Question 1Skipped

A media company is evaluating the possibility of moving its IT infrastructure to the AWS Cloud. The company needs at least 10 terabytes of storage with the maximum possible I/O performance for processing certain files which are mostly large videos. The company also needs close to 450 terabytes of very durable storage for storing media content and almost double of it, i.e. 900 terabytes for archival of legacy data.

As a Solutions Architect, which set of services will you recommend to meet these requirements?

Amazon S3 standard storage for maximum performance, Amazon S3 Intelligent-Tiering for intelligent, durable storage, and Amazon S3 Glacier Deep Archive for archival storage

Amazon EC2 instance store for maximum performance, AWS Storage Gateway for on-premises durable data access and Amazon S3 Glacier Deep Archive for archival storage

Correct answer

Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage

Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage

Overall explanation

Correct option:

Amazon EC2 instance store for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for the temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers.

You can specify instance store volumes for an instance only when you launch it. You can't detach an instance store volume from one instance and attach it to a different instance.

Some instance types use NVMe or SATA-based solid-state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures.

Amazon S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, Amazon S3 Standard is appropriate for a wide variety of use cases, including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and big data analytics.

Amazon S3 Glacier is a secure, durable, and low-cost storage class for data archiving. You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions. To keep costs low yet suitable for varying needs, Amazon S3 Glacier

provides three retrieval options that range from a few minutes to hours. You can upload objects directly to Amazon S3 Glacier, or use S3 Lifecycle policies to transfer data between any of the Amazon S3 Storage Classes for active data (S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, and S3 One Zone-IA) and S3 Glacier.

Incorrect options:

Amazon S3 standard storage for maximum performance, Amazon S3 Intelligent-Tiering for intelligent, durable storage, and Amazon S3 Glacier Deep Archive for archival storage - Amazon EC2 instance store volumes provide the best I/O performance for low latency requirement, as in the current use case. The Amazon S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead.

Amazon S3 Glacier Deep Archive is Amazon S3's lowest-cost storage class and supports long-term retention and digital preservation for data that may be accessed once or twice a year. It is designed for customers — particularly those in highly-regulated industries, such as the Financial Services, Healthcare, and Public Sectors — that retain data sets for 7-10 years or longer to meet regulatory compliance requirements.

Amazon EBS for maximum performance, Amazon S3 for durable data storage, and Amazon S3 Glacier for archival storage - Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with EC2 instances. Amazon EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. For high I/O performance, instance store volumes are a better option.

Amazon EC2 instance store for maximum performance, AWS Storage Gateway for on-premises durable data access and Amazon S3 Glacier Deep Archive for archival storage - AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. AWS Storage Gateway will be the right answer if the customer wanted to retain the on-premises data storage and just move the applications to AWS Cloud. In the absence of such requirements, instance store is a better option for high performance and Amazon S3 for durable storage.

Reference:

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

Domain

Design High-Performing Architectures

Question 2Skipped

A media company wants to get out of the business of owning and maintaining its own IT infrastructure. As part of this digital transformation, the media company wants to archive about 5 petabytes of data in its on-premises data center to durable long term storage.

As a solutions architect, what is your recommendation to migrate this data in the MOST cost-optimal way?

Transfer the on-premises data into multiple AWS Snowball Edge Storage Optimized devices. Copy the AWS Snowball Edge data into Amazon S3 Glacier

Setup AWS Site-to-Site VPN connection between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier

Correct answer

Transfer the on-premises data into multiple AWS Snowball Edge Storage Optimized devices. Copy the AWS Snowball Edge data into Amazon S3 and create a lifecycle policy to transition the data into Amazon S3 Glacier

Setup AWS direct connect between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier

Overall explanation

Correct option:

Transfer the on-premises data into multiple AWS Snowball Edge Storage Optimized devices. Copy the AWS Snowball Edge data into Amazon S3 and create a lifecycle policy to transition the data into Amazon S3 Glacier

AWS Snowball Edge Storage Optimized is the optimal choice if you need to securely and quickly transfer dozens of terabytes to petabytes of data to AWS. It provides up to 80 TB of usable HDD storage, 40 vCPUs, 1 TB of SATA SSD storage, and up to 40 Gb network connectivity to address large scale data transfer and pre-processing use cases. The data stored on AWS Snowball Edge device can be copied into Amazon S3 bucket and later transitioned into Amazon S3 Glacier via a lifecycle policy. You can't directly copy data from AWS Snowball Edge devices into Amazon S3 Glacier.

Incorrect options:

Transfer the on-premises data into multiple AWS Snowball Edge Storage Optimized devices. Copy the AWS Snowball Edge data into Amazon S3 Glacier - As mentioned earlier, you can't directly copy data from AWS Snowball Edge devices into Amazon S3 Glacier. Hence, this option is incorrect.

Setup AWS direct connect between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier - AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. Direct Connect involves significant monetary investment and takes more than a month to set up, therefore it's not the correct fit for this use-case where just a one-time data transfer has to be done.

Setup AWS Site-to-Site VPN connection between the on-premises data center and AWS Cloud. Use this connection to transfer the data into Amazon S3 Glacier - AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon Virtual Private Cloud (Amazon VPC). VPN Connections are a good solution if you have an immediate need, and have low to modest bandwidth requirements. Because of the high data volume for the given use-case, Site-to-Site VPN is not the correct choice.

Reference:

https://aws.amazon.com/snowball/

Domain

Design Cost-Optimized Architectures

Question 3Skipped

Your company is evolving towards a microservice approach for their website. The company plans to expose the website from the same load balancer, linked to different target groups with different URLs, that are similar to these - checkout.mycorp.com, www.mycorp.com, mycorp.com/profile, and mycorp.com/search.

As a Solutions Architect, which Load Balancer type do you recommend to achieve this routing feature with MINIMUM configuration and development effort?

Correct answer

Create an Application Load Balancer

Create a Network Load Balancer

Create an NGINX based load balancer on an Amazon EC2 instance to have advanced routing capabilities

Create a Classic Load Balancer

Overall explanation

Correct option:

Create an Application Load Balancer

Application Load Balancer can automatically distribute incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

If your application is composed of several individual services, an Application Load Balancer can route a request to a service based on the content of the request.

Here are the different types -

Host-based Routing: You can route a client request based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer. You can use host conditions to define rules that route requests based on the hostname in the host header (also known as host-based routing). This enables you to support multiple domains using a single load balancer. Example hostnames: example.com test.example.com *.example.com The rule *.example.com matches test.example.com but doesn't match example.com.

Path-based Routing: You can route a client request based on the URL path of the HTTP header. You can use path conditions to define rules that route requests based on the URL in the request (also known as path-based routing). Example path patterns: /img/* /img//pics *The path pattern is used to route requests but does not alter them. For example, if a rule has a path pattern of /img/*, the rule would forward a request for /img/picture.jpg to the specified target group as a request for /img/picture.jpg. The path pattern is applied only to the path of the URL, not to its query parameters.

HTTP header-based routing: You can route a client request based on the value of any standard or custom HTTP header.

HTTP method-based routing: You can route a client request based on any standard or custom HTTP method.

Query string parameter-based routing: You can route a client request based on query string or query parameters.

Source IP address CIDR-based routing: You can route a client request based on source IP address CIDR from where the request originates.

Path based routing and host based routing are only available for the Application Load Balancer (ALB). Therefore this is the correct option for the given use-case.

Incorrect options:

Create an NGINX based load balancer on an Amazon EC2 instance to have advanced routing capabilities - Although it is technically possible to set up NGINX based load balancer, however, this option involves a lot of configuration effort, so this option is ruled out for the given use-case. So, deploying an NGINX load balancer on Amazon EC2 would work but would suffer management and scaling issues.

Create a Network Load Balancer - Network Load Balancer is best suited for use-cases involving low latency and high throughput workloads that involve scaling to millions of requests per second. Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, microservices, and containers – within Amazon Virtual Private Cloud (Amazon VPC) based on IP protocol data.

Create a Classic Load Balancer - Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

As mentioned in the description above, these two options are incorrect for the given use-case.

References:

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

https://aws.amazon.com/blogs/aws/new-host-based-routing-support-for-aws-application-load-balancers/

Domain

Design High-Performing Architectures

Question 4Skipped

A DevOps engineer at an IT company was recently added to the admin group of the company's AWS account. The AdministratorAccess managed policy is attached to this group.

Can you identify the AWS tasks that the DevOps engineer CANNOT perform even though he has full Administrator privileges (Select two)?

Delete the IAM user for his manager

Correct selection

Configure an Amazon S3 bucket to enable AWS Multi-Factor Authentication (AWS MFA) delete

Delete an Amazon S3 bucket from the production environment

Correct selection

Close the company's AWS account

Change the password for his own IAM user account

Overall explanation

Correct options:

Configure an Amazon S3 bucket to enable AWS Multi-Factor Authentication (AWS MFA) delete

Close the company's AWS account

An IAM user with full administrator access can perform almost all AWS tasks except a few tasks designated only for the root account user. Some of the AWS tasks that only a root account user can do are as follows: change account name or root password or root email address, change AWS support plan, close AWS account, enable AWS Multi-Factor Authentication (AWS MFA) on S3 bucket delete, create Cloudfront key pair, register for GovCloud. Even though the DevOps engineer is part of the admin group, he cannot configure an Amazon S3 bucket to enable AWS MFA delete or close the company's AWS account.

Incorrect Options:

Delete the IAM user for his manager

Delete an Amazon S3 bucket from the production environment

[@@-E

The DevOps engineer is part of the admin group, so he can delete any IAM user, delete the Amazon S3 bucket, and change the password for his own IAM user account.

For the complete list of AWS tasks that require AWS account root user credentials, please review this reference link:

Reference:

https://docs.aws.amazon.com/general/latest/gr/aws_tasks-that-require-root.html

Domain

Design Secure Architectures

Question 5Skipped

A company hires experienced specialists to analyze the customer service calls attended by its call center representatives. Now, the company wants to move to AWS Cloud and is looking at an

automated solution to analyze customer service calls for sentiment analysis via ad-hoc SQL queries.

