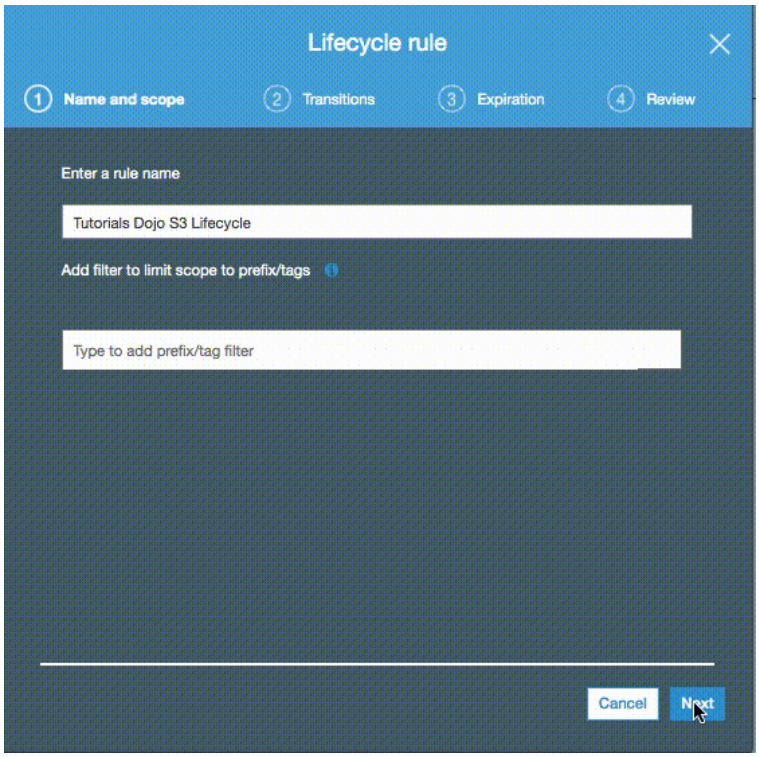
Incorrect

Initially, the files will be accessed frequently, and S3 is a durable and highly available storage solution for that. After a month has passed, the files won't be accessed frequently anymore, so it is a good idea to use lifecycle policies to move them to a storage class that would have a lower cost for storing them.



Since the files are easily reproducible and some of them are needed to be retrieved quickly based on a specific prefix filter (`tdojo-finance`), S3-One Zone IA would be a good choice for storing them. The other files that do not contain such prefix would then be moved to Glacier for low-cost archival. This setup would also be the most cost-effective for the client.

Hence, the correct answer is: ***Store the files in S3 then after a month, change the storage class of the*** `tdojo-finance` ***prefix to One Zone-IA while the remaining go to Glacier using lifecycle policy****.



The option that says: ***Storing the files in S3 then after a month, changing the storage class of the bucket to S3-IA using lifecycle policy*** is incorrect. Although it is valid to move the files to S3-IA, this solution still costs more compared with using a combination of S3-One Zone IA and Glacier.

The option that says: ***Storing the files in S3 then after a month, changing the storage class of the bucket to Intelligent-Tiering using lifecycle policy*** is incorrect. While S3 Intelligent-Tiering can automatically move data between two access tiers (frequent access and infrequent access) when access patterns change, it is more suitable for scenarios where you don't know the access patterns of your data. It may take some time for S3 Intelligent-Tiering to analyze the access patterns before it moves the data to a cheaper storage class like S3-IA which means you may still end up paying more in the beginning. In addition, you already know the access patterns of the files which means you can directly change the storage class immediately and save cost right away.

The option that says: ***Storing the files in S3 then after a month, changing the storage class of the*** `tdojo-finance` ***prefix to S3-IA while the remaining go to Glacier using lifecycle policy*** is incorrect. Even though S3-IA costs less than the S3 Standard storage class, it is still more expensive than S3-One Zone IA. Remember that the files are easily reproducible so you can safely move the data to S3-One Zone IA and in case there is an outage, you can simply generate the missing data again.

References:

https://aws.amazon.com/blogs/compute/amazon-s3-adds-prefix-and-suffix-filters-for-lambda-function-triggering

https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/lifecycle-configuration-examples.html

https://aws.amazon.com/s3/pricing

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/

1. QUESTION

Category: CSAA – Design High-Performing Architectures

A company has a web-based ticketing service that utilizes Amazon SQS and a fleet of EC2 instances. The EC2 instances that consume messages from the SQS queue are configured to poll the queue as often as possible to keep end-to-end throughput as high as possible. The Solutions Architect noticed that polling the queue in tight loops is using unnecessary CPU cycles, resulting in increased operational costs due to empty responses.

In this scenario, what should the Solutions Architect do to make the system more cost-effective?

- Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to zero.
- Configure Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero.
- Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to zero.
- Configure Amazon SQS to use short polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero.

Correct

In this scenario, the application is deployed in a fleet of EC2 instances that are polling messages from a single SQS queue. Amazon SQS uses short polling by default, querying only a subset of the servers (based on a weighted random distribution) to determine whether any messages are available for inclusion in the response. Short polling works for scenarios that require higher throughput. However, you can also configure the queue to use Long polling instead, to reduce cost.

The ReceiveMessageWaitTimeSeconds is the queue attribute that determines whether you are using Short or Long polling. By default, its value is zero which means it is using Short polling. If it is set to a value greater than zero, then it is Long polling.

Hence, ***configuring Amazon SQS to use long polling by setting the ReceiveMessageWaitTimeSeconds to a number greater than zero is the correct answer.***

Quick facts about SQS Long Polling:

– Long polling helps reduce your cost of using Amazon SQS by reducing the number of empty responses when there are no messages available to return in reply to a *ReceiveMessage* request sent to an Amazon SQS queue and eliminating false empty responses when messages are available in the queue but aren't included in the response.

– Long polling reduces the number of empty responses by allowing Amazon SQS to wait until a message is available in the queue before sending a response. Unless the connection times out, the response to the `ReceiveMessage` request contains at least one of the available messages, up to the maximum number of messages specified in the `ReceiveMessage` action.

– Long polling eliminates false empty responses by querying all (rather than a limited number) of the servers. Long polling returns messages as soon any message becomes available.

Reference:

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-long-polling.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

2. QUESTION

Category: CSAA – Design High-Performing Architectures

The start-up company that you are working for has a batch job application that is currently hosted on an EC2 instance. It is set to process messages from a queue created in SQS with default settings. You configured the application to process the messages once a week. After 2 weeks, you noticed that not all messages are being processed by the application.

What is the root cause of this issue?

- Missing permissions in SQS.
- The SQS queue is set to short-polling.
- Amazon SQS has automatically deleted the messages that have been in a queue for more than the maximum message retention period.
- The batch job application is configured to long polling.

Incorrect

Amazon SQS automatically deletes messages that have been in a queue for more than the maximum message retention period. The default message retention period is 4 days. Since the queue is configured to the default settings and the batch job application only processes the messages once a week, the messages that are in the queue for more than 4 days are deleted. This is the root cause of the issue.

To fix this, you can increase the message retention period to a maximum of 14 days using the SetQueueAttributes action.

References:

https://aws.amazon.com/sqs/faqs/

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-message-lifecycle.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

3. QUESTION

Category: CSAA – Design High-Performing Architectures

An e-commerce application is using a fanout messaging pattern for its order management system. For every order, it sends an Amazon SNS message to an SNS topic, and the message is replicated and pushed to multiple Amazon SQS queues for parallel asynchronous processing. A Spot EC2 instance retrieves the message from each SQS queue and processes the message. There was an incident that while an EC2 instance is currently processing a message, the instance was abruptly terminated, and the processing was not completed in time.

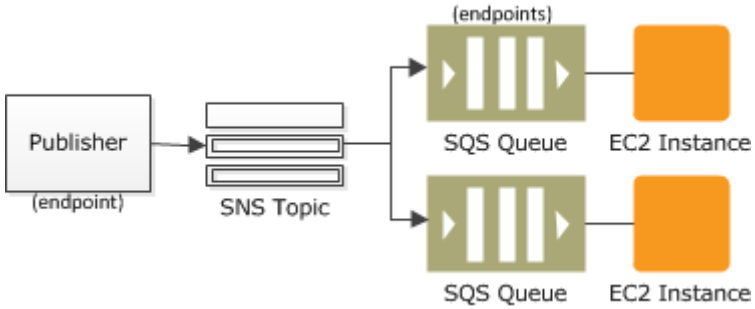
In this scenario, what happens to the SQS message?

- The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online.
- The message will be sent to a Dead Letter Queue in AWS DataSync.
- When the message visibility timeout expires, the message becomes available for processing by other EC2 instances
- The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout.

Correct

A “fanout” pattern is when an Amazon SNS message is sent to a topic and then replicated and pushed to multiple Amazon SQS queues, HTTP endpoints, or email addresses. This allows for parallel asynchronous processing. For example, you could develop an application that sends an Amazon SNS message to a topic whenever an order is placed for a product. Then, the Amazon SQS queues that are subscribed to that

topic would receive identical notifications for the new order. The Amazon EC2 server instance attached to one of the queues could handle the processing or fulfillment of the order, while the other server instance could be attached to a data warehouse for analysis of all orders received.



When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a *visibility timeout*, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

The option that says: ***The message will automatically be assigned to the same EC2 instance when it comes back online within or after the visibility timeout*** is incorrect because the message will not be automatically assigned to the same EC2 instance once it is abruptly terminated. When the message visibility timeout expires, the message becomes available for processing by other EC2 instances.

The option that says: ***The message is deleted and becomes duplicated in the SQS when the EC2 instance comes online*** is incorrect because the message will not be deleted and won't be duplicated in the SQS queue when the EC2 instance comes online.

The option that says: ***The message will be sent to a Dead Letter Queue in AWS DataSync*** is incorrect because although the message could be programmatically sent to a Dead Letter Queue (DLQ), it won't be handled by AWS DataSync but by Amazon SQS instead. AWS DataSync is primarily used to simplify your migration with AWS. It makes it simple and fast to move large amounts of data online between on-premises storage and Amazon S3 or Amazon Elastic File System (Amazon EFS).

References:

http://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

https://docs.aws.amazon.com/sns/latest/dg/sns-common-scenarios.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs

4. QUESTION

Category: CSAA – Design Resilient Architectures

An investment bank has a distributed batch processing application which is hosted in an Auto Scaling group of Spot EC2 instances with an SQS queue. You configured your components to use client-side buffering so that the calls made from the client will be buffered first and then sent as a batch request to SQS.

What is a period of time during which the SQS queue prevents other consuming components from receiving and processing a message?

- Visibility Timeout
- Component Timeout
- Receiving Timeout
- Processing Timeout

Correct

The visibility timeout is a period of time during which Amazon SQS prevents other consuming components from receiving and processing a message.

When a consumer receives and processes a message from a queue, the message remains in the queue. Amazon SQS doesn't automatically delete the message. Because Amazon SQS is a distributed system, there's no guarantee that the consumer actually receives the message (for example, due to a connectivity issue, or due to an issue in the consumer application). Thus, the consumer must delete the message from the queue after receiving and processing it.

Immediately after the message is received, it remains in the queue. To prevent other consumers from processing the message again, Amazon SQS sets a ***visibility timeout***, a period of time during which Amazon SQS prevents other consumers from receiving and processing the message. The default visibility timeout for a message is 30 seconds. The maximum is 12 hours.

References:

https://aws.amazon.com/sqs/faqs/

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-visibility-timeout.html

Check out this Amazon SQS Cheat Sheet:

https://tutorialsdojo.com/amazon-sqs/

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company needs to integrate the Lightweight Directory Access Protocol (LDAP) directory service from the on-premises data center to the AWS VPC using IAM. The identity store which is currently being used is not compatible with SAML.

Which of the following provides the most valid approach to implement the integration?