As a Solutions Architect, which of the following solutions would you recommend?

Correct answer

Use Amazon Transcribe to convert audio files to text and Amazon Athena to perform SQL based analysis to understand the underlying customer sentiments

Use Amazon Transcribe to convert audio files to text and Amazon Quicksight to perform SQL based analysis on these text files to understand the underlying patterns. Visualize and display them onto user Dashboards for reporting purposes

Use Amazon Kinesis Data Streams to read the audio files and Amazon Alexa to convert them into text. Amazon Kinesis Data Analytics can be used to analyze these files and Amazon Quicksight can be used to visualize and display the output

Use Amazon Kinesis Data Streams to read the audio files and machine learning (ML) algorithms to convert the audio files into text and run customer sentiment analysis

Overall explanation

Correct option:

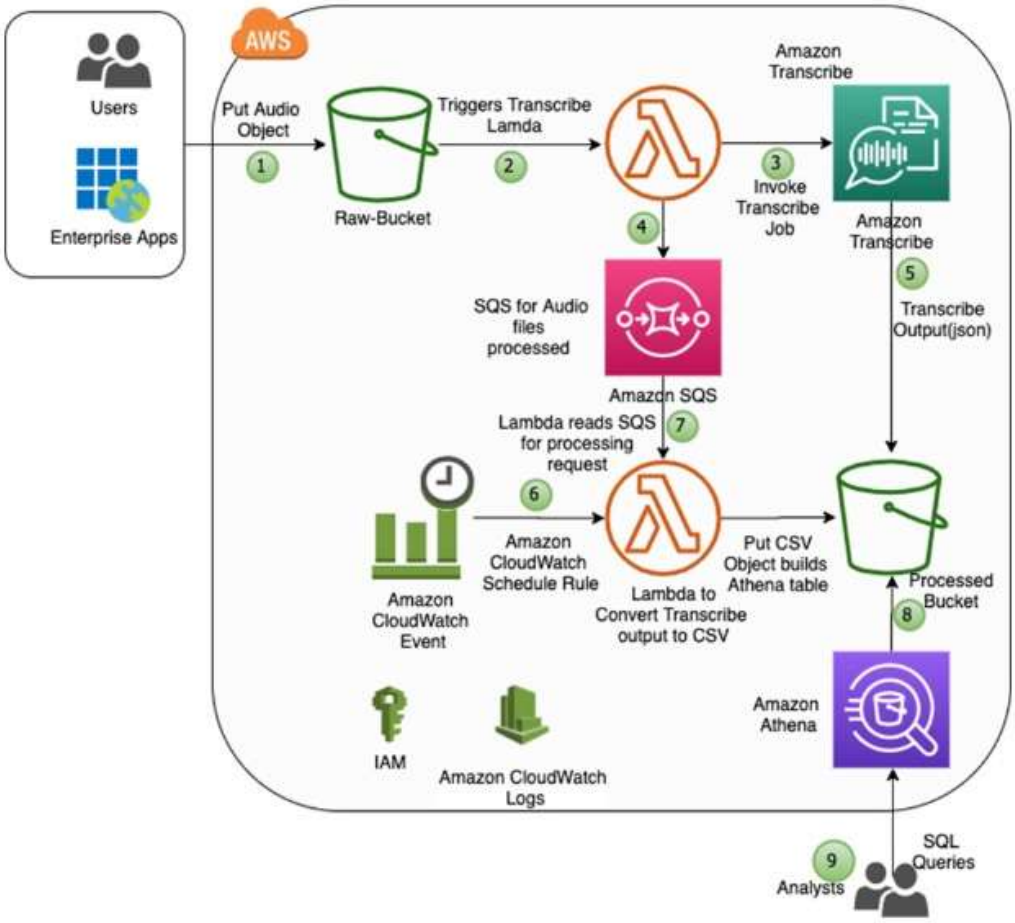
Use Amazon Transcribe to convert audio files to text and Amazon Athena to perform SQL based analysis to understand the underlying customer sentiments

Amazon Transcribe is an automatic speech recognition (ASR) service that makes it easy to convert audio to text. One key feature of the service is called speaker identification, which you can use to label each individual speaker when transcribing multi-speaker audio files. You can specify Amazon Transcribe to identify 2–10 speakers in the audio clip.

Amazon Athena is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to manage, and you pay only for the queries that you run. To leverage Athena, you can simply point to your data in Amazon S3, define the schema, and start querying using standard SQL. Most results are delivered within seconds.

Analyzing multi-speaker audio files using Amazon Transcribe and Amazon Athena:

Diagram: Analyze Multi-Speaker Audio Files Using Amazon Transcribe and Amazon Athena



via - https://aws.amazon.com/blogs/machine-learning/automating-the-analysis-of-multi-speaker-audio-files-using-amazon-transcribe-and-amazon-athena

Incorrect options:

Use Amazon Kinesis Data Streams to read the audio files and machine learning (ML) algorithms to convert the audio files into text and run customer sentiment analysis - Amazon Kinesis can be used to stream real-time data for further analysis and storage. Kinesis Data Streams cannot read audio files. You will still need to use AWS Transcribe for ASR services.

Use Amazon Kinesis Data Streams to read the audio files and Amazon Alexa to convert them into text. Amazon Kinesis Data Analytics can be used to analyze these files and Amazon Quicksight can be used to visualize and display the output - Amazon Kinesis Data Streams cannot read audio files. Amazon Alexa cannot be used as an Automatic Speech Recognition (ASR) service, though Alexa internally uses ASR for its working.

Use Amazon Transcribe to convert audio files to text and Amazon Quicksight to perform SQL based analysis on these text files to understand the underlying patterns. Visualize and display them onto user Dashboards for reporting purposes - Amazon Quicksight is used for the visual representation of data through dashboards. However, it is not an SQL query based analysis tool like Amazon Athena. So, this option is incorrect.

References:

https://aws.amazon.com/blogs/machine-learning/automating-the-analysis-of-multi-speaker-audio-files-using-amazon-transcribe-and-amazon-athena

https://aws.amazon.com/athena

Domain

Design High-Performing Architectures

Question 6Skipped

An Internet-of-Things (IoT) company is looking for a database solution on AWS Cloud that has Auto Scaling capabilities and is highly available. The database should be able to handle any changes in data attributes over time, in case the company updates the data feed from its IoT devices. The database must provide the capability to output a continuous stream with details of any changes to the underlying data.

As a Solutions Architect, which database will you recommend?

Amazon Redshift

Amazon Aurora

Correct answer

Amazon DynamoDB

Amazon Relational Database Service (Amazon RDS)

Overall explanation

Correct option:

Amazon DynamoDB

Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-Region, multi-master, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second. DynamoDB is serverless with no servers to provision, patch, or manage and no software to install, maintain, or operate.

A Amazon DynamoDB stream is an ordered flow of information about changes to items in a DynamoDB table. When you enable a stream on a table, Amazon 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 attributes 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 horizontally scalable, has a DynamoDB streams capability and is multi-AZ by default. On top of it, we can adjust the RCU and WCU automatically using Auto Scaling. This is the right choice for current requirements.

Incorrect options:

Amazon Relational Database Service (Amazon RDS) - Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. Schema changes on relational databases are not straight forward and are hard to maintain if the schema requirements change often.

Amazon Aurora - Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud, that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 64TB per database instance. Aurora is not an in-memory database. Schema changes on relational databases are not straight forward and are hard to maintain if the schema requirements change often.

Amazon Redshift - Amazon Redshift is a fully-managed petabyte-scale cloud based data warehouse product designed for large scale data set storage and analysis. It is a powerful warehousing service from Amazon. The current requirement, however, is not looking for a warehousing solution and hence Redshift is not an option here.

References:

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

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

Domain

Design High-Performing Architectures

Question 7Skipped

A silicon valley based startup helps its users legally sign highly confidential contracts. To meet the compliance guidelines, the startup must ensure that the signed contracts are encrypted using the AES-256 algorithm via an encryption key that is generated as well as managed internally. The startup is now migrating to AWS Cloud and would like the data to be encrypted on AWS. The startup wants to continue using their existing encryption key generation as well as key management mechanism.

What do you recommend?

Client-Side Encryption

Correct answer

SSE-C

SSE-KMS

SSE-S3

Overall explanation

Correct option:

SSE-C

With Server-Side Encryption with Customer-Provided Keys (SSE-C), you manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption when you access your objects. With SSE-C, the startup can still generate and manage the encryption key but let AWS do the encryption. Therefore, this is the correct option.

Incorrect options:

SSE-KMS - AWS Key Management Service (AWS KMS) is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud. When you use server-side encryption with AWS KMS (SSE-KMS), you can specify a customer-managed CMK that you have already created. But, you never get to know the actual key here.

SSE-S3 - When you use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3), each object is encrypted with a unique key. However, this option does not provide the ability to audit trail the usage of the encryption keys.

Client-Side Encryption - 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 a AWS KMS key stored in AWS Key Management Service (AWS KMS), Use a master key you store within your application. Since the customer wants to use AWS provided facility, this is not an option.

References:

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

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

https://docs.aws.amazon.com/AmazonS3/latest/dev/serv-side-encryption.html

Domain

Design Secure Architectures

Question 8Skipped

You have built an application that is deployed with Elastic Load Balancing and an Auto Scaling Group. As a Solutions Architect, you have configured aggressive Amazon CloudWatch alarms, making your Auto Scaling Group (ASG) scale in and out very quickly, renewing your fleet of Amazon EC2 instances on a daily basis. A production bug appeared two days ago, but the team is unable to SSH into the instance to debug the issue, because the instance has already been terminated by the Auto Scaling Group. The log files are saved on the Amazon EC2 instance.

How will you resolve the issue and make sure it doesn't happen again?