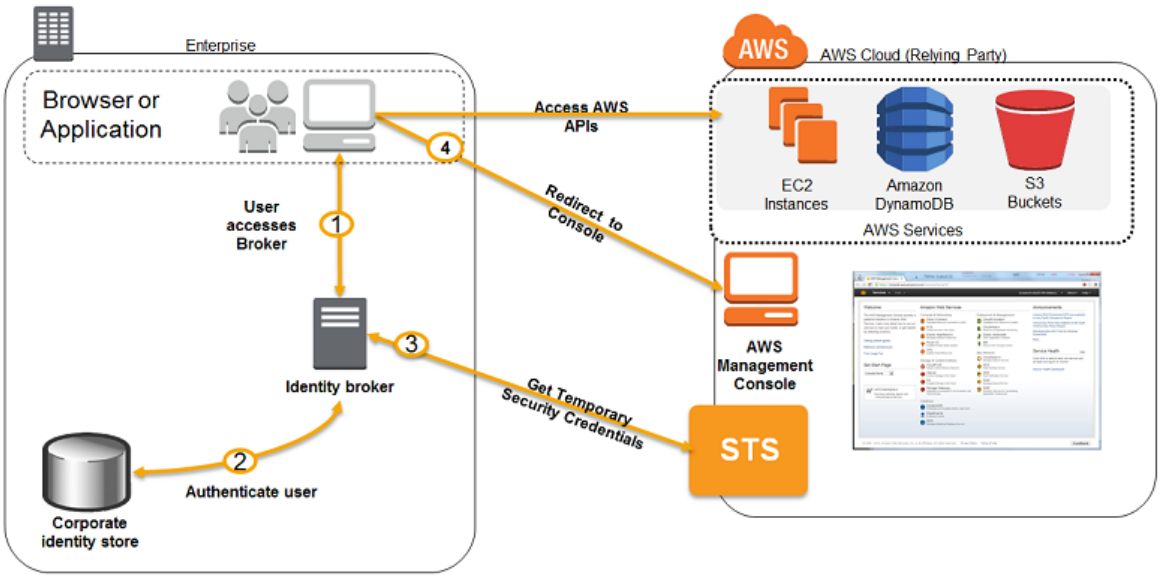
- Use AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP.
- Use an IAM policy that references the LDAP identifiers and AWS credentials.
- Use IAM roles to rotate the IAM credentials whenever LDAP credentials are updated.
- Develop an on-premises custom identity broker application and use STS to issue short-lived AWS credentials.

Incorrect

If your identity store is not compatible with SAML 2.0 then you can build a custom identity broker application to perform a similar function. The broker application authenticates users, requests temporary credentials for users from AWS, and then provides them to the user to access AWS resources.

The application verifies that employees are signed into the existing corporate network's identity and authentication system, which might use LDAP, Active Directory, or another system. The identity broker application then obtains temporary security credentials for the employees.

To get temporary security credentials, the identity broker application calls either `AssumeRole` or `GetFederationToken` to obtain temporary security credentials, depending on how you want to manage the policies for users and when the temporary credentials should expire. The call returns temporary security credentials consisting of an AWS access key ID, a secret access key, and a session token. The identity broker application makes these temporary security credentials available to the internal company application. The app can then use the temporary credentials to make calls to AWS directly. The app caches the credentials until they expire, and then requests a new set of temporary credentials.



Using an IAM policy that references the LDAP identifiers and AWS credentials is incorrect because using an IAM policy is not enough to integrate your LDAP service to IAM. You need to use SAML, STS, or a custom identity broker.

Using AWS Single Sign-On (SSO) service to enable single sign-on between AWS and your LDAP is incorrect because the scenario did not require SSO and in addition, the identity store that you are using is not SAML-compatible.

Using IAM roles to rotate the IAM credentials whenever LDAP credentials are updated is incorrect because manually rotating the IAM credentials is not an optimal solution to integrate your on-premises and VPC network. You need to use SAML, STS, or a custom identity broker.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_common-scenarios_federated-users.html

https://aws.amazon.com/blogs/aws/aws-identity-and-access-management-now-with-identity-federation/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

2. QUESTION

Category: CSAA – Design Secure Applications and Architectures

An Intelligence Agency developed a missile tracking application that is hosted on both development and production AWS accounts. The Intelligence agency's junior developer only has access to the development account. She has received security clearance to access the agency's production account but the access is only temporary and only write access to EC2 and S3 is allowed.

Which of the following allows you to issue short-lived access tokens that act as temporary security credentials to allow access to your AWS resources?

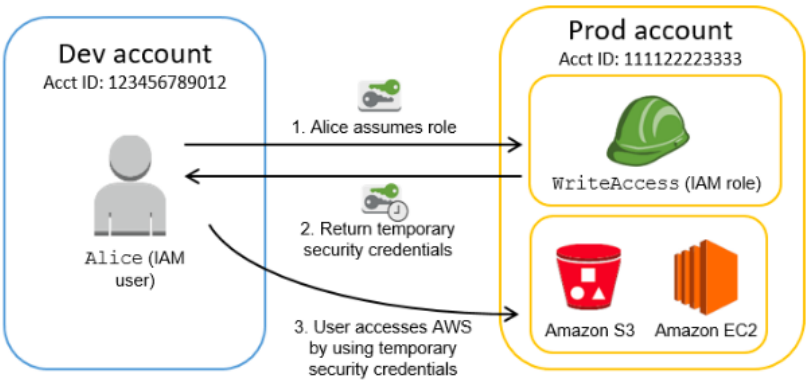
- Use AWS Cognito to issue JSON Web Tokens (JWT)
- All of the given options are correct.
- Use AWS STS
- Use AWS SSO

Correct

AWS Security Token Service (AWS STS) is the service that you can use to create and provide trusted users with temporary security credentials that can control access to your AWS resources. Temporary security credentials work almost identically to the long-term access key credentials that your IAM users can use.

In this diagram, IAM user Alice in the Dev account (the role-assuming account) needs to access the Prod account (the role-owning account). Here's how it works:

1. Alice in the Dev account assumes an IAM role (WriteAccess) in the Prod account by calling AssumeRole.
2. STS returns a set of temporary security credentials.
3. Alice uses the temporary security credentials to access services and resources in the Prod account. Alice could, for example, make calls to Amazon S3 and Amazon EC2, which are granted by the WriteAccess role.



Using AWS Cognito to issue JSON Web Tokens (JWT) is incorrect because the Amazon Cognito service is primarily used for user authentication and not for providing access to your AWS resources. A JSON Web Token (JWT) is meant to be used for user authentication and session management.

Using AWS SSO is incorrect. Although the AWS SSO service uses STS, it does not issue short-lived credentials by itself. AWS Single Sign-On (SSO) is a cloud SSO service that makes it easy to centrally manage SSO access to multiple AWS accounts and business applications.

The option that says ***All of the above*** is incorrect as only STS has the ability to provide temporary security credentials.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_temp.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

3. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A tech company that you are working for has undertaken a Total Cost Of Ownership (TCO) analysis evaluating the use of Amazon S3 versus acquiring more storage hardware. The result was that all 1200 employees would be granted access to use Amazon S3 for storage of their personal documents.

Which of the following will you need to consider so you can set up a solution that incorporates single sign-on feature from your corporate AD or LDAP directory and also restricts access for each individual user to a designated user folder in an S3 bucket? (Select TWO.)

- Configure an IAM role and an IAM Policy to access the bucket.
- Set up a Federation proxy or an Identity provider, and use AWS Security Token Service to generate temporary tokens.
- Set up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket.
- Use 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others.
- Map each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents.

Incorrect

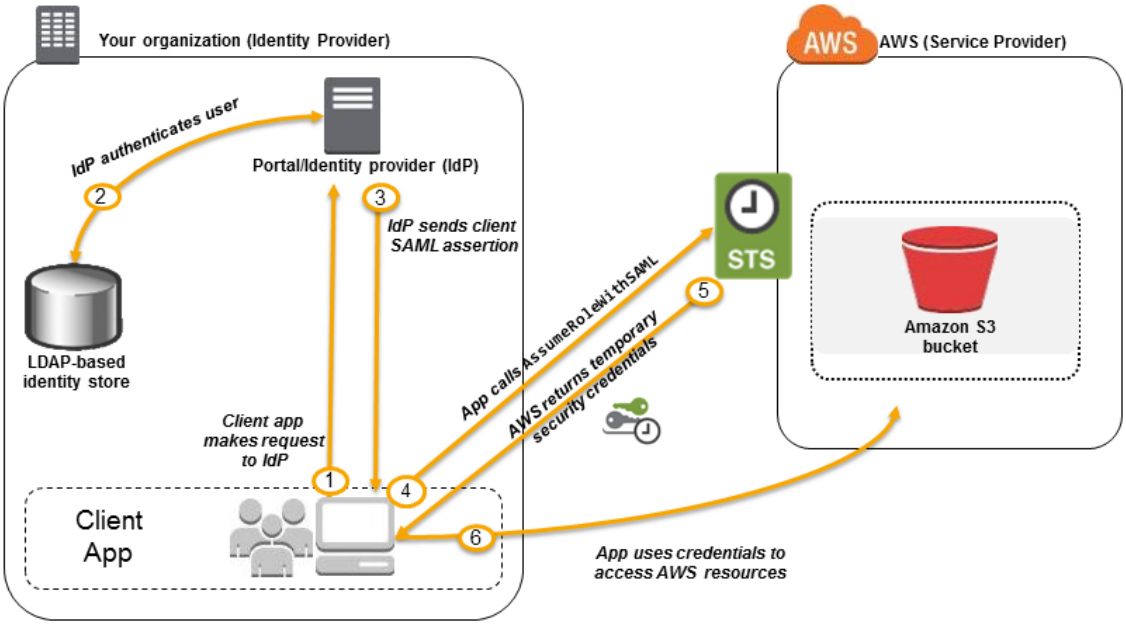
The question refers to one of the common scenarios for temporary credentials in AWS. Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles. In this example, it is called **enterprise identity federation** considering that you also need to set up a single sign-on (SSO) capability.

The correct answers are:

– Setup a Federation proxy or an Identity provider

– Setup an AWS Security Token Service to generate temporary tokens

– Configure an IAM role and an IAM Policy to access the bucket.



In an enterprise identity federation, you can authenticate users in your organization's network, and then provide those users access to AWS without creating new AWS identities for them and requiring them to sign in with a separate user name and password. This is known as the *single sign-on* (SSO) approach to temporary access. AWS STS supports open standards like Security Assertion Markup Language (SAML) 2.0, with which you can use Microsoft AD FS to leverage your Microsoft Active Directory. You can also use SAML 2.0 to manage your own solution for federating user identities.

Using 3rd party Single Sign-On solutions such as Atlassian Crowd, OKTA, OneLogin and many others is incorrect since you don't have to use 3rd party solutions to provide the access. AWS already provides the necessary tools that you can use in this situation.

Mapping each individual user to a designated user folder in S3 using Amazon WorkDocs to access their personal documents is incorrect as there is no direct way of integrating Amazon S3 with Amazon WorkDocs for this particular scenario. Amazon WorkDocs is simply a fully managed, secure content creation, storage, and collaboration service. With Amazon WorkDocs, you can easily create, edit, and share content. And because it's stored centrally on AWS, you can access it from anywhere on any device.

Setting up a matching IAM user for each of the 1200 users in your corporate directory that needs access to a folder in the S3 bucket is incorrect since creating that many IAM users would be unnecessary. Also, you want the account to integrate with your AD or LDAP directory, hence, IAM Users does not fit these criteria.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc.html

https://aws.amazon.com/blogs/security/writing-iam-policies-grant-access-to-user-specific-folders-in-an-amazon-s3-bucket/

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A mobile application stores pictures in Amazon Simple Storage Service (S3) and allows application sign-in using an OpenID Connect-compatible identity provider.

Which AWS Security Token Service approach to temporary access should you use for this scenario?

- Web Identity Federation
- SAML-based Identity Federation
- Cross-Account Access
- AWS Identity and Access Management roles

Incorrect

With web identity federation, you don't need to create custom sign-in code or manage your own user identities. Instead, users of your app can sign in using a well-known identity provider (IdP) —such as Login with Amazon, Facebook, Google, or any other OpenID Connect (OIDC)-compatible IdP, receive an authentication token, and then exchange that token for temporary security credentials in AWS that map to an IAM role with permissions to use the resources in your AWS account. Using an IdP helps you keep your AWS account secure because you don't have to embed and distribute long-term security credentials with your application.

Reference:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

5. QUESTION

Catgory: CSAA – Design Secure Applications and Architectures

A Solutions Architect is managing a company's AWS account of approximately 300 IAM users. They have a new company policy that requires changing the associated permissions of all 100 IAM users that control the access to Amazon S3 buckets.

What will the Solutions Architect do to avoid the time-consuming task of applying the policy to each user?

- Create a new S3 bucket access policy with unlimited access for each IAM user.
- Create a new IAM group and then add the users that require access to the S3 bucket. Afterwards, apply the policy to IAM group.
- Create a new IAM role and add each user to the IAM role.
- Create a new policy and apply it to multiple IAM users using a shell script.

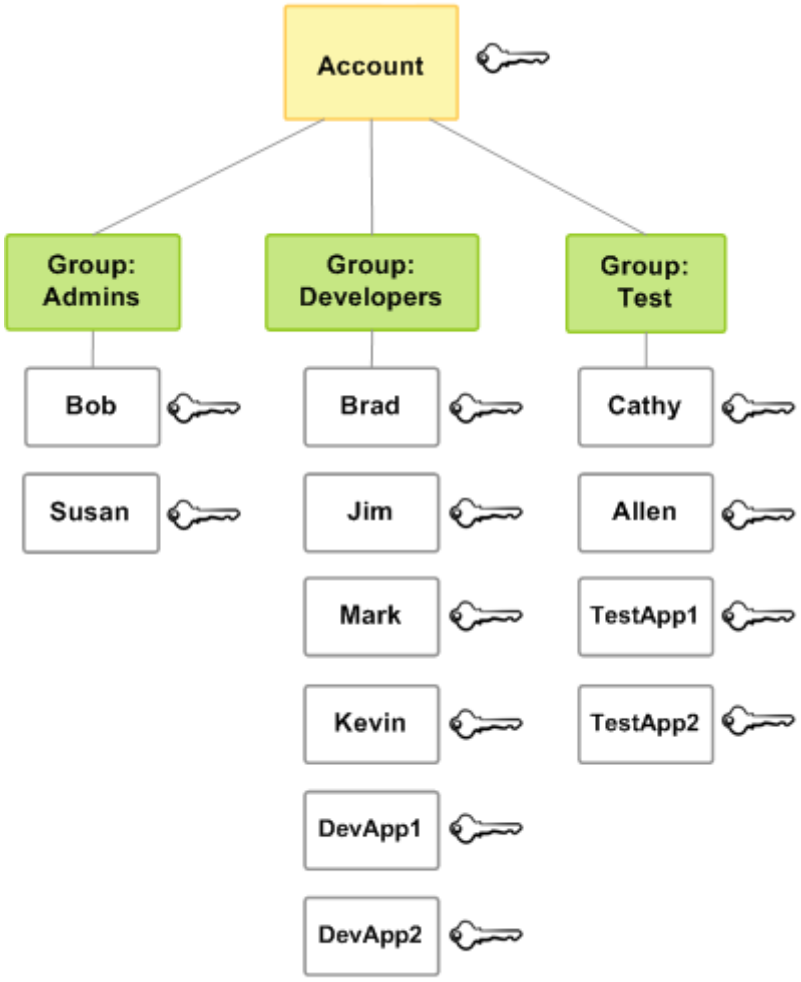
Correct

In this scenario, the best option is to ***group the set of users in an IAM Group and then apply a policy with the required access to the Amazon S3 bucket***. This will enable you to easily add, remove, and manage the users instead of manually adding a policy to each and every 100 IAM users.

Creating a new policy and applying it to multiple IAM users using a shell script is incorrect because you need a new IAM Group for this scenario and not assign a policy to each user via a shell script. This method can save you time but afterward, it will be difficult to manage all 100 users that are not contained in an IAM Group.

Creating a new S3 bucket access policy with unlimited access for each IAM user is incorrect because you need a new IAM Group and the method is also time-consuming.

Creating a new IAM role and adding each user to the IAM role is incorrect because you need to use an IAM Group and not an IAM role.



Reference:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Tutorials Dojo's AWS Certified Solutions Architect Associate Exam Study Guide:

https://tutorialsdojo.com/aws-certified-solutions-architect-associate/

6. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A Solutions Architect created a brand new IAM User with a default setting using AWS CLI. This is intended to be used to send API requests to Amazon S3, DynamoDB, Lambda, and other AWS resources of the company's cloud infrastructure.

Which of the following must be done to allow the user to make API calls to the AWS resources?

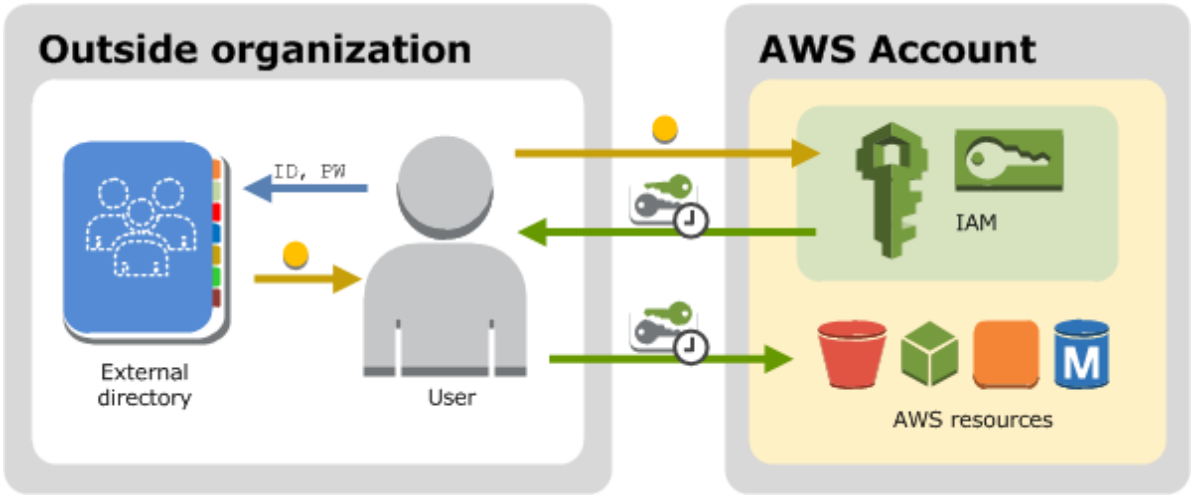
- Do nothing as the IAM User is already capable of sending API calls to your AWS resources.
- Create a set of Access Keys for the user and attach the necessary permissions.
- Enable Multi-Factor Authentication for the user.
- Assign an IAM Policy to the user to allow it to send API calls.

Incorrect

You can choose the credentials that are right for your IAM user. When you use the AWS Management Console to create a user, you must choose to at least include a console password or access keys. By default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. You must create the type of credentials for an IAM user based on the needs of your user.

Access keys are long-term credentials for an IAM user or the AWS account root user. You can use access keys to sign programmatic requests to the AWS CLI or AWS API (directly or using the AWS SDK). Users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.

To fill this need, you can create, modify, view, or rotate access keys (access key IDs and secret access keys) for IAM users. When you create an access key, IAM returns the access key ID and secret access key. You should save these in a secure location and give them to the user.



The option that says: ***Do nothing as the IAM User is already capable of sending API calls to your AWS resources*** is incorrect because by default, a brand new IAM user created using the AWS CLI or AWS API has no credentials of any kind. Take note that in the scenario, you created the new IAM user using the AWS CLI and not via the AWS Management Console, where you must choose to at least include a console password or access keys when creating a new IAM user.

Enabling Multi-Factor Authentication for the user is incorrect because this will still not provide the required Access Keys needed to send API calls to your AWS resources. You have to grant the IAM user with Access Keys to meet the requirement.

Assigning an IAM Policy to the user to allow it to send API calls is incorrect because adding a new IAM policy to the new user will not grant the needed Access Keys needed to make API calls to the AWS resources.

References:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users.html#id_users_creds

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

7. QUESTION

Category: CSAA – Design Secure Applications and Architectures

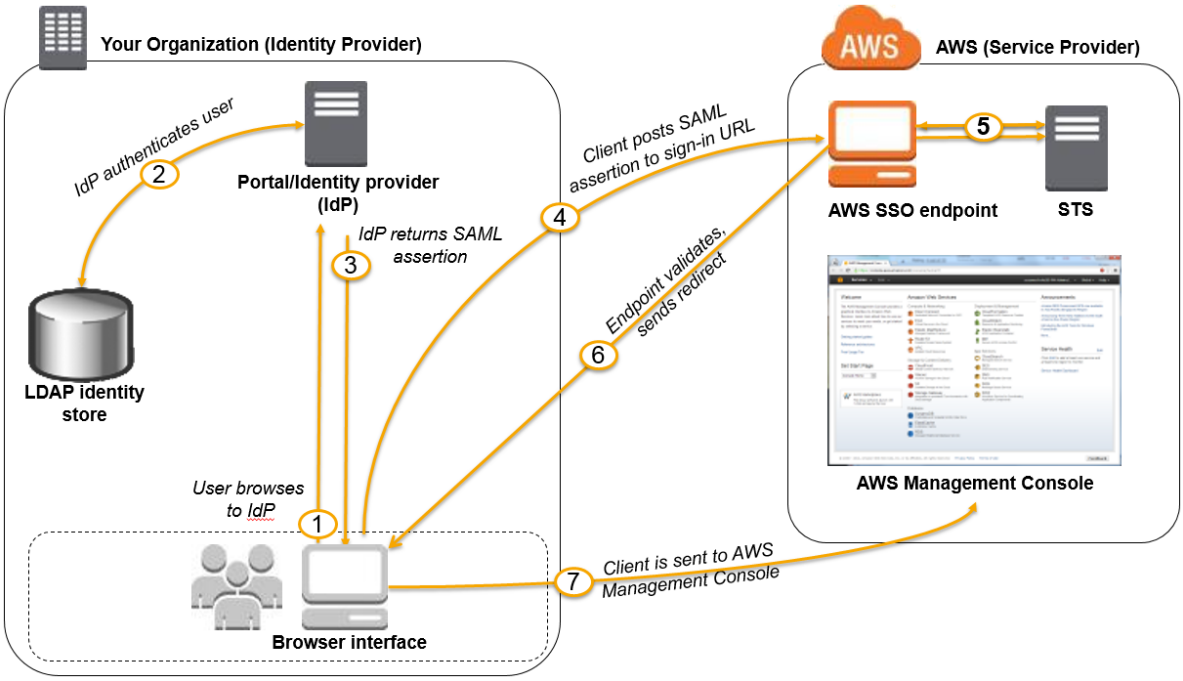
A pharmaceutical company has resources hosted on both their on-premises network and in AWS cloud. They want all of their Software Architects to access resources on both environments using their on-premises credentials, which is stored in Active Directory.

In this scenario, which of the following can be used to fulfill this requirement?

- Set up SAML 2.0-Based Federation by using a Web Identity Federation.
- Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).
- Use IAM users
- Use Amazon VPC

Incorrect

Since the company is using Microsoft Active Directory which implements Security Assertion Markup Language (SAML), you can set up a SAML-Based Federation for API Access to your AWS cloud. In this way, you can easily connect to AWS using the login credentials of your on-premises network.



AWS supports identity federation with SAML 2.0, an open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS APIs without you having to create an IAM user for everyone in your organization. By using SAML, you can simplify the process of configuring federation with AWS, because you can use the IdP's service instead of writing custom identity proxy code.

Before you can use SAML 2.0-based federation as described in the preceding scenario and diagram, you must configure your organization's IdP and your AWS account to trust each other. The general process for configuring this trust is described in the following steps. Inside your organization, you must have an IdP that supports SAML 2.0, like Microsoft Active Directory Federation Service (AD FS, part of Windows Server), Shibboleth, or another compatible SAML 2.0 provider.

Hence, the correct answer is: ***Set up SAML 2.0-Based Federation by using a Microsoft Active Directory Federation Service (AD FS).***

Setting up SAML 2.0-Based Federation by using a Web Identity Federation is incorrect because this is primarily used to let users sign in via a well-known external identity provider (IdP), such as Login with Amazon, Facebook, Google. It does not utilize Active Directory.

Using IAM users is incorrect because the situation requires you to use the existing credentials stored in their Active Directory, and not user accounts that will be generated by IAM.

Using Amazon VPC is incorrect because this only lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. This has nothing to do with user authentication or Active Directory.

References:

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_saml.html

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers.html

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

8. QUESTION

Category: CSAA – Design Secure Applications and Architectures

A company recently adopted a hybrid architecture that integrates its on-premises data center to AWS cloud. You are assigned to configure the VPC and implement the required IAM users, IAM roles, IAM groups, and IAM policies.

In this scenario, what is the best practice when creating IAM policies?

- Use the principle of least privilege which means granting only the permissions required to perform a task.
- Determine what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations.
- Grant all permissions to any EC2 user.
- Use the principle of least privilege which means granting only the least number of people with full root access.

Correct

One of the best practices in AWS IAM is to **grant least privilege**.

When you create IAM policies, follow the standard security advice of granting *least privilege*—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies for them that let the users perform *only* those tasks.

Therefore, ****using the principle of least privilege which means granting only the permissions required to perform a task**** is the correct answer.

Start with a minimum set of permissions and grant additional permissions as necessary. Defining the right set of permissions requires some understanding of the user's objectives. Determine what is required for the specific task, what actions a particular service supports, and what permissions are required in order to perform those actions.

****Granting all permissions to any EC2 user**** is incorrect since you don't want your users to gain access to everything and perform unnecessary actions. Doing so is not a good security practice.

****Using the principle of least privilege which means granting only the least number of people with full root access**** is incorrect because this is not the correct definition of what the principle of least privilege is.