Use AWS Lambda to regularly SSH into the Amazon EC2 instances and copy the log files to Amazon S3

Make a snapshot of the Amazon EC2 instance just before it gets terminated

Disable the Termination from the Auto Scaling Group any time a user reports an issue

Correct answer

Install an Amazon CloudWatch Logs agents on the Amazon EC2 instances to send logs to Amazon CloudWatch

Overall explanation

Correct option:

Install an Amazon CloudWatch Logs agents on the Amazon EC2 instances to send logs to Amazon CloudWatch

You can use the Amazon CloudWatch Logs agent installer on an existing Amazon EC2 instance to install and configure the Amazon CloudWatch Logs agent. After installation is complete, logs automatically flow from the instance to the log stream you create while installing the agent. The agent confirms that it has started and it stays running until you disable it.

Here, the natural and by far the easiest solution would be to use the Amazon CloudWatch Logs agents on the Amazon EC2 instances to automatically send log files into Amazon CloudWatch, so we can analyze them in the future easily should any problem arise.

To control whether an Auto Scaling group can terminate a particular instance when scaling in, use instance scale-in protection. You can enable the instance scale-in protection setting on an Auto Scaling group or on an individual Auto Scaling instance. When the Auto Scaling group launches an instance, it inherits the instance scale-in protection setting of the Auto Scaling group. You can change the instance scale-in protection setting for an Auto Scaling group or an Auto Scaling instance at any time.

Incorrect options:

Disable the Termination from the Auto Scaling Group any time a user reports an issue - Disabling the Termination from the Auto Scaling Group would prevent our Auto Scaling Group from being Elastic and impact our costs. Therefore this option is incorrect.

Make a snapshot of the Amazon EC2 instance just before it gets terminated - Making a snapshot of the Amazon EC2 instance before it gets terminated could work but it's tedious, not elastic and very expensive, since our interest is just the log files. Therefore this option is not the best fit for the given use-case.

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data.

Use AWS Lambda to regularly SSH into the Amazon EC2 instances and copy the log files to Amazon S3 - AWS Lambda lets you run code without provisioning or managing servers. It cannot be used for production-grade serverless log analytics. Using AWS Lambda would be extremely hard to use for this task. Therefore this option is not the best fit for the given use-case.

Reference:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html

Domain

Design High-Performing Architectures

Question 9Skipped

A solutions architect has been tasked to design a low-latency solution for a static, single-page application, accessed by users through a custom domain name. The solution must be serverless, provide in-transit data encryption and needs to be cost-effective.

Which AWS services can be combined to build the simplest possible solution for the company's requirement?

Host the application on AWS Fargate and front it with Elastic Load Balancing for an improved performance

Correct answer

Use Amazon S3 to host the static website and Amazon CloudFront to distribute the content for low latency access

Configure Amazon S3 to store the static data and use AWS Fargate for hosting the application

Host the application on Amazon EC2 instance with instance store volume for high performance and low latency access to users

Overall explanation

Correct option:

Use Amazon S3 to host the static website and Amazon CloudFront to distribute the content for low latency access

To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you must enable website hosting, set permissions, and create and add an index document. Depending on your website requirements, you can also configure redirects, web traffic logging, and a custom error document.

After you configure your bucket as a static website, you can access the bucket through the AWS Region-specific Amazon S3 website endpoints for your bucket. Website endpoints are different from the endpoints where you send REST API requests. Amazon S3 doesn't support HTTPS access for website endpoints. If you want to use HTTPS, you can use CloudFront to serve a static website hosted on Amazon S3.

You can use Amazon CloudFront to improve the performance of your website. CloudFront makes your website files (such as HTML, images, and video) available from data centers around the world (called edge locations). When a visitor requests a file from your website, Amazon CloudFront automatically redirects the request to a copy of the file at the nearest edge location. This results in faster download times than if the visitor had requested the content from a data center that is located farther away.

Amazon CloudFront caches content at edge locations for a period of time that you specify. If a visitor requests content that has been cached for longer than the expiration date, Amazon

CloudFront checks the origin server to see if a newer version of the content is available. If a newer version is available, Amazon CloudFront copies the new version to the edge location. Changes that you make to the original content are replicated to edge locations as visitors request the content.

Incorrect options:

Host the application on Amazon EC2 instance with instance store volume for high performance and low latency access to users - Since the use case speaks about a serverless solution, Amazon EC2 cannot be the answer, since Amazon EC2 is not serverless.

Host the application on AWS Fargate and front it with Elastic Load Balancing for an improved performance - AWS Fargate is a serverless compute engine for containers that works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS). Elastic Load Balancing can spread the incoming requests across a fleet of Amazon EC2 instances. This added complexity is not needed since we are looking at a static single-page webpage.

Configure Amazon S3 to store the static data and use AWS Fargate for hosting the application - AWS Fargate is overkill for hosting a static single-page webpage.

Reference:

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

Domain

Design High-Performing Architectures

Question 10Skipped

A big data analytics company is using Amazon Kinesis Data Streams (KDS) to process IoT data from the field devices of an agricultural sciences company. Multiple consumer applications are using the incoming data streams and the engineers have noticed a performance lag for the data delivery speed between producers and consumers of the data streams.

As a solutions architect, which of the following would you recommend for improving the performance for the given use-case?

Swap out Amazon Kinesis Data Streams with Amazon Kinesis Data Firehose

Correct answer

Use Enhanced Fanout feature of Amazon Kinesis Data Streams

Swap out Amazon Kinesis Data Streams with Amazon SQS FIFO queues

Swap out Amazon Kinesis Data Streams with Amazon SQS Standard queues

Overall explanation

Correct option:

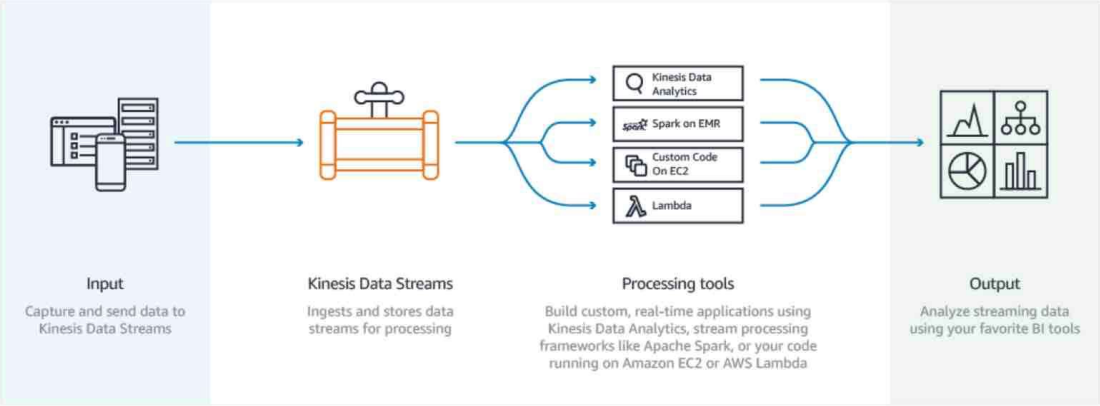
Use Enhanced Fanout feature of Amazon Kinesis Data Streams

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

of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events.

By default, the 2MB/second/shard output is shared between all of the applications consuming data from the stream. You should use enhanced fan-out if you have multiple consumers retrieving data from a stream in parallel. With enhanced fan-out developers can register stream consumers to use enhanced fan-out and receive their own 2MB/second pipe of read throughput per shard, and this throughput automatically scales with the number of shards in a stream.

Amazon Kinesis Data Streams Fanout:



Kinesis Data Streams are scaled using the concept of a **shard**. One shard provides an ingest capacity of 1MB/second or 1000 records/second and an output capacity of 2MB/second. It's not uncommon for customers to have thousands or tens of thousands of shards supporting 10s of GB/sec of ingest and egress. Before the enhanced fan-out capability, that 2MB/second/shard output was shared between all of the applications consuming data from the stream. With enhanced fan-out developers can register stream consumers to use enhanced fan-out and receive their own 2MB/second pipe of read throughput per shard, and this throughput automatically scales with the number of shards in a stream. Prior to the launch of Enhanced Fan-out customers would frequently fan-out their data out to multiple streams to support their desired read throughput for their downstream applications. That sounds like undifferentiated heavy lifting to us, and that's something we decided our customers shouldn't need to worry about. Customers pay for enhanced fan-out based on the amount of data retrieved from the stream using enhanced fan-out and the number of consumers registered per-shard. You can find additional info on the pricing page.

via - https://aws.amazon.com/blogs/aws/kds-enhanced-fanout/

Incorrect options:

Swap out Amazon Kinesis Data Streams with Amazon Kinesis Data Firehose - Amazon Kinesis Data Firehose is the easiest way to reliably load streaming data into data lakes, data stores, and analytics tools. It is a fully managed service that automatically scales to match the throughput of your data and requires no ongoing administration. It can also batch, compress, transform, and encrypt the data before loading it, minimizing the amount of storage used at the destination and increasing security. Amazon Kinesis Data Firehose can only write to Amazon S3, Amazon Redshift, Amazon Elasticsearch or Splunk. You can't have applications consuming data streams from Amazon Kinesis Data Firehose, that's the job of Amazon Kinesis Data Streams. Therefore this option is not correct.

Swap out Amazon Kinesis Data Streams with Amazon SQS Standard queues

Swap out Amazon Kinesis Data Streams with Amazon SQS FIFO queues

Amazon Simple Queue Service (Amazon SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. Amazon SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. Amazon SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent. As multiple applications are consuming the same stream concurrently, both Amazon SQS Standard and Amazon SQS FIFO are not the right fit for the given use-case.

Exam Alert:

Please understand the differences between the capabilities of Amazon Kinesis Data Streams vs Amazon SQS, as you may be asked scenario-based questions on this topic in the exam.

Q: When should I use Amazon Kinesis Data Streams, and when should I use Amazon SQS?