****Determining what users need to do and then craft policies for them that let the users perform those tasks including additional administrative operations**** is incorrect since there are some users who you should not give administrative access to. You should follow the principle of least privilege when providing permissions and accesses to your resources.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#use-groups-for-permissions

Check out this AWS IAM Cheat Sheet:

https://tutorialsdojo.com/aws-identity-and-access-management-iam/

Service Control Policies (SCP) vs IAM Policies:

https://tutorialsdojo.com/service-control-policies-scp-vs-iam-policies/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services/

1. QUESTION

Category: CSAA – Design Secure Applications and Architectures

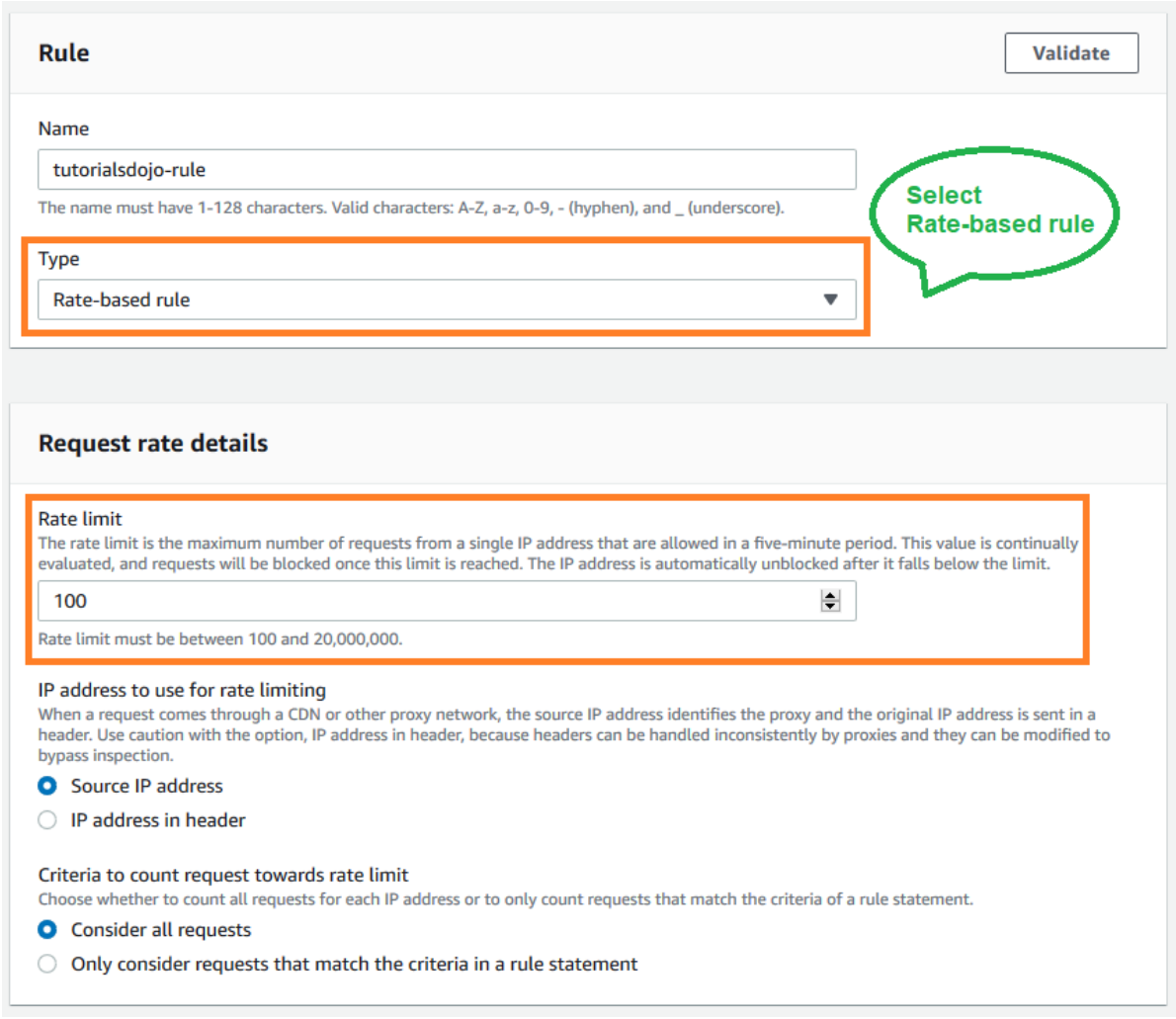
A company hosted an e-commerce website on an Auto Scaling group of EC2 instances behind an Application Load Balancer. The Solutions Architect noticed that the website is receiving a large number of illegitimate external requests from multiple systems with IP addresses that constantly change. To resolve the performance issues, the Solutions Architect must implement a solution that would block the illegitimate requests with minimal impact on legitimate traffic.

Which of the following options fulfills this requirement?

- Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.
- Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer.
- Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests.
- Create a custom rule in the security group of the Application Load Balancer to block the offending requests.

Correct

AWS WAF is tightly integrated with Amazon CloudFront, the Application Load Balancer (ALB), Amazon API Gateway, and AWS AppSync – services that AWS customers commonly use to deliver content for their websites and applications. When you use AWS WAF on Amazon CloudFront, your rules run in all AWS Edge Locations, located around the world close to your end-users. This means security doesn't come at the expense of performance. Blocked requests are stopped before they reach your web servers. When you use AWS WAF on regional services, such as Application Load Balancer, Amazon API Gateway, and AWS AppSync, your rules run in the region and can be used to protect Internet-facing resources as well as internal resources.
WAF: Web Application Firewall



A rate-based rule tracks the rate of requests for each originating IP address and triggers the rule action on IPs with rates that go over a limit. You set the limit as the number of requests per 5-minute time span. You can use this type of rule to put a temporary block on requests from an IP address that's sending excessive requests.

Based on the given scenario, the requirement is to limit the number of requests from the illegitimate requests without affecting the genuine requests. To accomplish this requirement, you can use AWS WAF web ACL. There are two types of rules in creating your own web ACL rule: regular and rate-based rules. You need to select the latter to add a rate limit to your web ACL. After creating the web ACL, you can associate it with ALB. When the rule action triggers, AWS WAF applies the action to additional requests from the IP address until the request rate falls below the limit.

Hence, the correct answer is: ***Create a rate-based rule in AWS WAF and associate the web ACL to an Application Load Balancer.***

The option that says: ***Create a regular rule in AWS WAF and associate the web ACL to an Application Load Balancer*** is incorrect because a regular rule only matches the statement defined in the rule. If you need to add a rate limit to your rule, you should create a rate-based rule.

The option that says: ***Create a custom network ACL and associate it with the subnet of the Application Load Balancer to block the offending requests*** is incorrect. Although NACLs can help you block incoming traffic, this option wouldn't be able to limit the number of requests from a single IP address that is dynamically changing.

The option that says: ***Create a custom rule in the security group of the Application Load Balancer to block the offending requests*** is incorrect because the security group can only allow incoming traffic. Remember that you can't deny traffic using security groups. In addition, it is not capable of limiting the rate of traffic to your application unlike AWS WAF.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-rate-based.html

https://aws.amazon.com/waf/faqs/

Check out this AWS WAF Cheat Sheet:

https://tutorialsdojo.com/aws-waf/

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:

https://youtu.be/-1S-RdeAmMo

2. QUESTION

Category: CSAA – Design Resilient Architectures

A DevOps Engineer is required to design a cloud architecture in AWS. The Engineer is planning to develop a highly available and fault-tolerant architecture that is composed of an Elastic Load Balancer and an Auto Scaling group of EC2 instances deployed across multiple Availability Zones. This will be used by an online accounting application that requires path-based routing, host-based routing, and bi-directional communication channels using WebSockets.

Which is the most suitable type of Elastic Load Balancer that will satisfy the given requirement?

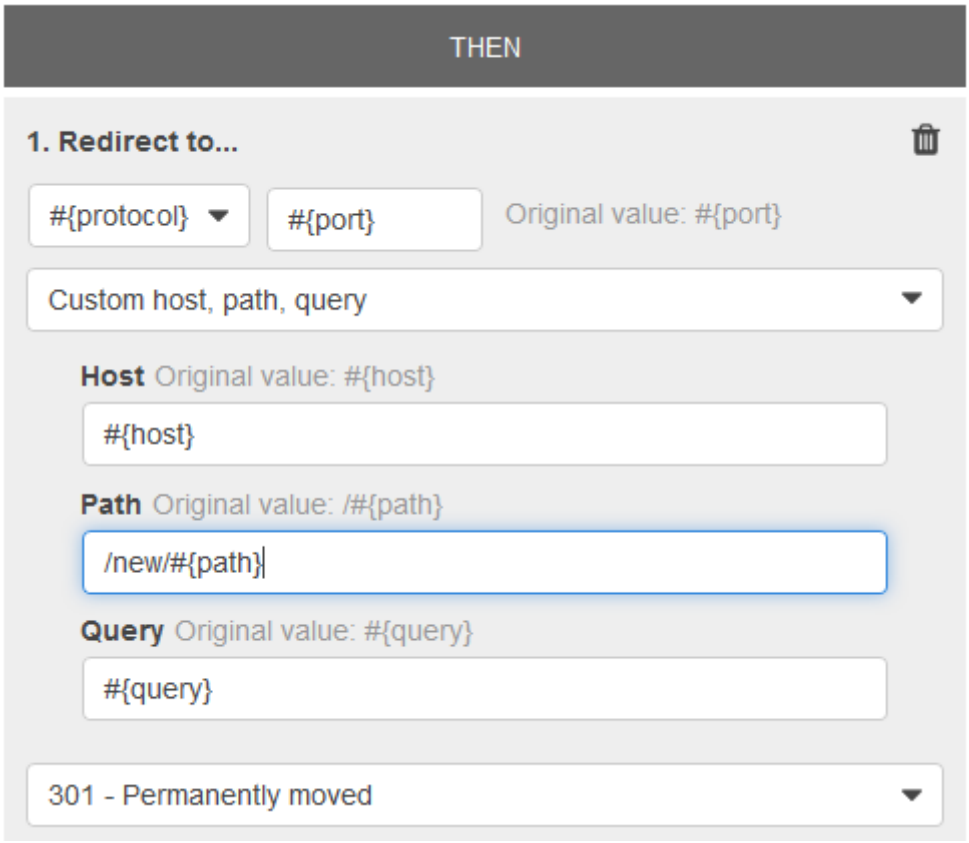
- Either a Classic Load Balancer or a Network Load Balancer
- Classic Load Balancer
- Network Load Balancer
- Application Load Balancer

Incorrect

Elastic Load Balancing supports three types of load balancers. You can select the appropriate load balancer based on your application needs.

If you need flexible application management and TLS termination then it is recommended to use Application Load Balancer. If extreme performance and static IP is needed for your application then it is recommend that you use Network Load Balancer. If your application is built within the EC2 Classic network then you should use Classic Load Balancer.

An **Application Load Balancer** functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action. You can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.



Application Load Balancers support path-based routing, host-based routing, and support for containerized applications hence, ***Application Load Balancer*** is the correct answer.

Network Load Balancer, ***Classic Load Balancer***, and ***either a Classic Load Balancer or a Network Load Balancer*** are all incorrect as none of these support path-based routing and host-based routing, unlike an Application Load Balancer.

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits

https://aws.amazon.com/elasticloadbalancing/faqs/

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

3. QUESTION

Category: CSAA – Design Secure Applications and Architectures

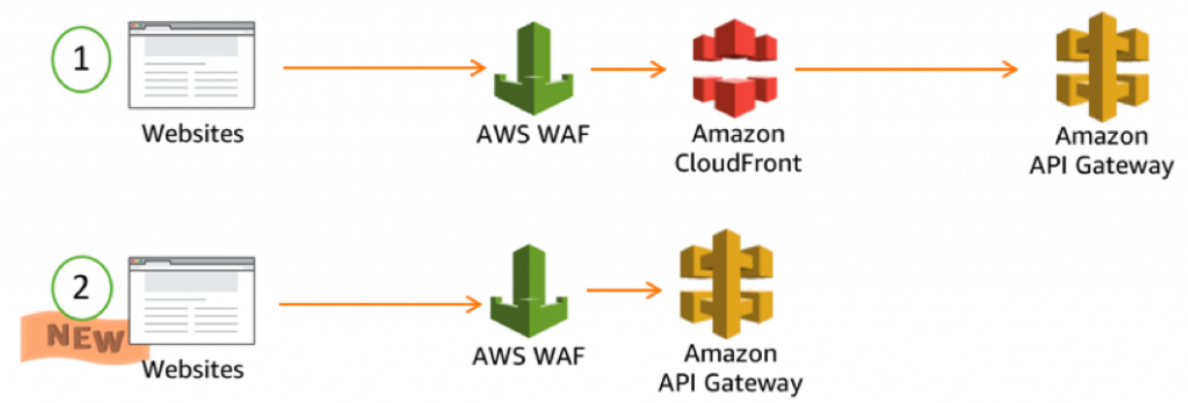
A company is hosting its web application in an Auto Scaling group of EC2 instances behind an Application Load Balancer. Recently, the Solutions Architect identified a series of SQL injection attempts and cross-site scripting attacks to the application, which had adversely affected their production data.