We recommend Amazon Kinesis Data Streams for use cases with requirements that are similar to the following:

- Routing related records to the same record processor (as in streaming MapReduce). For example, counting and aggregation are simpler when all records for a given key are routed to the same record processor.
- Ordering of records. For example, you want to transfer log data from the application host to the processing/archival host while maintaining the order of log statements.
- Ability for multiple applications to consume the same stream concurrently. For example, you have one application that updates a real-time dashboard and another that archives data to Amazon Redshift. You want both applications to consume data from the same stream concurrently and independently.
- Ability to consume records in the same order a few hours later. For example, you have a billing application and an audit application that runs a few hours behind the billing application. Because Amazon Kinesis Data Streams stores data for up to 7 days, you can run the audit application up to 7 days behind the billing application.

We recommend Amazon SQS for use cases with requirements that are similar to the following:

- Messaging semantics (such as message-level ack/fail) and visibility timeout. For example, you have a queue of work items and want to track the successful completion of each item independently. Amazon SQS tracks the ack/fail, so the application does not have to maintain a persistent checkpoint/cursor. Amazon SQS will delete acked messages and redeliver failed messages after a configured visibility timeout.
- Individual message delay. For example, you have a job queue and need to schedule individual jobs with a delay. With Amazon SQS, you can configure individual messages to have a delay of up to 15 minutes.
- Dynamically increasing concurrency/throughput at read time. For example, you have a work queue and want to add more readers until the backlog is cleared. With Amazon Kinesis Data Streams, you can scale up to a sufficient number of shards (note, however, that you'll need to provision enough shards ahead of time).
- Leveraging Amazon SQS's ability to scale transparently. For example, you buffer requests and the load changes as a result of occasional load spikes or the natural growth of your business. Because each buffered request can be processed independently, Amazon SQS can scale transparently to handle the load without any provisioning instructions from you.

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

References:

https://aws.amazon.com/blogs/aws/kds-enhanced-fanout/

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

Domain

Design High-Performing Architectures

Question 11Skipped

An application hosted on Amazon EC2 contains sensitive personal information about all its customers and needs to be protected from all types of cyber-attacks. The company is considering using the AWS Web Application Firewall (AWS WAF) to handle this requirement.

Can you identify the correct solution leveraging the capabilities of AWS WAF?

Configure an Application Load Balancer (ALB) to balance the workload for all the Amazon EC2 instances. Configure Amazon CloudFront to distribute from an Application Load Balancer since AWS WAF cannot be directly configured on ALB. This configuration not only provides necessary safety but is scalable too

AWS WAF can be directly configured on Amazon EC2 instances for ensuring the security of the underlying application data

Correct answer

Create Amazon CloudFront distribution for the application on Amazon EC2 instances. Deploy AWS WAF on Amazon CloudFront to provide the necessary safety measures

AWS WAF can be directly configured only on an Application Load Balancer or an Amazon API Gateway. One of these two services can then be configured with Amazon EC2 to build the needed secure architecture

Overall explanation

Correct option:

Create Amazon CloudFront distribution for the application on Amazon EC2 instances. Deploy AWS WAF on Amazon CloudFront to provide the necessary safety measures

When you use AWS WAF with Amazon CloudFront, you can protect your applications running on any HTTP webserver, whether it's a webserver that's running in Amazon Elastic Compute Cloud (Amazon EC2) or a web server that you manage privately. You can also configure Amazon CloudFront to require HTTPS between CloudFront and your own webserver, as well as between viewers and Amazon CloudFront.

AWS WAF is tightly integrated with Amazon CloudFront and the Application Load Balancer (ALB), 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 Application Load Balancer, your rules run in the region and can be used to protect internet-facing as well as internal load balancers.

Incorrect options:

Configure an Application Load Balancer (ALB) to balance the workload for all the Amazon EC2 instances. Configure Amazon CloudFront to distribute from an Application Load Balancer since AWS WAF cannot be directly configured on ALB. This configuration not only provides necessary safety but is scalable too - This statement is wrong. You can configure AWS WAF on Application Load Balancers (ALB).

AWS WAF can be directly configured on Amazon EC2 instances for ensuring the security of the underlying application data - AWS WAF can be deployed on Amazon CloudFront, the Application Load Balancer (ALB), and Amazon API Gateway. It cannot be configured directly on an Amazon EC2 instance.

AWS WAF can be directly configured only on an Application Load Balancer or an Amazon API Gateway. One of these two services can then be configured with Amazon EC2 to build the needed secure architecture - This statement is only partially correct. AWS WAF can also be deployed on Amazon CloudFront service.

References:

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

https://docs.aws.amazon.com/waf/latest/developerguide/cloudfront-features.html

Domain

Design Secure Architectures

Question 12Skipped

A medical devices company uses Amazon S3 buckets to store critical data. Hundreds of buckets are used to keep the data segregated and well organized. Recently, the development team noticed that the lifecycle policies on the Amazon S3 buckets have not been applied optimally, resulting in higher costs.

As a Solutions Architect, can you recommend a solution to reduce storage costs on Amazon S3 while keeping the IT team's involvement to a minimum?

Use Amazon S3 Outposts storage class to reduce the costs on Amazon S3 storage by storing the data on-premises

Configure Amazon EFS to provide a fast, cost-effective and sharable storage service

Use Amazon S3 One Zone-Infrequent Access, to reduce the costs on Amazon S3 storage

Correct answer

Use Amazon S3 Intelligent-Tiering storage class to optimize the Amazon S3 storage costs

Overall explanation

Correct option:

Use Amazon S3 Intelligent-Tiering storage class to optimize the Amazon S3 storage costs

The Amazon S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. It works by storing objects in two access tiers: one tier that is optimized for frequent access and another lower-cost tier that is optimized for infrequent access.

For a small monthly monitoring and automation fee per object, Amazon S3 monitors access patterns of the objects in Amazon S3 Intelligent-Tiering and moves the ones that have not been accessed for 30 consecutive days to the infrequent access tier. If an object in the infrequent access tier is accessed, it is automatically moved back to the frequent access tier. There are no

retrieval fees when using the Amazon S3 Intelligent-Tiering storage class, and no additional tiering fees when objects are moved between access tiers. It is the ideal storage class for long-lived data with access patterns that are unknown or unpredictable.

Amazon S3 Storage Classes can be configured at the object level and a single bucket can contain objects stored in Amazon S3 Standard, Amazon S3 Intelligent-Tiering, Amazon S3 Standard-IA, and Amazon S3 One Zone-IA. You can upload objects directly to Amazon S3 Intelligent-Tiering, or use S3 Lifecycle policies to transfer objects from Amazon S3 Standard and Amazon S3 Standard-IA to Amazon S3 Intelligent-Tiering. You can also archive objects from Amazon S3 Intelligent-Tiering to Amazon S3 Glacier.

Incorrect options:

Configure Amazon EFS to provide a fast, cost-effective and sharable storage service - Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. Amazon EFS offers sharable service, unlike Amazon Elastic Block Storage (EBS) that cannot be shared by instances. Amazon EFS is costlier than storing data in Amazon S3. Also, Amazon EFS needs an Amazon EC2 instance or an AWS Direct Connect network connection. Hence, this is not the correct option.

Use Amazon S3 One Zone-Infrequent Access, to reduce the costs on Amazon S3 storage - Amazon S3 One Zone-IA is for data that is accessed less frequently but requires rapid access when needed. Unlike other Amazon S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), Amazon S3 One Zone-IA stores data in a single AZ and costs 20% less than Amazon S3 Standard-IA. Amazon S3 One Zone-IA is ideal for customers who want a lower-cost option for infrequently accessed data but do not require the availability and resilience of Amazon S3 Standard or Amazon S3 Standard-IA. Not a right option, since data stored is business-critical and cannot be risked by using Amazon S3 One Zone-IA.

Use Amazon S3 Outposts storage class to reduce the costs on Amazon S3 storage by storing the data on-premises - This is a distractor as Amazon S3 on Outposts (S3 Outposts) delivers object storage to your on-premises AWS Outposts environment. It is used in conjunction with AWS Outposts and has no relevance to the current use case.

Reference:

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

Domain

Design Cost-Optimized Architectures

Question 13Skipped

A multi-national company is looking at optimizing their AWS resources across various countries and regions. They want to understand the best practices on cost optimization, performance, and security for their system architecture spanning across multiple business units.

Which AWS service is the best fit for their requirements?

AWS Systems Manager

AWS Management Console

Correct answer

AWS Trusted Advisor

AWS Config

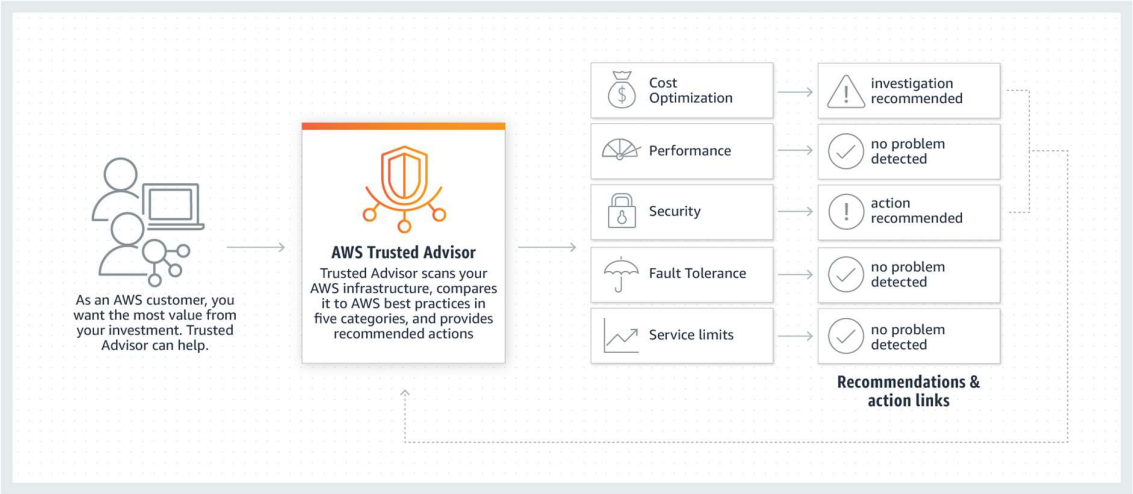
Overall explanation

Correct option:

AWS Trusted Advisor

AWS Trusted Advisor is an online tool that draws upon best practices learned from AWS's aggregated operational history of serving hundreds of thousands of AWS customers. AWS Trusted Advisor inspects your AWS environment and makes recommendations for saving money, improving system performance, or closing security gaps. It scans your AWS infrastructure and compares it to AWS Best practices in five categories (Cost Optimization, Performance, Security, Fault Tolerance, Service limits) and then provides recommendations.