Which of the following should the Architect implement to mitigate this kind of attack?

- Block all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List.
- Using AWS Firewall Manager, set up security rules that block SQL injection and cross-site scripting attacks. Associate the rules to the Application Load Balancer.
- Use Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application.
- Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.

Correct

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to an Amazon API Gateway API, Amazon CloudFront or an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, API Gateway, CloudFront or an Application Load Balancer responds to requests either with the requested content or with an HTTP 403 status code (Forbidden). You also can configure CloudFront to return a custom error page when a request is blocked.



At the simplest level, AWS WAF lets you choose one of the following behaviors:

Allow all requests except the ones that you specify – This is useful when you want CloudFront or an Application Load Balancer to serve content for a public website, but you also want to block requests from attackers.

Block all requests except the ones that you specify – This is useful when you want to serve content for a restricted website whose users are readily identifiable by properties in web requests, such as the IP addresses that they use to browse to the website.

Count the requests that match the properties that you specify – When you want to allow or block requests based on new properties in web requests, you first can configure AWS WAF to count the requests that match those properties without allowing or blocking those requests. This lets you confirm that you didn't accidentally configure AWS WAF to block all the traffic to your website. When you're confident that you specified the correct properties, you can change the behavior to allow or block requests.

Hence, the correct answer in this scenario is: ***Set up security rules that block SQL injection and cross-site scripting attacks in AWS Web Application Firewall (WAF). Associate the rules to the Application Load Balancer.****

Using Amazon GuardDuty to prevent any further SQL injection and cross-site scripting attacks in your application* is incorrect because Amazon GuardDuty is just a threat detection service that continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads.

Using AWS Firewall Manager to set up security rules that block SQL injection and cross-site scripting attacks, then associating the rules to the Application Load Balancer* is incorrect because AWS Firewall Manager just simplifies your AWS WAF and AWS Shield Advanced administration and maintenance tasks across multiple accounts and resources.

Blocking all the IP addresses where the SQL injection and cross-site scripting attacks originated using the Network Access Control List* is incorrect because this is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. NACLs are not effective in blocking SQL injection and cross-site scripting attacks

References:

https://aws.amazon.com/waf/

https://docs.aws.amazon.com/waf/latest/developerguide/what-is-aws-waf.html

Check out this AWS WAF Cheat Sheet:

https://tutorialsdojo.com/aws-waf/

AWS Security Services Overview – WAF, Shield, CloudHSM, KMS:*

https://youtu.be/-1S-RdeAmMo

4. QUESTION

Category: CSAA – Design Secure Applications and Architectures

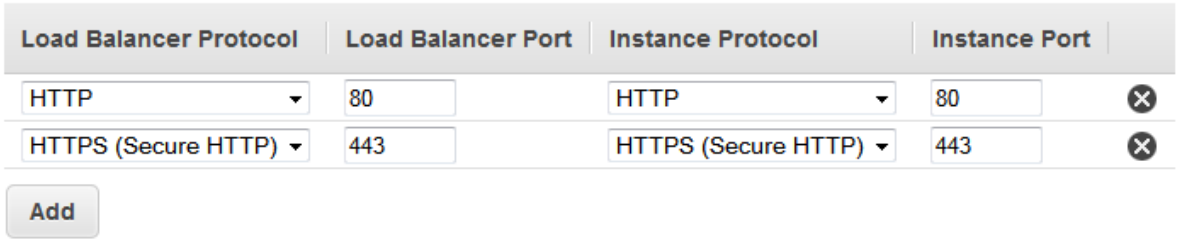
A company has a web application hosted on a fleet of EC2 instances located in two Availability Zones that are all placed behind an Application Load Balancer. As a Solutions Architect, you have to add a health check configuration to ensure your application is highly-available.

Which health checks will you implement?

- HTTP or HTTPS health check
- ICMP health check
- TCP health check
- FTP health check

Correct

A load balancer takes requests from clients and distributes them across the EC2 instances that are registered with the load balancer. You can create a load balancer that listens to both the HTTP (80) and HTTPS (443) ports. If you specify that the HTTPS listener sends requests to the instances on port 80, the load balancer terminates the requests, and communication from the load balancer to the instances is not encrypted. If the HTTPS listener sends requests to the instances on port 443, communication from the load balancer to the instances is not 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/

5. QUESTION

Category: CSAA – Design High-Performing Architectures

A fast food company is using AWS to host their online ordering system which uses an Auto Scaling group of EC2 instances deployed across multiple Availability Zones with an Application Load Balancer in front. To better handle the incoming traffic from various digital devices, you are planning to implement a new routing system where requests which have a URL of /api/android are forwarded to one specific target group named "Android-Target-Group". Conversely, requests which have a URL of /api/ios are forwarded to another separate target group named "iOS-Target-Group".

How can you implement this change in AWS?

- Use path conditions to define rules that forward requests to different target groups based on the URL in the request.
- Replace your ALB with a Classic Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request.
- Use host conditions to define rules that forward requests to different target groups based on the host name in the host header. This enables you to support multiple domains using a single load balancer.
- Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request.

Correct

You can use path conditions to define rules that forward requests to different target groups based on the URL in the request (also known as *path-based routing*). This type of routing is the most appropriate solution for this scenario hence, ****using path conditions to define rules that forward requests to different target groups based on the URL in the request**** is the correct answer.

Each path condition has one path pattern. If the URL in a request matches the path pattern in a listener rule exactly, the request is routed using that rule.

A path pattern is case-sensitive, can be up to 128 characters in length, and can contain any of the following characters. You can include up to three wildcard characters.

- A–Z, a–z, 0–9
- _ - . $ / ~ " ' @ : +
- & (using &)
- * (matches 0 or more characters)
- ? (matches exactly 1 character)

Example path patterns

- `/img/*`
- `/js/*`

The option that says: ****Use host conditions to define rules that forward requests to different target groups based on the host name in the host header. This enables you to support multiple domains using a single load balancer**** is incorrect because host-based routing defines rules that forward requests to different target groups based on the host name in the host header instead of the URL, which is what is needed in this scenario.

The option that says: ****Replace your ALB with a Classic Load Balancer then use path conditions to define rules that forward requests to different target groups based on the URL in the request**** is incorrect because a Classic Load Balancer does not support path-based routing. You must use an Application Load Balancer.

The option that says: ****Replace your ALB with a Network Load Balancer then use host conditions to define rules that forward requests to different target groups based on the URL in the request**** is incorrect because a Network Load Balancer is used for applications that need extreme network performance and static IP. It also does not support path-based routing which is what is needed in this

scenario. Furthermore, the statement mentions host-based routing yet, the description is about path-based routing.

References:

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/introduction.html#application-load-balancer-benefits

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-listeners.html#path-conditions

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer:

https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

6. QUESTION

Category: CSAA – Design Resilient Architectures

A company hosted a movie streaming app in Amazon Web Services. The application is deployed to several EC2 instances on multiple availability zones.

Which of the following configurations allows the load balancer to distribute incoming requests evenly to all EC2 instances across multiple Availability Zones?

- Elastic Load Balancing request routing
- An Amazon Route 53 weighted routing policy
- An Amazon Route 53 latency routing policy
- Cross-zone load balancing

Correct

The right answer is to enable ***cross-zone load balancing.***

If the load balancer nodes for your Classic Load Balancer can distribute requests regardless of Availability Zone, this is known as *cross-zone load balancing*. With cross-zone load balancing enabled, your load balancer nodes distribute incoming requests evenly across the Availability Zones enabled for your load balancer. Otherwise, each load balancer node distributes requests only to instances in its Availability Zone.

For example, if you have 10 instances in Availability Zone us-west-2a and 2 instances in us-west-2b, the requests are distributed evenly across all 12 instances if cross-zone load balancing is enabled. Otherwise, the 2 instances in us-west-2b serve the same number of requests as the 10 instances in us-west-2a.

Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances. However, we still recommend that you maintain approximately equivalent numbers of instances in each enabled Availability Zone for higher fault tolerance.

Reference:

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-disable-crosszone-lb.html

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

7. QUESTION

Category: CSAA – Design High-Performing Architectures

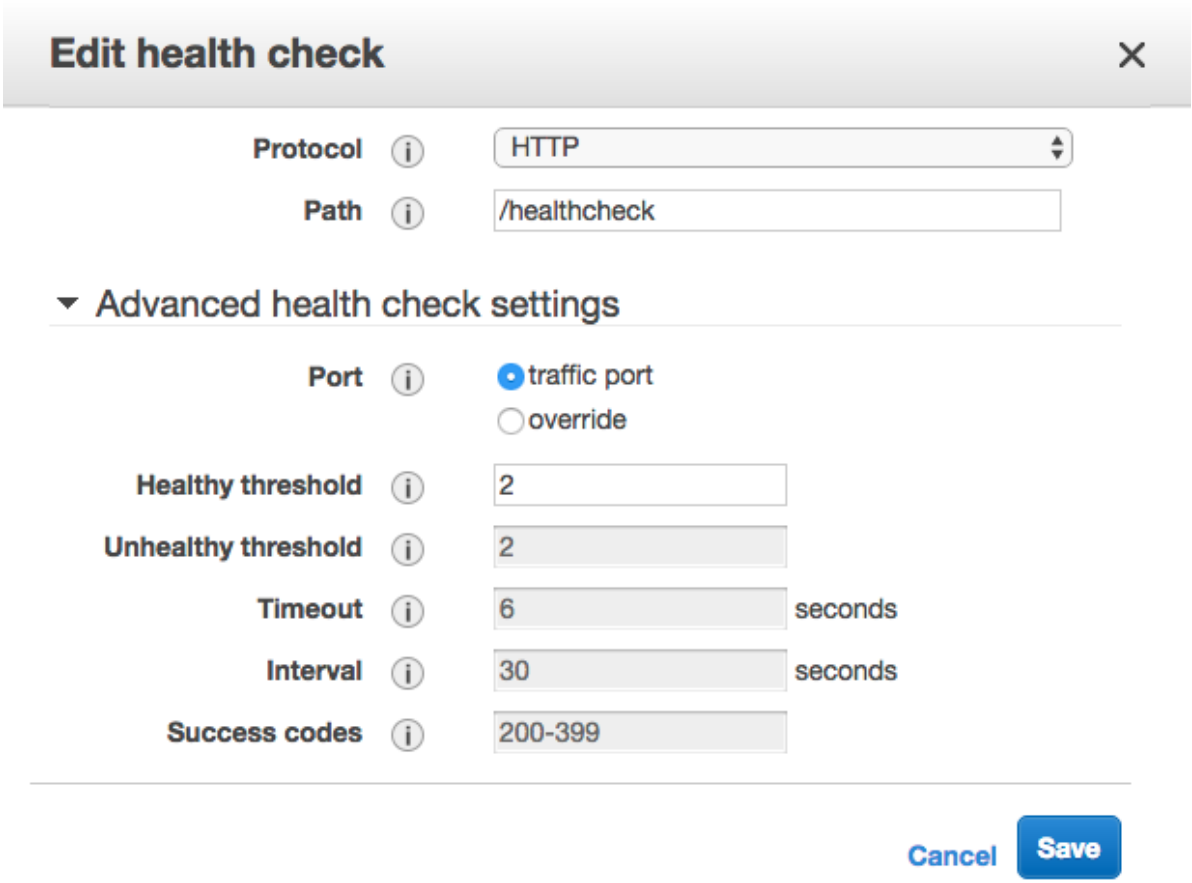
A company plans to design a highly available architecture in AWS. They have two target groups with three EC2 instances each, which are added to an Application Load Balancer. In the security group of the EC2 instance, you have verified that port 80 for HTTP is allowed. However, the instances are still showing out of service from the load balancer.

What could be the root cause of this issue?

- The instances are using the wrong AMI.
- The health check configuration is not properly defined.
- The wrong instance type was used for the EC2 instance.
- The wrong subnet was used in your VPC

Correct

Since the security group is properly configured, the issue may be caused by a wrong health check configuration in the Target Group.



Your **Application Load Balancer** periodically sends requests to its registered targets to test their status. These tests are called *health checks*. Each load balancer node routes requests only to the healthy targets in the enabled Availability Zones for the load balancer. Each load balancer node checks the health of each target, using the health check settings for the target group with which the target is registered. After your target is registered, it must pass one health check to be considered healthy. After each health check is completed, the load balancer node closes the connection that was established for the health check.