How AWS Trusted Advisor Works:



via - https://aws.amazon.com/premiumsupport/technology/trusted-advisor/

Incorrect options:

AWS Config - AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. You can use Config to answer questions such as - "What did my AWS resource look like at xyz point in time?". It does not offer any feedback about architectural best practices.

AWS Management Console - The AWS Management Console is a web application that comprises and refers to a broad collection of service consoles for managing Amazon Web Services. You log into your AWS account using the AWS Management console. It does not offer any feedback about architectural best practices.

AWS Systems Manager - AWS Systems Manager is an AWS service that you can use to view and control your infrastructure on AWS. Using the Systems Manager console, you can view

operational data from multiple AWS services and automate operational tasks across your AWS resources. With Systems Manager, you can group resources, like Amazon EC2 instances, Amazon S3 buckets, or Amazon RDS instances, by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources. It does not offer any feedback about architectural best practices.

Reference:

https://aws.amazon.com/premiumsupport/technology/trusted-advisor/

Domain

Design Cost-Optimized Architectures

Question 14Skipped

A leading media company wants to do an accelerated online migration of hundreds of terabytes of files from their on-premises data center to Amazon S3 and then establish a mechanism to access the migrated data for ongoing updates from the on-premises applications.

As a solutions architect, which of the following would you select as the MOST performant solution for the given use-case?

Use Amazon S3 Transfer Acceleration (Amazon S3TA) to migrate existing data to Amazon S3 and then use AWS DataSync for ongoing updates from the on-premises applications

Use AWS DataSync to migrate existing data to Amazon S3 as well as access the Amazon S3 data for ongoing updates

Use File Gateway configuration of AWS Storage Gateway to migrate data to Amazon S3 and then use Amazon S3 Transfer Acceleration (Amazon S3TA) for ongoing updates from the on-premises applications

Correct answer

Use AWS DataSync to migrate existing data to Amazon S3 and then use File Gateway to retain access to the migrated data for ongoing updates from the on-premises applications

Overall explanation

Correct option:

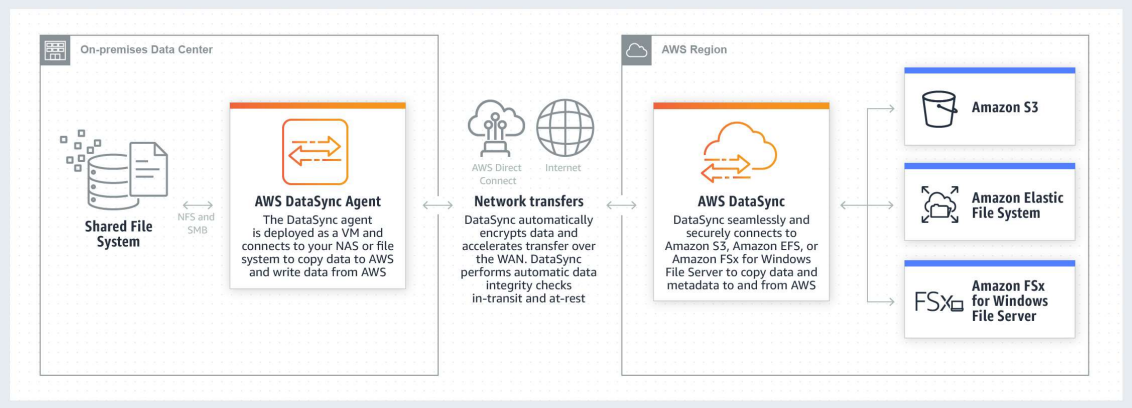
Use AWS DataSync to migrate existing data to Amazon S3 and then use File Gateway to retain access to the migrated data for ongoing updates from the on-premises applications

AWS DataSync is an online data transfer service that simplifies, automates, and accelerates copying large amounts of data to and from AWS storage services over the internet or AWS Direct Connect.

AWS DataSync fully automates and accelerates moving large active datasets to AWS, up to 10 times faster than command-line tools. It is natively integrated with Amazon S3, Amazon EFS, Amazon FSx for Windows File Server, Amazon CloudWatch, and AWS CloudTrail, which provides seamless and secure access to your storage services, as well as detailed monitoring of the transfer. DataSync uses a purpose-built network protocol and scale-out architecture to transfer data. A single AWS DataSync agent is capable of saturating a 10 Gbps network link.

AWS DataSync fully automates the data transfer. It comes with retry and network resiliency mechanisms, network optimizations, built-in task scheduling, monitoring via the AWS DataSync API and Console, and Amazon CloudWatch metrics, events, and logs that provide granular visibility into the transfer process. AWS DataSync performs data integrity verification both during the transfer and at the end of the transfer.

How AWS DataSync Works:



via - https://aws.amazon.com/datasync/

AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. The service provides three different types of gateways – Tape Gateway, File Gateway, and Volume Gateway – that seamlessly connect on-premises applications to cloud storage, caching data locally for low-latency access. File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching.

The combination of AWS DataSync and File Gateway is the correct solution. AWS DataSync enables you to automate and accelerate online data transfers to AWS storage services. File Gateway then provides your on-premises applications with low latency access to the migrated data.

Incorrect options:

Use AWS DataSync to migrate existing data to Amazon S3 as well as access the Amazon S3 data for ongoing updates - AWS DataSync is used to easily transfer data to and from AWS with up to 10x faster speeds. It is used to transfer data and cannot be used to facilitate ongoing updates to the migrated files from the on-premises applications.

Use File Gateway configuration of AWS Storage Gateway to migrate data to Amazon S3 and then use Amazon S3 Transfer Acceleration (Amazon S3TA) for ongoing updates from the on-premises applications - File Gateway can be used to move on-premises data to AWS Cloud, but it not an optimal solution for high volumes. Migration services such as AWS DataSync are best suited for this purpose. Amazon S3 Transfer Acceleration cannot facilitate ongoing updates to the migrated files from the on-premises applications.

Use Amazon S3 Transfer Acceleration (Amazon S3TA) to migrate existing data to Amazon S3 and then use AWS DataSync for ongoing updates from the on-premises applications - If your application is already integrated with the Amazon S3 API, and you want higher throughput for transferring large files to Amazon S3, Amazon S3 Transfer Acceleration can be used.

However AWS DataSync cannot be used to facilitate ongoing updates to the migrated files from the on-premises applications.

Reference:

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

Domain

Design High-Performing Architectures

Question 15Skipped

An IT company has built a custom data warehousing solution for a retail organization by using Amazon Redshift. As part of the cost optimizations, the company wants to move any historical data (any data older than a year) into Amazon S3, as the daily analytical reports consume data for just the last one year. However the analysts want to retain the ability to cross-reference this historical data along with the daily reports.

The company wants to develop a solution with the LEAST amount of effort and MINIMUM cost. As a solutions architect, which option would you recommend to facilitate this use-case?

Setup access to the historical data via Amazon Athena. The analytics team can run historical data queries on Amazon Athena and continue the daily reporting on Amazon Redshift. In case the reports need to be cross-referenced, the analytics team need to export these in flat files and then do further analysis

Correct answer

Use Amazon Redshift Spectrum to create Amazon Redshift cluster tables pointing to the underlying historical data in Amazon S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift

Use AWS Glue ETL job to load the Amazon S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift

Use the Amazon Redshift COPY command to load the Amazon S3 based historical data into Amazon Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift

Overall explanation

Correct option:

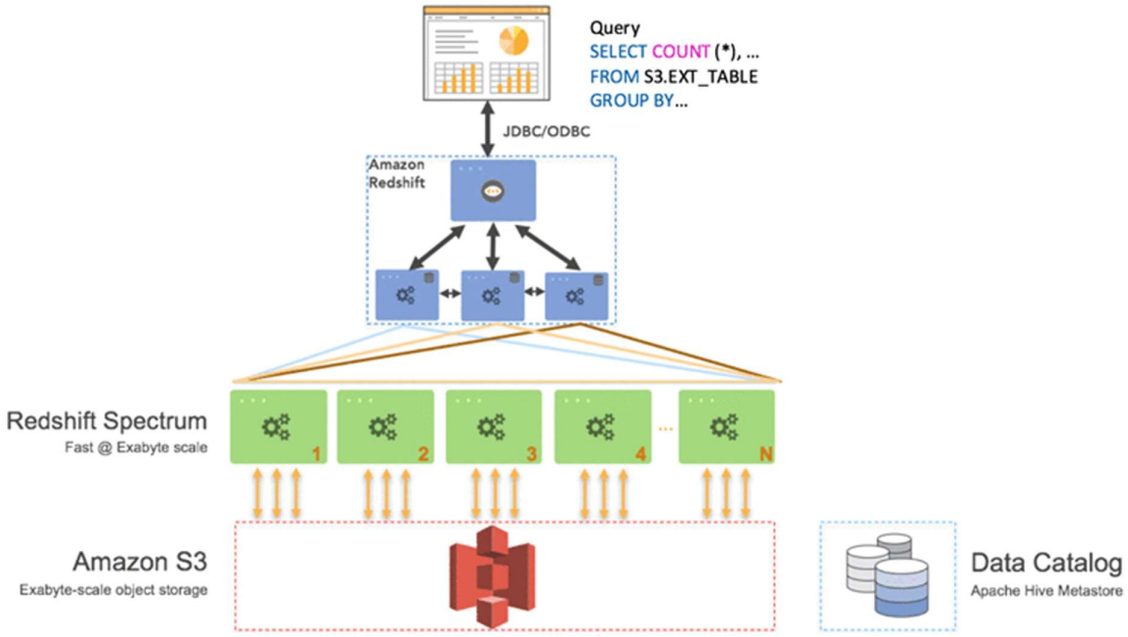
Use Amazon Redshift Spectrum to create Amazon Redshift cluster tables pointing to the underlying historical data in Amazon S3. The analytics team can then query this historical data to cross-reference with the daily reports from Redshift

Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis.