Reference:

http://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-healthchecks.html

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

ELB Health Checks vs Route 53 Health Checks For Target Health Monitoring:

https://tutorialsdojo.com/elb-health-checks-vs-route-53-health-checks-for-target-health-monitoring/

8. QUESTION

Category: CSAA – Design Secure Applications and Architectures

The social media company that you are working for needs to capture the detailed information of all HTTP requests that went through their public-facing application load balancer every five minutes. They want to use this data for analyzing traffic patterns and for troubleshooting their web applications in AWS.

Which of the following options meet the customer requirements?

- Add an Amazon CloudWatch Logs agent on the application load balancer.
- Enable access logs on the application load balancer.
- Enable AWS CloudTrail for their application load balancer.
- Enable Amazon CloudWatch metrics on the application load balancer.

Incorrect

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and troubleshoot issues.

Access logging is an optional feature of Elastic Load Balancing that is disabled by default. After you enable access logging for your load balancer, Elastic Load Balancing captures the logs and stores them in the Amazon S3 bucket that you specify as compressed files. You can disable access logging at any time.

Reference:

http://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html

AWS Elastic Load Balancing Overview:

Check out this AWS Elastic Load Balancing (ELB) Cheat Sheet:

https://tutorialsdojo.com/aws-elastic-load-balancing-elb/

Application Load Balancer vs Network Load Balancer vs Classic Load Balancer vs Gateway Load Balancer:

https://tutorialsdojo.com/application-load-balancer-vs-network-load-balancer-vs-classic-load-balancer/

1. QUESTION

Category: CSAA – Design Resilient Architectures

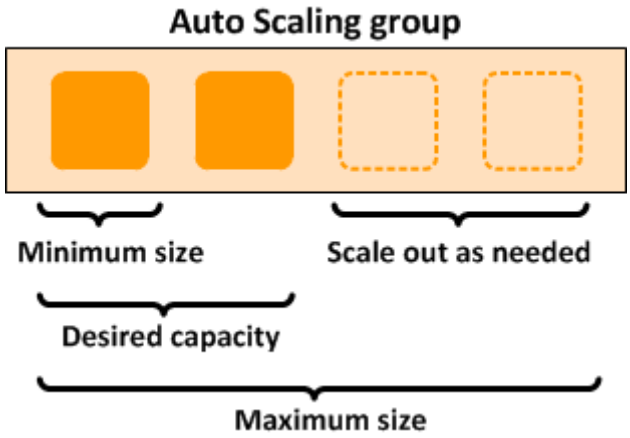
A company needs to deploy at least 2 EC2 instances to support the normal workloads of its application and automatically scale up to 6 EC2 instances to handle the peak load. The architecture must be highly available and fault-tolerant as it is processing mission-critical workloads.

As the Solutions Architect of the company, what should you do to meet the above requirement?

- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B.
- Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.

Incorrect

Amazon EC2 Auto Scaling helps ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can also specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.



To achieve highly available and fault-tolerant architecture for your applications, you must deploy all your instances in different Availability Zones. This will help you isolate your resources if an outage occurs. Take note that to achieve fault tolerance, you need to have redundant resources in place to avoid any system degradation in the event of a server fault or an Availability Zone outage. Having a fault-tolerant architecture entails an extra cost in running additional resources than what is usually needed. This is to ensure that the mission-critical workloads are processed.

Since the scenario requires at least 2 instances to handle regular traffic, you should have 2 instances running all the time even if an AZ outage occurred. You can use an Auto Scaling Group to automatically scale your compute resources across two or more Availability Zones. You have to specify the minimum capacity to 4 instances and the maximum capacity to 6 instances. If each AZ has 2 instances running, even if an AZ fails, your system will still run a minimum of 2 instances.

Hence, the correct answer in this scenario is: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 4 and the maximum capacity to 6. Deploy 2 instances in Availability Zone A and another 2 instances in Availability Zone B.****

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Deploy 4 instances in Availability Zone A**** is incorrect because the instances are only deployed in a single Availability Zone. It cannot protect your applications and data from datacenter or AZ failures.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 6. Use 2 Availability Zones and deploy 1 instance for each AZ**** is incorrect. It is required to have 2 instances running all the time. If an AZ outage happened, ASG will launch a new instance on the unaffected AZ. This provisioning does not happen instantly, which means that for a certain period of time, there will only be 1 running instance left.

The option that says: ***Create an Auto Scaling group of EC2 instances and set the minimum capacity to 2 and the maximum capacity to 4. Deploy 2 instances in Availability Zone A and 2 instances in Availability Zone B**** is incorrect. Although this fulfills the requirement of at least 2 EC2 instances and high availability, the maximum capacity setting is wrong. It should be set to 6 to properly handle the peak load. If an AZ outage occurs and the system is at its peak load, the number of running instances in this setup will only be 4 instead of 6 and this will affect the performance of your application.

References:

https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

https://docs.aws.amazon.com/documentdb/latest/developerguide/regions-and-azs.html

Check out this AWS Auto Scaling Cheat Sheet:

https://tutorialsdojo.com/aws-auto-scaling/

2. QUESTION

Category: CSAA – Design High-Performing Architectures

A company deployed a high-performance computing (HPC) cluster that spans multiple EC2 instances across multiple Availability Zones and processes various wind simulation models. Currently, the Solutions Architect is experiencing a slowdown in their applications and upon further investigation, it was discovered that it was due to latency issues.

Which is the MOST suitable solution that the Solutions Architect should implement to provide low-latency network performance necessary for tightly-coupled node-to-node communication of the HPC cluster?

- Use EC2 Dedicated Instances.
- Set up AWS Direct Connect connections across multiple Availability Zones for increased bandwidth throughput and more consistent network experience.
- Set up a spread placement group across multiple Availability Zones in multiple AWS Regions.
- Set up a cluster placement group within a single Availability Zone in the same AWS Region.

Incorrect

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

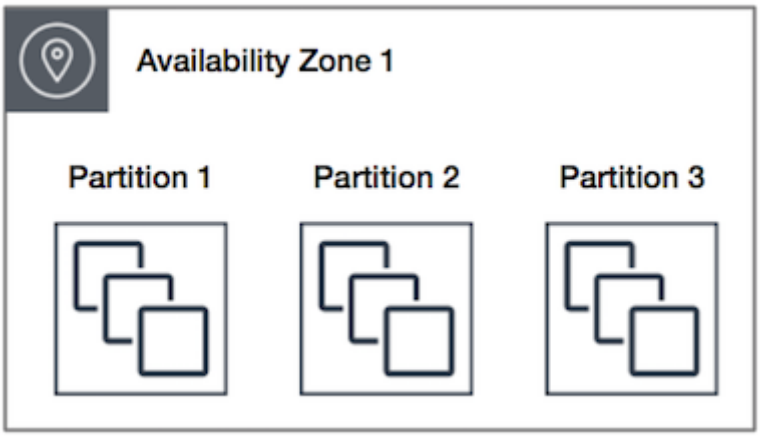
Cluster* – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.

Partition* – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.

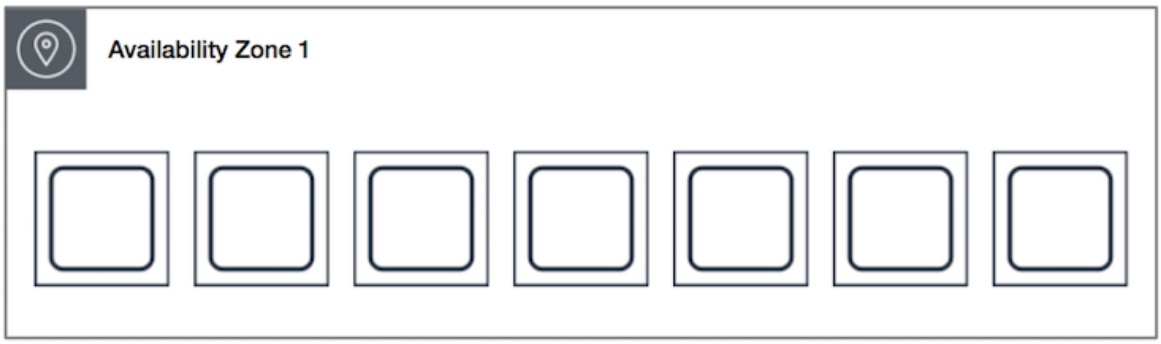
Spread – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group. To provide the lowest latency and the highest packet-per-second network performance for your placement group, choose an instance type that supports enhanced networking.

Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.



Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same racks. Spread placement groups provide access to distinct racks, and are therefore suitable for mixing instance types or launching instances over time. A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group.



Hence, the correct answer is: ***Set up a cluster placement group within a single Availability Zone in the same AWS Region.****

The option that says:***Set up a spread placement group across multiple Availability Zones in multiple AWS Regions**** is incorrect because although using a placement group is valid for this particular scenario, you can only set up a placement group in a **single** AWS Region only. A spread placement group can span multiple Availability Zones in the same Region.

The option that says: ***Set up AWS Direct Connect connections across multiple Availability Zones for increased bandwidth throughput and more consistent network experience**** is incorrect because this is primarily used for hybrid architectures. It bypasses the public Internet and establishes a secure, dedicated connection from your on-premises data center into AWS, and not used for having low latency within your AWS network.

The option that says: ***Use EC2 Dedicated Instances**** is incorrect because these are EC2 instances that run in a VPC on hardware that is dedicated to a single customer and are physically isolated at the host hardware level from instances that belong to other AWS accounts. It is not used for reducing latency.

References:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html

https://aws.amazon.com/hpc/

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

3. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

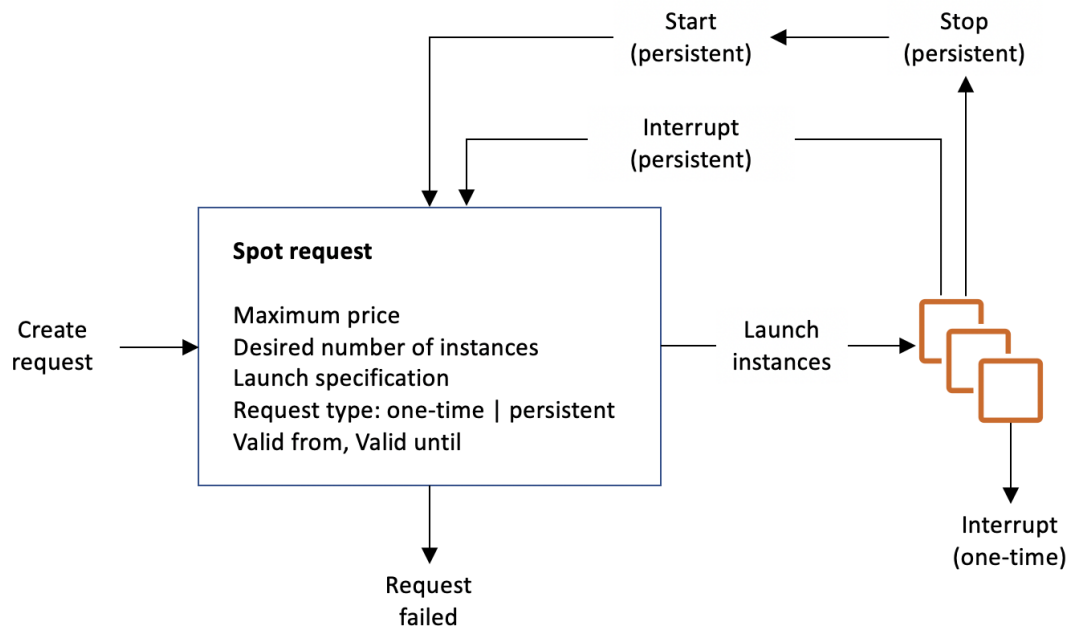
The media company that you are working for has a video transcoding application running on Amazon EC2. Each EC2 instance polls a queue to find out which video should be transcoded, and then runs a transcoding process. If this process is interrupted, the video will be transcoded by another instance based on the queuing system. This application has a large backlog of videos which need to be transcoded. Your manager would like to reduce this backlog by adding more EC2 instances, however, these instances are only needed until the backlog is reduced.

In this scenario, which type of Amazon EC2 instance is the most cost-effective type to use?

- Spot instances
- Reserved instances
- Dedicated instances
- On-demand instances

Correct

You require an instance that will be used not as a primary server but as a spare compute resource to augment the transcoding process of your application. These instances should also be terminated once the backlog has been significantly reduced. In addition, the scenario mentions that if the current process is interrupted, the video can be transcoded by another instance based on the queuing system. This means that the application can gracefully handle an unexpected termination of an EC2 instance, like in the event of a Spot instance termination when the Spot price is greater than your set maximum price. Hence, an Amazon EC2 Spot instance is the best and cost-effective option for this scenario.