Using Amazon Redshift Spectrum, you can efficiently query and retrieve structured and semistructured data from files in Amazon S3 without having to load the data into Amazon Redshift tables.

Amazon Redshift Spectrum resides on dedicated Amazon Redshift servers that are independent of your cluster. Redshift Spectrum pushes many compute-intensive tasks, such as predicate filtering and aggregation, down to the Redshift Spectrum layer. Thus, Amazon Redshift Spectrum queries use much less of your cluster's processing capacity than other queries.

Redshift Spectrum Overview:



via - https://aws.amazon.com/blogs/big-data/amazon-redshift-spectrum-extends-data-warehousing-out-to-exabytes-no-loading-required/

Incorrect options:

Setup access to the historical data via Amazon Athena. The analytics team can run historical data queries on Amazon Athena and continue the daily reporting on Amazon Redshift. In case the reports need to be cross-referenced, the analytics team need to export these in flat files and then do further analysis - Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries. Providing access to historical data via Athena would mean that historical data reconciliation would become difficult as the daily report would still be produced via Redshift. Such a setup is cumbersome to maintain on a day to day basis. Hence the option to use Athena is ruled out.

Use the Amazon Redshift COPY command to load the Amazon S3 based historical data into Amazon Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift

Use AWS Glue ETL job to load the Amazon S3 based historical data into Redshift. Once the ad-hoc queries are run for the historic data, it can be removed from Amazon Redshift

Loading historical data into Amazon Redshift via COPY command or AWS Glue ETL job would cost heavy for a one-time ad-hoc process. The same result can be achieved more cost-efficiently by using Amazon Redshift Spectrum. Therefore both these options to load historical data into Redshift are also incorrect for the given use-case.

References:

https://docs.aws.amazon.com/redshift/latest/dg/c-using-spectrum.html#c-spectrum-overview
https://aws.amazon.com/blogs/big-data/

amazon-redshift-spectrum-extends-data-warehousing-out-to-exabytes-no-loading-required/

Domain

Design High-Performing Architectures

Question 16Skipped

A financial services company is moving its IT infrastructure to AWS Cloud and wants to enforce adequate data protection mechanisms on Amazon Simple Storage Service (Amazon S3) to meet compliance guidelines. The engineering team has hired you as a solutions architect to build a solution for this requirement.

Can you help the team identify the INCORRECT option from the choices below?

Amazon S3 can protect data at rest using Client-Side Encryption

Amazon S3 can protect data at rest using Server-Side Encryption

Correct answer

Amazon S3 can encrypt object metadata by using Server-Side Encryption

Amazon S3 can encrypt data in transit using HTTPS (TLS)

Overall explanation

Correct option:

Amazon S3 can encrypt object metadata by using Server-Side Encryption

Amazon S3 is a simple key-value store designed to store as many objects as you want. You store these objects in one or more buckets, and each object can be up to 5 TB in size.

An object consists of the following:

Key – The name that you assign to an object. You use the object key to retrieve the object.

Version ID – Within a bucket, a key and version ID uniquely identify an object.

Value – The content that you are storing.

Metadata – A set of name-value pairs with which you can store information regarding the object.

Subresources – Amazon S3 uses the subresource mechanism to store object-specific additional information.

Access Control Information – You can control access to the objects you store in Amazon S3.

Metadata, which can be included with the object, is not encrypted while being stored on Amazon S3. Therefore, AWS recommends that customers not place sensitive information in Amazon S3 metadata.

Incorrect options:

Amazon S3 can protect data at rest using Server-Side Encryption - This is possible and AWS provides three different ways of doing this - Server-side encryption with Amazon S3-managed keys (SSE-S3), Server-side encryption with customer master keys stored in AWS Key Management Service (SSE-KMS), Server-side encryption with customer-provided keys (SSE-C).

Amazon S3 can protect data at rest using Client-Side Encryption - This is a possible scenario too. You can encrypt data on the client-side and upload the encrypted data to Amazon S3. In this case, the client manages the encryption process, the encryption keys, and related tools.

Amazon S3 can encrypt data in transit using HTTPS (TLS) - This is also possible and you can use HTTPS (TLS) to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/security-best-practices.html#server-side

https://d1.awsstatic.com/whitepapers/aws-security-whitepaper.pdf?did=wp_card&trk=wp_card

Domain

Design Secure Architectures

Question 17Skipped

An Internet-of-Things (IoT) company is planning on distributing a master sensor in people's homes to measure the key metrics from its smart devices. In order to provide adjustment commands for these devices, the company would like to have a streaming system that supports ordered data based on the sensor's key, and also sustains high throughput messages (thousands of messages per second).

As a solutions architect, which of the following AWS services would you recommend for this use-case?

Correct answer

Amazon Kinesis Data Streams

AWS Lambda

Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Notification Service (Amazon SNS)

Overall explanation

Correct option:

Amazon Kinesis Data Streams

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events. The throughput of an Amazon Kinesis data stream is designed to scale without limits via increasing the number of shards within a data stream.

However, there are certain limits you should keep in mind while using Amazon Kinesis Data Streams:

A Kinesis data stream stores records from 24 hours by default, up to 8760 hours (365 days).

The maximum size of a data blob (the data payload before Base64-encoding) within one record is 1 megabyte (MB). Each shard can support up to 1000 PUT records per second.

Kinesis is the right answer here, as by providing a partition key in your message, you can guarantee ordered messages for a specific sensor, even if your stream is sharded.

Incorrect options:

Amazon Simple Queue Service (Amazon SQS) - Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. Amazon SQS eliminates the complexity and overhead associated with managing and operating message-oriented middleware, and empowers developers to focus on differentiating work. Using Amazon SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Kinesis is better for streaming data since queues aren't meant for real-time streaming of data.

Amazon Simple Notification Service (Amazon SNS) - Amazon Simple Notification Service (Amazon SNS) is a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. Amazon SNS provides topics for high-throughput, push-based, many-to-many messaging. SNS cannot be used for data streaming. Therefore this option is not the best fit for the given use-case.

AWS Lambda - AWS Lambda lets you run code without provisioning or managing servers. It cannot be used for production-grade serverless log analytics. Lambda isn't meant to retain data either. Therefore this option is not the best fit for the given use-case.

Reference:

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

Domain

Design High-Performing Architectures

Question 18Skipped

The infrastructure team at a company maintains 5 different VPCs (let's call these VPCs A, B, C, D, E) for resource isolation. Due to the changed organizational structure, the team wants to interconnect all VPCs together. To facilitate this, the team has set up VPC peering connection

between VPC A and all other VPCs in a hub and spoke model with VPC A at the center. However, the team has still failed to establish connectivity between all VPCs.

As a solutions architect, which of the following would you recommend as the MOST resource-efficient and scalable solution?

Use an internet gateway to interconnect the VPCs

Establish VPC peering connections between all VPCs

Use a VPC endpoint to interconnect the VPCs

Correct answer

Use AWS transit gateway to interconnect the VPCs

Overall explanation

Correct option:

Use AWS transit gateway to interconnect the VPCs

An AWS transit gateway is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

AWS Transit Gateway Overview:

What is a transit gateway?

PDF

A *transit gateway* is a network transit hub that you can use to interconnect your virtual private clouds (VPC) and on-premises networks.

For more information, see AWS Transit Gateway.

Transit gateway concepts

The following are the key concepts for transit gateways:

- **attachment** — You can attach a VPC, an AWS Direct Connect gateway, a peering connection with another transit gateway, or a VPN connection to a transit gateway.
- **transit gateway Maximum Transmission Unit (MTU)** — The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data can be passed in a single packet. A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, Direct Connect and peering attachments. Traffic over VPN connections can have an MTU of 1500 bytes.
- **transit gateway route table** — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be a VPC or a VPN connection. By default, transit gateway attachments are associated with the default transit gateway route table.
- **associations** — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.
- **route propagation** — A VPC or VPN connection can dynamically propagate routes to a transit gateway route table. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP). With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment.

via - https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html

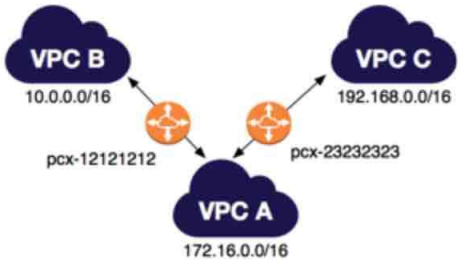
A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Transitive Peering does not work for VPC peering connections. So, if you have a VPC peering connection between VPC A and VPC B (pcx-aaaabbbb), and between VPC A and VPC C (pcx-aaaacccc). Then, there is no VPC peering connection between VPC B and VPC C. Instead of using VPC peering, you can use an AWS Transit Gateway that acts as a network transit hub, to interconnect your VPCs or connect your VPCs with on-premises networks. Therefore this is the correct option.

VPC Peering Connections Overview:

Multiple VPC peering connections

A VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported. You do not have any peering relationship with VPCs that your VPC is not directly peered with.

The following diagram is an example of one VPC peered to two different VPCs. There are two VPC peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.



via - https://docs.aws.amazon.com/vpc/latest/peering/vpc-peering-basics.html

Incorrect options:

Use an internet gateway to interconnect the VPCs - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth constraints on your network traffic. You cannot use an internet gateway to interconnect your VPCs and on-premises networks, hence this option is incorrect.

Use a VPC endpoint to interconnect the VPCs - A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. You cannot use a VPC endpoint to interconnect your VPCs and on-premises networks, hence this option is incorrect.

Establish VPC peering connections between all VPCs - Establishing VPC peering between all VPCs is an inelegant and clumsy way to establish connectivity between all VPCs. Instead, you should use a Transit Gateway that acts as a network transit hub to interconnect your VPCs and on-premises networks.

References:

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

https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html

Domain

Design Secure Architectures

Question 19Skipped

A silicon valley based healthcare startup uses AWS Cloud for its IT infrastructure. The startup stores patient health records on Amazon Simple Storage Service (Amazon S3). The engineering team needs to implement an archival solution based on Amazon S3 Glacier to enforce regulatory and compliance controls on data access.

As a solutions architect, which of the following solutions would you recommend?

Correct answer

Use Amazon S3 Glacier vault to store the sensitive archived data and then use a vault lock policy to enforce compliance controls

Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 lifecycle policy to enforce compliance controls

Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls

Use Amazon S3 Glacier vault to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls

Overall explanation

Correct option:

Use Amazon S3 Glacier vault to store the sensitive archived data and then use a vault lock policy to enforce compliance controls

Amazon S3 Glacier is a secure, durable, and extremely low-cost Amazon S3 cloud storage class for data archiving and long-term backup. It is designed to deliver 99.999999999% durability, and provide comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements.

An Amazon S3 Glacier vault is a container for storing archives. When you create a vault, you specify a vault name and the AWS Region in which you want to create the vault. Amazon S3 Glacier Vault Lock allows you to easily deploy and enforce compliance controls for individual Amazon S3 Glacier vaults with a vault lock policy. You can specify controls such as “write once read many” (WORM) in a vault lock policy and lock the policy from future edits. Therefore, this is the correct option.

Incorrect options:

Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 lifecycle policy to enforce compliance controls - You can use lifecycle policy to define actions you want Amazon S3 to take during an object's lifetime. For example, use a lifecycle policy to transition objects to another storage class, archive them, or delete them after a specified period. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

Use Amazon S3 Glacier vault to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls- Amazon S3 access control lists (ACLs) enable you to manage access to buckets and objects. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

Use Amazon S3 Glacier to store the sensitive archived data and then use an Amazon S3 Access Control List to enforce compliance controls - Amazon S3 access control lists (ACLs) enable you to manage access to buckets and objects. It cannot be used to enforce compliance controls. Therefore, this option is incorrect.

References:

https://docs.aws.amazon.com/amazonglacier/latest/dev/working-with-vaults.html

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

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/create-lifecycle.html

Domain

Design Secure Architectures

Question 20Skipped

A pharma company is working on developing a vaccine for the COVID-19 virus. The researchers at the company want to process the reference healthcare data in a highly available as well as HIPAA compliant in-memory database that supports caching results of SQL queries.

As a solutions architect, which of the following AWS services would you recommend for this task?

Amazon DocumentDB

Amazon DynamoDB

Correct answer

Amazon ElastiCache for Redis/Memcached

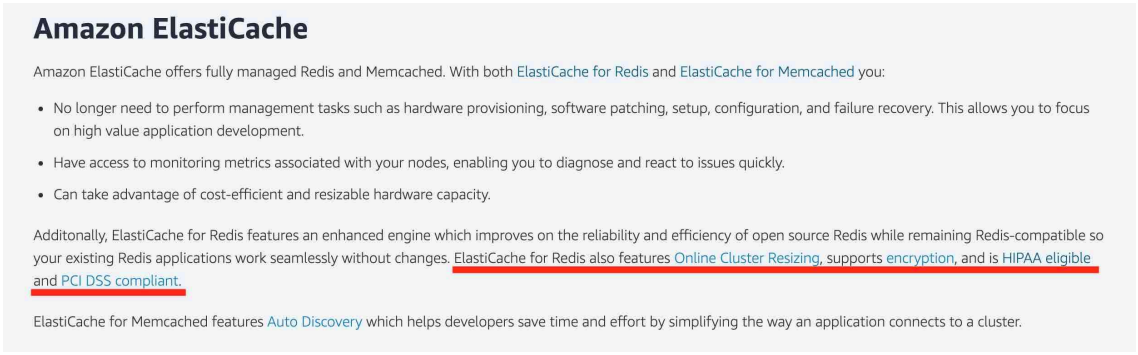
Amazon DynamoDB Accelerator (DAX)

Overall explanation

Correct option:

Amazon ElastiCache for Redis/Memcached

Amazon ElastiCache Overview:



Amazon ElastiCache

Amazon ElastiCache offers fully managed Redis and Memcached. With both ElastiCache for Redis and ElastiCache for Memcached you:

- No longer need to perform management tasks such as hardware provisioning, software patching, setup, configuration, and failure recovery. This allows you to focus on high value application development.
- Have access to monitoring metrics associated with your nodes, enabling you to diagnose and react to issues quickly.
- Can take advantage of cost-efficient and resizable hardware capacity.

Additonally, ElastiCache for Redis features an enhanced engine which improves on the reliability and efficiency of open source Redis while remaining Redis-compatible so your existing Redis applications work seamlessly without changes. ElastiCache for Redis also features Online Cluster Resizing, supports encryption, and is HIPAA eligible and PCI DSS compliant.

ElastiCache for Memcached features Auto Discovery which helps developers save time and effort by simplifying the way an application connects to a cluster.

via - https://aws.amazon.com/elasticache/redis-vs-memcached/

Amazon ElastiCache for Redis is a blazing fast in-memory data store that provides sub-millisecond latency to power internet-scale real-time applications. Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store. ElastiCache for Redis supports replication, high availability, and cluster sharding right out of the box.

Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Amazon ElastiCache for Memcached is a great choice for implementing an in-memory cache to decrease access latency, increase throughput, and ease the load off your relational or NoSQL database. Session stores are easy to create with Amazon ElastiCache for Memcached.

Both Amazon ElastiCache for Redis and Amazon ElastiCache for Memcached are HIPAA Eligible. Therefore, this is the correct option.

Exam Alert:

Please review this comparison sheet for Redis vs Memcached features:

Choosing between Redis and Memcached

Redis and Memcached are popular, open-source, in-memory data stores. Although they are both easy to use and offer high performance, there are important differences to consider when choosing an engine. Memcached is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Understand your requirements and what each engine offers to decide which solution better meets your needs.

Learn about Amazon ElastiCache for Redis | Learn about Amazon ElastiCache for Memcached

	Memcached	Redis
Sub-millisecond latency	Yes	Yes
Developer ease of use	Yes	Yes
Data partitioning	Yes	Yes
Support for a broad set of programming languages	Yes	Yes
Advanced data structures	-	Yes
Multithreaded architecture	Yes	-
Snapshots	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Lua scripting	-	Yes
Geospatial support	-	Yes

via - https://aws.amazon.com/elasticache/redis-vs-memcached/

Incorrect Options:

Amazon DynamoDB Accelerator (DAX) - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-master, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DAX is a DynamoDB-compatible caching service that enables you to benefit from fast in-memory performance for demanding applications. DAX does not support SQL query caching.

Amazon DynamoDB - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multi-region, multi-master, durable database with built-in security, backup and restore, and in-memory caching (via DAX) for internet-scale applications. Amazon DynamoDB is not an in-memory database, so this option is incorrect.

Amazon DocumentDB - Amazon DocumentDB is a fast, scalable, highly available, and fully managed document database service that supports MongoDB workloads. As a document database, Amazon DocumentDB makes it easy to store, query, and index JSON data. Amazon DocumentDB is not an in-memory database, so this option is incorrect.

References:

https://aws.amazon.com/about-aws/whats-new/2017/11/amazon-elasticache-for-redis-is-now-hipaa-eligible-to-help-you-power-secure-healthcare-applications-with-sub-millisecond-latency/

https://aws.amazon.com/elasticache/redis/

https://aws.amazon.com/about-aws/whats-new/2022/08/amazon-elasticache-memcached-hipaa-eligible/

https://aws.amazon.com/blogs/database/automating-sql-caching-for-amazon-elasticache-and-amazon-rds/

Domain

Design Secure Architectures

Question 21Skipped

A health-care company manages its web application on Amazon EC2 instances running behind Auto Scaling group (ASG). The company provides ambulances for critical patients and needs the application to be reliable. The workload of the company can be managed on 2 Amazon EC2 instances and can peak up to 6 instances when traffic increases.

As a Solutions Architect, which of the following configurations would you select as the best fit for these requirements?

Correct answer

The Auto Scaling group should be configured with the minimum capacity set to 4, with 2 instances each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6

The Auto Scaling group should be configured with the minimum capacity set to 4, with 2 instances each in two different AWS Regions. The maximum capacity of the Auto Scaling group should be set to 6

The Auto Scaling group should be configured with the minimum capacity set to 2, with 1 instance each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6

The Auto Scaling group should be configured with the minimum capacity set to 2 and the maximum capacity set to 6 in a single Availability Zone

Overall explanation

Correct option:

The Auto Scaling group should be configured with the minimum capacity set to 4, with 2 instances each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6

You configure the size of your Auto Scaling group by setting the minimum, maximum, and desired capacity. The minimum and maximum capacity are required to create an Auto Scaling group, while the desired capacity is optional. If you do not define your desired capacity upfront, it defaults to your minimum capacity.

Amazon EC2 Auto Scaling enables you to take advantage of the safety and reliability of geographic redundancy by spanning Auto Scaling groups across multiple Availability Zones within a Region. When one Availability Zone becomes unhealthy or unavailable, Auto Scaling launches new instances in an unaffected Availability Zone. When the unhealthy Availability Zone returns to a healthy state, Auto Scaling automatically redistributes the application instances evenly across all of the designated Availability Zones. Since the application is extremely critical and needs to have a reliable architecture to support it, the Amazon EC2 instances should be maintained in at least two Availability Zones (AZs) for uninterrupted service.

Amazon EC2 Auto Scaling attempts to distribute instances evenly between the Availability Zones that are enabled for your Auto Scaling group. This is why the minimum capacity should be 4 instances and not 2. Auto Scaling group will launch 2 instances each in both the AZs and this redundancy is needed to keep the service available always.