Amazon EC2 Spot instances are **spare** compute capacity in the AWS cloud available to you at steep discounts compared to On-Demand prices. EC2 Spot enables you to optimize your costs on the AWS cloud and scale your application's throughput up to 10X for the same budget. By simply selecting Spot when launching EC2 instances, you can save up-to 90% on On-Demand prices. The only difference between On-Demand instances and Spot Instances is that Spot instances can be interrupted by EC2 with two minutes of notification when the EC2 needs the capacity back.

You can specify whether Amazon EC2 should hibernate, stop, or terminate Spot Instances when they are interrupted. You can choose the interruption behavior that meets your needs.

Take note that there is no *"bid price"* anymore for Spot EC2 instances **since March 2018**. You simply have to set your **maximum price** instead.

Reserved instances and ***Dedicated instances*** are incorrect as both do not act as spare compute capacity.

On-demand instances is a valid option but a Spot instance is much cheaper than On-Demand.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/spot-interruptions.html

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/how-spot-instances-work.html

https://aws.amazon.com/blogs/compute/new-amazon-ec2-spot-pricing

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

4. QUESTION

Category: CSAA – Design Resilient Architectures

A company has a cloud architecture that is composed of Linux and Windows EC2 instances that process high volumes of financial data 24 hours a day, 7 days a week. To ensure high availability of the systems, the Solutions Architect needs to create a solution that allows them to monitor the memory and disk utilization metrics of all the instances.

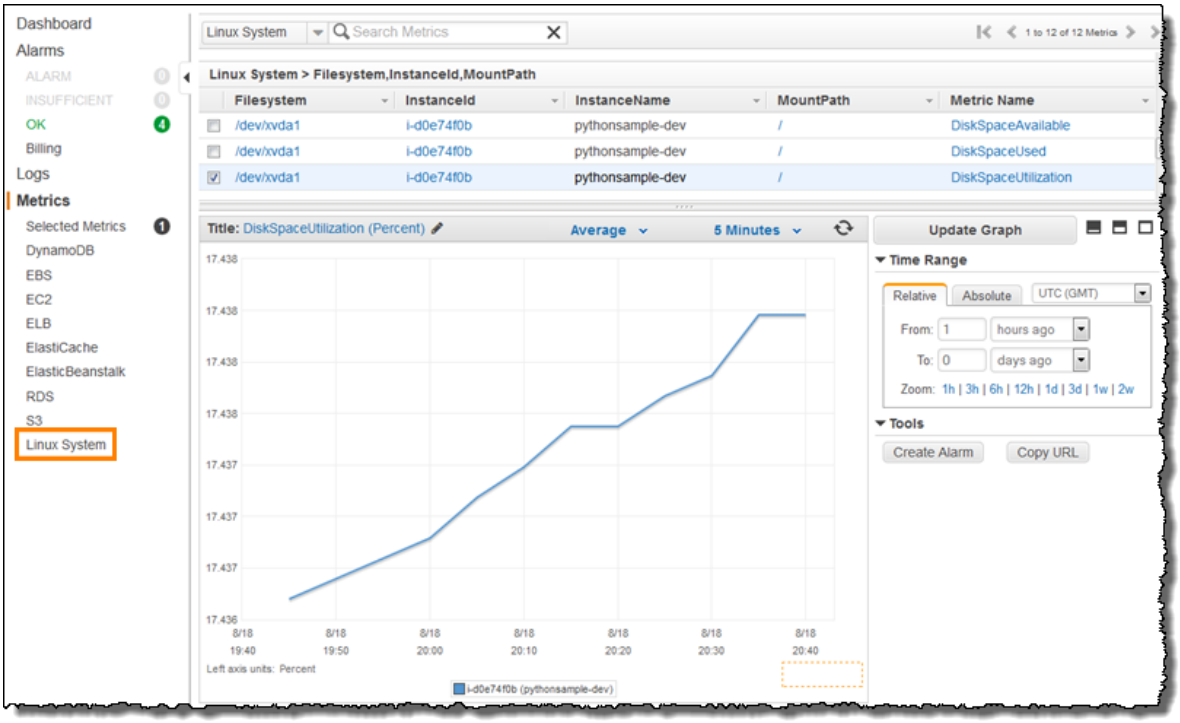
Which of the following is the most suitable monitoring solution to implement?

- Use Amazon Inspector and install the Inspector agent to all EC2 instances.
- Install the CloudWatch agent to all the EC2 instances that gather the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.
- Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard.
- Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances.

Correct

Amazon CloudWatch has available Amazon EC2 Metrics for you to use for monitoring CPU utilization, Network utilization, Disk performance, and Disk Reads/Writes. In case you need to monitor the below items, you need to prepare a custom metric using a Perl or other shell script, as there are no ready to use metrics for:

1. Memory utilization
2. Disk swap utilization
3. Disk space utilization
4. Page file utilization
5. Log collection



Take note that there is a multi-platform CloudWatch agent which can be installed on both Linux and Windows-based instances. You can use a single agent to collect both system metrics and log files from Amazon EC2 instances and on-premises servers. This agent supports both Windows Server and Linux and enables you to select the metrics to be collected, including sub-resource metrics such as per-CPU core. It is recommended that you use the new agent instead of the older monitoring scripts to collect metrics and logs.

Hence, the correct answer is: ***Install the CloudWatch agent to all the EC2 instances that gathers the memory and disk utilization data. View the custom metrics in the Amazon CloudWatch console.***

The option that says: ***Use the default CloudWatch configuration to EC2 instances where the memory and disk utilization metrics are already available. Install the AWS Systems Manager (SSM) Agent to all the EC2 instances*** is incorrect because, by default, CloudWatch does not automatically provide memory and disk utilization metrics of your instances. You have to set up custom CloudWatch metrics to monitor the memory, disk swap, disk space, and page file utilization of your instances.

The option that says: ***Enable the Enhanced Monitoring option in EC2 and install CloudWatch agent to all the EC2 instances to be able to view the memory and disk utilization in the CloudWatch dashboard*** is incorrect because Enhanced Monitoring is a feature of Amazon RDS. By default, Enhanced Monitoring metrics are stored for 30 days in the CloudWatch Logs.

The option that says: ***Use Amazon Inspector and install the Inspector agent to all EC2 instances*** is incorrect because Amazon Inspector is an automated security assessment service that helps you test the network accessibility of your Amazon EC2 instances and the security state of your applications running on the instances. It does not provide a custom metric to track the memory and disk utilization of each and every EC2 instance in your VPC.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring_ec2.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/mon-scripts.html#using_put_script

Check out this Amazon CloudWatch Cheat Sheet:

https://tutorialsdojo.com/amazon-cloudwatch/

CloudWatch Agent vs SSM Agent vs Custom Daemon Scripts:

https://tutorialsdojo.com/cloudwatch-agent-vs-ssm-agent-vs-custom-daemon-scripts/

Comparison of AWS Services Cheat Sheets:

https://tutorialsdojo.com/comparison-of-aws-services/

5. QUESTION

Category: CSAA – Design Resilient Architectures

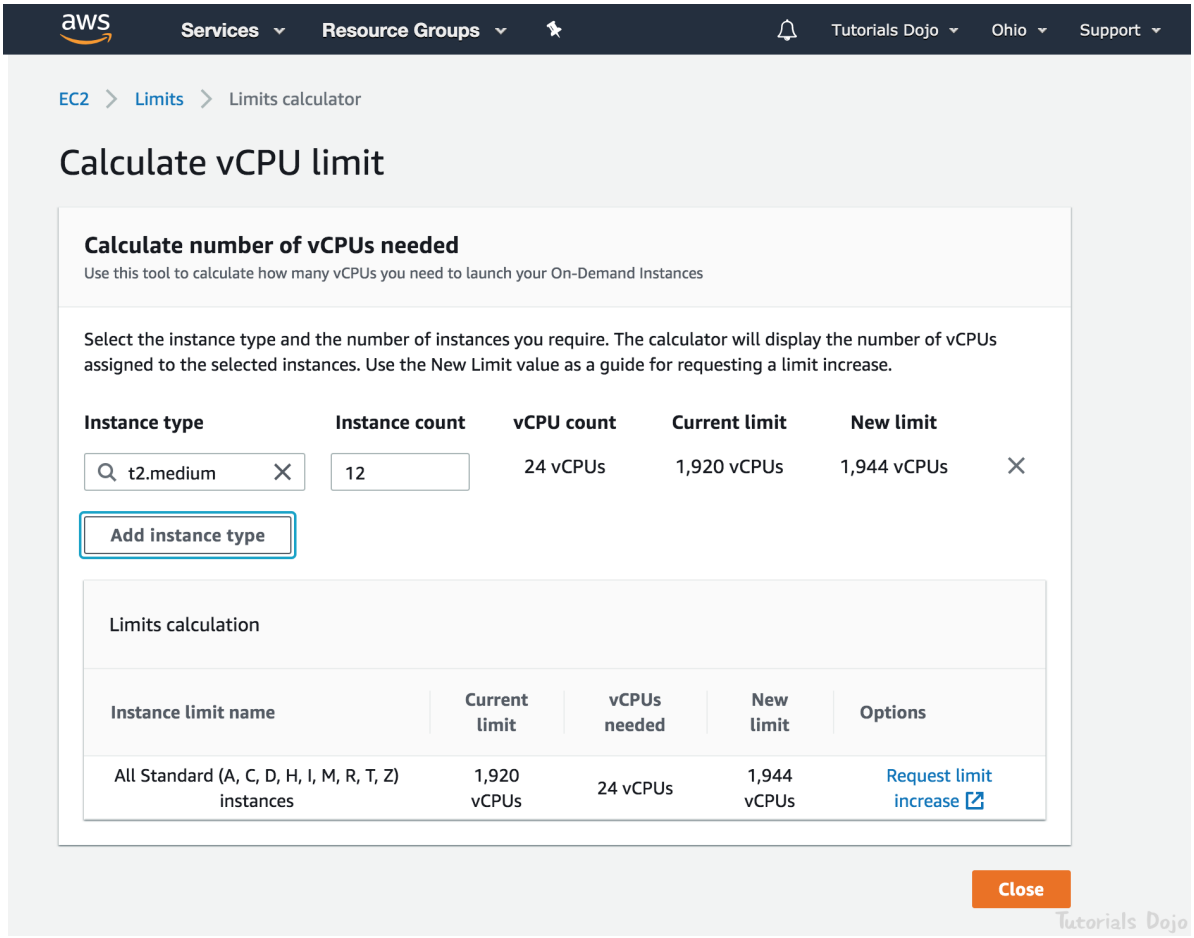
You are automating the creation of EC2 instances in your VPC. Hence, you wrote a python script to trigger the Amazon EC2 API to request 50 EC2 instances in a single Availability Zone. However, you noticed that after 20 successful requests, subsequent requests failed.

What could be a reason for this issue and how would you resolve it?

- There was an issue with the Amazon EC2 API. Just resend the requests and these will be provisioned successfully.
- There is a vCPU-based On-Demand Instance limit per region which is why subsequent requests failed. Just submit the limit increase form to AWS and retry the failed requests once approved.
- By default, AWS allows you to provision a maximum of 20 instances per region. Select a different region and retry the failed request.
- By default, AWS allows you to provision a maximum of 20 instances per Availability Zone. Select a different Availability Zone and retry the failed request.

Incorrect

You are limited to running On-Demand Instances per your vCPU-based On-Demand Instance limit, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region. New AWS accounts may start with limits that are lower than the limits described here.



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/

6. QUESTION

Category: CSAA – Design High-Performing Architectures

An organization needs to provision a new Amazon EC2 instance with a persistent block storage volume to migrate data from its on-premises network to AWS. The required maximum performance for the storage volume is 64,000 IOPS.

In this scenario, which of the following can be used to fulfill this requirement?

- Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance.
- Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.
- Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O.
- Launch any type of Amazon EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.

Incorrect

An **Amazon EBS volume** is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible.

The **AWS Nitro System** is the underlying platform for the latest generation of EC2 instances that enables AWS to innovate faster, further reduce the cost of the customers, and deliver added benefits like increased security and new instance types.

	Solid-state drives (SSD)		
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD	
		io2	io1
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	• Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle	
Amazon EBS Multi-attach	Not supported	Not Supported	Supported
API name	gp2	io2	io1
Volume size	1 GiB - 16 TiB	4 GiB - 16 TiB	
Dominant performance attribute	IOPS	IOPS	
Max IOPS per volume	16,000 (16 KiB I/O) *	64,000 (16 KiB I/O) †	
Max throughput per volume	250 MiB/s *	1,000 MiB/s †	
Max IOPS per instance ††	160,000		
Max throughput per instance ††	4,750 MB/s		

Maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System provisioned with more than 32,000 IOPS.

Amazon EBS is a persistent block storage volume. It can persist independently from the life of an instance. Since the scenario requires you to have an EBS volume with up to 64,000 IOPS, you have to launch a Nitro-based EC2 instance.

Hence, the correct answer in this scenario is: ***Launch a Nitro-based EC2 instance and attach a Provisioned IOPS SSD EBS volume (io1) with 64,000 IOPS.***

The option that says: ***Directly attach multiple Instance Store volumes in an EC2 instance to deliver maximum IOPS performance*** is incorrect. Although an Instance Store is a block storage volume, it is not persistent and the data will be gone if the instance is restarted from the stopped state (*note that this is different from the OS-level reboot. In OS-level reboot, data still persists in the instance store*). **An instance store only provides temporary block-level storage for your instance. It means that the data in the instance store can be lost if the underlying disk drive fails, if the instance stops, and if the instance terminates.**

The option that says: ***Launch an Amazon EFS file system and mount it to a Nitro-based Amazon EC2 instance and set the performance mode to Max I/O*** is incorrect. Although Amazon EFS can provide over 64,000 IOPS, this solution uses a file system and not a block storage volume which is what is asked in the scenario.

The option that says: ***Launch an EC2 instance and attach an io1 EBS volume with 64,000 IOPS*** is incorrect. In order to achieve the 64,000 IOPS for a provisioned IOPS SSD, you must provision a Nitro-based EC2 instance. The maximum IOPS and throughput are guaranteed only on Instances built on the Nitro System provisioned with more than 32,000 IOPS. Other instances guarantee up to 32,000 IOPS only.

References:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html#EBSVolumeTypes_piops

https://aws.amazon.com/s3/storage-classes/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

Check out this Amazon EBS Cheat Sheet:

https://tutorialsdojo.com/amazon-ebs/

Amazon S3 vs EFS vs EBS Cheat Sheet:

https://tutorialsdojo.com/amazon-s3-vs-ebs-vs-efs/

7. QUESTION

Category: CSAA – Design Cost-Optimized Architectures

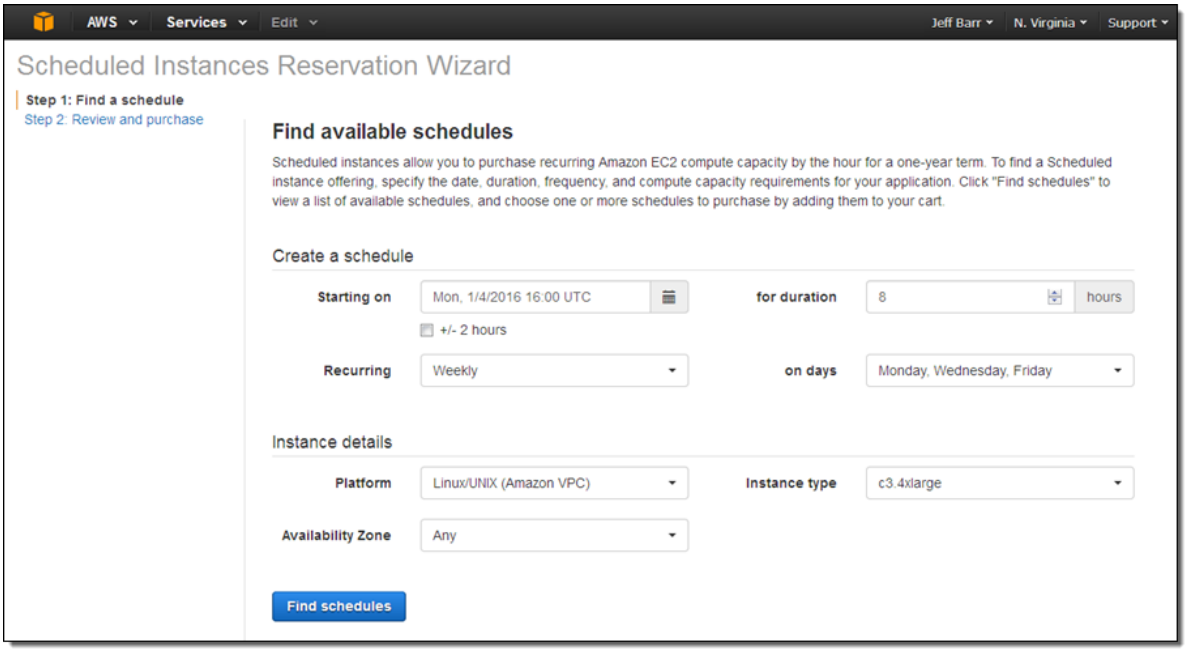
A multinational corporate and investment bank is regularly processing steady workloads of accruals, loan interests, and other critical financial calculations every night at 10 PM to 3 AM on their on-premises data center for their corporate clients. Once the process is done, the results are then uploaded to the Oracle General Ledger which means that the processing should not be delayed nor interrupted. The CTO has decided to move their IT infrastructure to AWS to save cost and to improve the scalability of their digital financial services.

As the Senior Solutions Architect, how can you implement a cost-effective architecture in AWS for their financial system?

- Use On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second.
- Use Spot EC2 Instances launched by a persistent Spot request, which can significantly lower your Amazon EC2 costs.
- Use Scheduled Reserved Instances, which provide compute capacity that is always available on the specified recurring schedule.
- Use Dedicated Hosts which provide a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.

Incorrect

Scheduled Reserved Instances (Scheduled Instances) enable you to purchase capacity reservations that recur on a daily, weekly, or monthly basis, with a specified start time and duration, for a one-year term. You reserve the capacity in advance, so that you know it is available when you need it. You pay for the time that the instances are scheduled, even if you do not use them.



Scheduled Instances are a good choice for workloads that do not run continuously, but do run on a regular schedule. For example, you can use Scheduled Instances for an application that runs during business hours or for batch processing that runs at the end of the week.

Hence, the correct answer is to ***use Scheduled Reserved Instances, which provide compute capacity that is always available on the specified recurring schedule***.

Using On-Demand EC2 instances which allows you to pay for the instances that you launch and use by the second is incorrect because although an On-Demand instance is stable and suitable for processing critical data, it costs more than any other option. Moreover, the critical financial calculations are only done every night from 10 PM to 3 AM only and not 24/7. This means that your compute capacity will not be utilized for a total of 19 hours every single day.

Using Spot EC2 Instances launched by a persistent Spot request, which can significantly lower your Amazon EC2 costs is incorrect because although this is the most cost-effective solution, this type is not suitable for processing critical financial data since a Spot Instance has a risk of being interrupted.

Using Dedicated Hosts which provide a physical host that is fully dedicated to running your instances, and bringing your existing per-socket, per-core, or per-VM software licenses to reduce costs is incorrect because the use of a fully dedicated physical host is not warranted in this scenario. Moreover, this will be underutilized since you only run the process for 5 hours (from 10 PM to 3 AM only), wasting 19 hours of compute capacity every single day.

References:

https://aws.amazon.com/blogs/aws/new-scheduled-reserved-instances/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-scheduled-instances.html

Check out this Amazon EC2 Cheat Sheet:

https://tutorialsdojo.com/amazon-elastic-compute-cloud-amazon-ec2/

8. QUESTION

Category: CSAA – Design Resilient Architectures

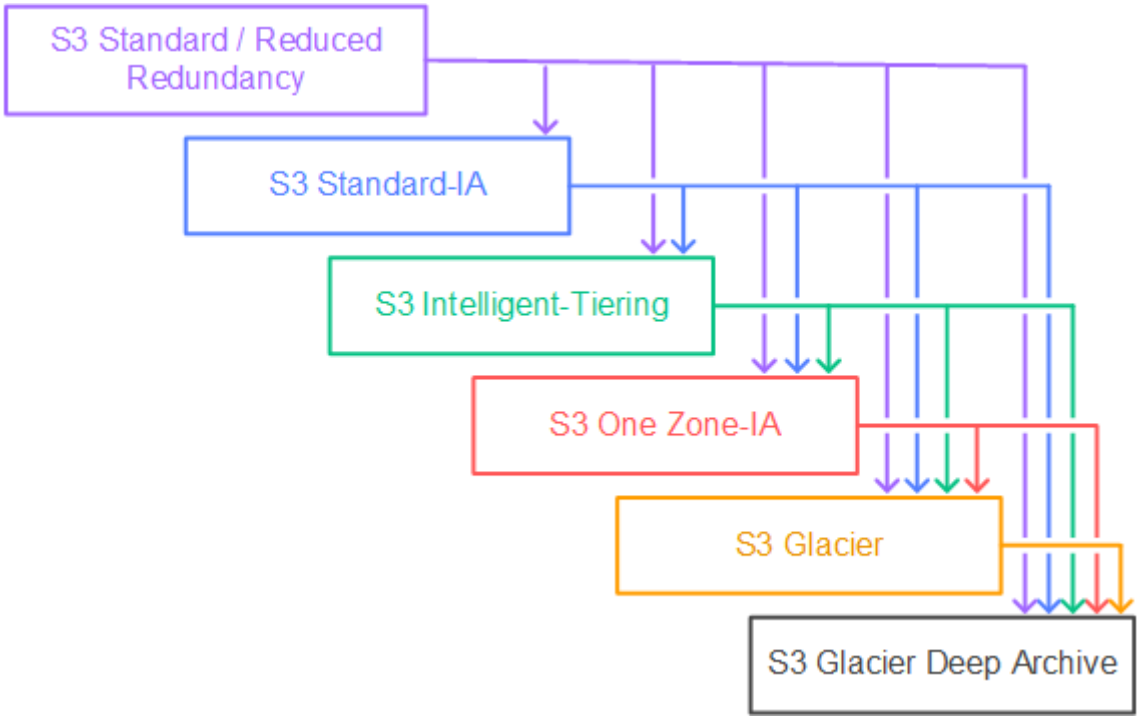
A company plans to host a web application in an Auto Scaling group of Amazon EC2 instances. The application will be used globally by users to upload and store several types of files. Based on user trends, files that are older than 2 years must be stored in a different storage class. The Solutions Architect of the company needs to create a cost-effective and scalable solution to store the old files yet still provide durability and high availability.

Which of the following approach can be used to fulfill this requirement? (Select TWO.)

- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.
- Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.
- Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years.
- Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years.
- Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.

Incorrect

Amazon S3 stores data as objects within buckets. An object is a file and any optional metadata that describes the file. To store a file in Amazon S3, you upload it to a bucket. When you upload a file as an object, you can set permissions on the object and any metadata. Buckets are containers for objects. You can have one or more buckets. You can control access for each bucket, deciding who can create, delete, and list objects in it. You can also choose the geographical region where Amazon S3 will store the bucket and its contents and view access logs for the bucket and its objects.



To move a file to a different storage class, you can use Amazon S3 or Amazon EFS. Both services have lifecycle configurations. Take note that Amazon EFS can only transition a file to the IA storage class after 90 days. Since you need to move the files that are older than 2 years to a more cost-effective and scalable solution, you should use the Amazon S3 lifecycle configuration. With S3 lifecycle rules, you can transition files to S3 Standard IA or S3 Glacier. Using S3 Glacier expedited retrieval, you can quickly access your files within 1-5 minutes.

Hence, the correct answers are:

– Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Glacier after 2 years.

– Use Amazon S3 and create a lifecycle policy that will move the objects to Amazon S3 Standard-IA after 2 years.

The option that says: ***Use Amazon EFS and create a lifecycle policy that will move the objects to Amazon EFS-IA after 2 years*** is incorrect because the maximum days for the EFS lifecycle policy is only 90 days. The requirement is to move the files that are older than 2 years or 730 days.

The option that says: ***Use Amazon EBS volumes to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because Amazon EBS costs more and is not as scalable as Amazon S3. It has some limitations when accessed by multiple EC2 instances. There are also huge costs involved in using the multi-attach feature on a Provisioned IOPS EBS volume to allow multiple EC2 instances to access the volume.

The option that says: ***Use a RAID 0 storage configuration that stripes multiple Amazon EBS volumes together to store the files. Configure the Amazon Data Lifecycle Manager (DLM) to schedule snapshots of the volumes after 2 years*** is incorrect because RAID (Redundant Array of Independent Disks) is just a data storage virtualization technology that combines multiple storage devices to achieve higher performance or data durability. RAID 0 can stripe multiple volumes together for greater I/O performance than you can achieve with a single volume. On the other hand, RAID 1 can mirror two volumes together to achieve on-instance redundancy.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

https://docs.aws.amazon.com/efs/latest/ug/lifecycle-management-efs.html

https://aws.amazon.com/s3/faqs/

Check out this Amazon S3 Cheat Sheet:

https://tutorialsdojo.com/amazon-s3/