Incorrect options:

The Auto Scaling group should be configured with the minimum capacity set to 2, with 1 instance each in two different Availability Zones. The maximum capacity of the Auto Scaling group should be set to 6

The Auto Scaling group should be configured with the minimum capacity set to 2 and the maximum capacity set to 6 in a single Availability Zone

The explanation above gives the correct rationale for minimum capacity as well as the instance distribution across AZs, so both these options are incorrect.

The Auto Scaling group should be configured with the minimum capacity set to 4, with 2 instances each in two different AWS Regions. The maximum capacity of the Auto Scaling group should be set to 6 - An Auto Scaling group can contain Amazon EC2 instances in one or

more Availability Zones within the same region. However, Auto Scaling groups cannot span multiple Regions.

Reference:

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

Domain

Design Resilient Architectures

Question 22Skipped

A DevOps engineer at an organization is debugging issues related to an Amazon EC2 instance. The engineer has SSH'ed into the instance and he needs to retrieve the instance public IP from within a shell script running on the instance command line.

Can you identify the correct URL path to get the instance public IP?

http://169.254.169.254/latest/user-data/public-ipv4

Correct answer

http://169.254.169.254/latest/meta-data/public-ipv4

http://254.169.254.169/latest/user-data/public-ipv4

http://254.169.254.169/latest/meta-data/public-ipv4

Overall explanation

Correct option:

http://169.254.169.254/latest/meta-data/public-ipv4

Instance metadata is the data about your instance that you can use to configure or manage the running instance.

Instance user data is the data that you specified in the form of a configuration script while launching your instance.

The following URL paths can be used to get the instance meta data and user data from within the instance: http://169.254.169.254/latest/meta-data/

http://169.254.169.254/latest/user-data/

Further, you can get the instance public IP via the URL - http://169.254.169.254/latest/meta-data/public-ipv4

Incorrect options:

http://169.254.169.254/latest/user-data/public-ipv4

http://254.169.254.169/latest/meta-data/public-ipv4

http://254.169.254.169/latest/user-data/public-ipv4

These three options do not meet the specification for the URL path to get the instance public IP, so these are incorrect.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-add-user-data.html

Domain

Design Resilient Architectures

Question 23Skipped

The engineering team at a retail company manages 3 Amazon EC2 instances that make read-heavy database requests to the Amazon RDS for the PostgreSQL database instance. As an AWS Certified Solutions Architect - Associate, you have been tasked to make the database instance resilient from a disaster recovery perspective.

Which of the following features will help you in disaster recovery of the database? (Select two)

Correct selection

Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups across multiple Regions

Use Amazon RDS Provisioned IOPS (SSD) Storage in place of General Purpose (SSD) Storage

Correct selection

Use cross-Region Read Replicas

Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups in a single AWS Region

Use the database cloning feature of the Amazon RDS Database cluster

Overall explanation

Correct options:

Use cross-Region Read Replicas

In addition to using Read Replicas to reduce the load on your source database instance, you can also use Read Replicas to implement a DR solution for your production DB environment. If the source DB instance fails, you can promote your Read Replica to a standalone source server. Read Replicas can also be created in a different Region than the source database. Using a cross-Region Read Replica can help ensure that you get back up and running if you experience a regional availability issue.

Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups across multiple Regions

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

The automated backup feature of Amazon RDS enables point-in-time recovery for your database instance. Amazon RDS will back up your database and transaction logs and store both for a user-specified retention period. If it's a Multi-AZ configuration, backups occur on standby to reduce the I/O impact on the primary. Amazon RDS supports Cross-Region Automated Backups. Manual snapshots and Read Replicas are also supported across multiple Regions.

Incorrect options:

Enable the automated backup feature of Amazon RDS in a multi-AZ deployment that creates backups in a single AWS Region - This is an incorrect statement. Automated backups can be created across AWS Regions.

Use Amazon RDS Provisioned IOPS (SSD) Storage in place of General Purpose (SSD) Storage - Amazon RDS Provisioned IOPS Storage is an SSD-backed storage option designed to deliver fast, predictable, and consistent I/O performance. This storage type enhances the performance of the RDS database, but this isn't a disaster recovery option.

Use the database cloning feature of the Amazon RDS Database cluster - This option has been added as a distractor. Database cloning is only available for Amazon Aurora and not for Amazon RDS.

References:

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

https://aws.amazon.com/blogs/database/implementing-a-disaster-recovery-strategy-with-amazon-rds/

https://aws.amazon.com/about-aws/whats-new/2021/07/amazon-rds-cross-region-automated-backups-regional-expansion/

Domain

Design Resilient Architectures

Question 24Skipped

A cyber security company is running a mission critical application using a single Spread placement group of Amazon EC2 instances. The company needs 15 Amazon EC2 instances for optimal performance.

How many Availability Zones (AZs) will the company need to deploy these Amazon EC2 instances per the given use-case?

14

15

7

Correct answer

3

Overall explanation

Correct option:

3

When you launch a new Amazon 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 placement group

Partition placement group

Spread placement group.

A Spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

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.

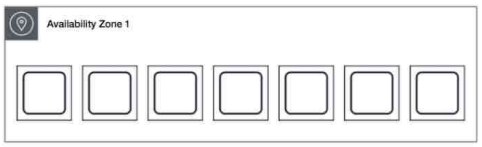
A spread placement group can span multiple Availability Zones in the same Region. You can have a maximum of seven running instances per Availability Zone per group. Therefore, to deploy 15 Amazon EC2 instances in a single Spread placement group, the company needs to use 3 Availability Zones.

Spread placement group overview:

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct racks, with each rack having its own network and power source.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks.



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.

If you start or launch an instance in a spread placement group and there is insufficient unique hardware to fulfill the request, the request fails. Amazon EC2 makes more distinct hardware available over time, so you can try your request again later.

via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html

Incorrect options:

7

14

15

These three options contradict the details provided in the explanation above, so these options are incorrect.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/placement-groups.html

Domain

Design Resilient Architectures

Question 25Skipped

While troubleshooting, a cloud architect realized that the Amazon EC2 instance is unable to connect to the internet using the Internet Gateway.

Which conditions should be met for internet connectivity to be established? (Select two)

The instance's subnet is associated with multiple route tables with conflicting configurations

Correct selection

The network access control list (network ACL) associated with the subnet must have rules to allow inbound and outbound traffic

The instance's subnet is not associated with any route table

Correct selection

The route table in the instance's subnet should have a route to an Internet Gateway

The subnet has been configured to be public and has no access to the internet

Overall explanation

Correct options:

The network access control list (network ACL) associated with the subnet must have rules to allow inbound and outbound traffic

The network access control list (network ACL) that is associated with the subnet must have rules to allow inbound and outbound traffic on port 80 (for HTTP traffic) and port 443 (for HTTPs traffic). This is a necessary condition for Internet Gateway connectivity.

The route table in the instance's subnet should have a route to an Internet Gateway

A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. The route table in the instance's subnet should have a route defined to the Internet Gateway.

Incorrect options:

The instance's subnet is not associated with any route table - This is an incorrect statement. A subnet is implicitly associated with the main route table if it is not explicitly associated with a particular route table. So, a subnet is always associated with some route table.

The instance's subnet is associated with multiple route tables with conflicting configurations - This is an incorrect statement. A subnet can only be associated with one route table at a time.

The subnet has been configured to be public and has no access to the internet - This is an incorrect statement. Public subnets have access to the internet via Internet Gateway.

Reference:

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

Domain

Design Secure Architectures

Question 26Skipped

As a Solutions Architect, you have been hired to work with the engineering team at a company to create a REST API using the serverless architecture.

Which of the following solutions will you recommend to move the company to the serverless architecture?

Amazon Route 53 with Amazon EC2 as backend

Correct answer

Amazon API Gateway exposing AWS Lambda Functionality

Public-facing Application Load Balancer with Amazon Elastic Container Service (Amazon ECS) on Amazon EC2

AWS Fargate with AWS Lambda at the front

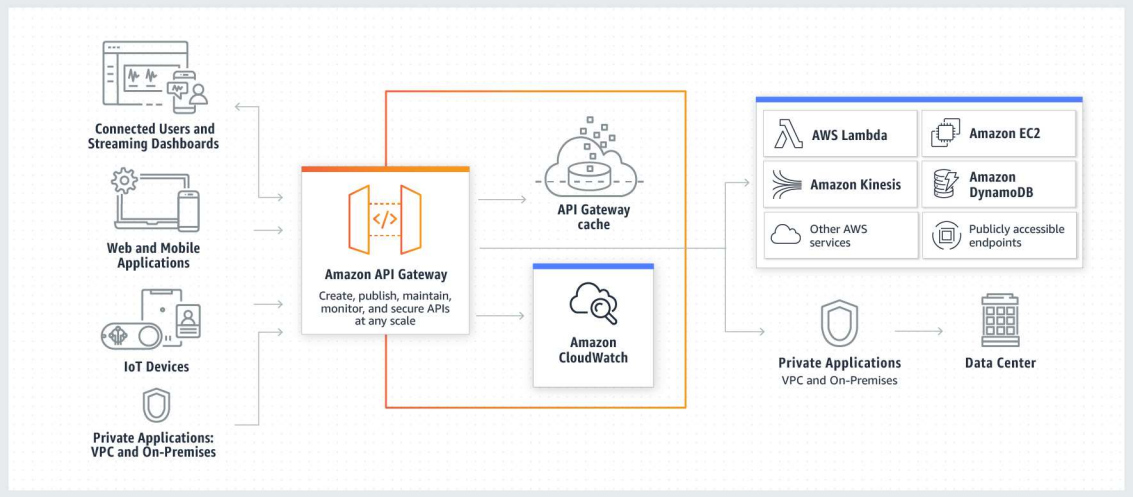
Overall explanation

Correct option:

Amazon API Gateway exposing AWS Lambda Functionality

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.

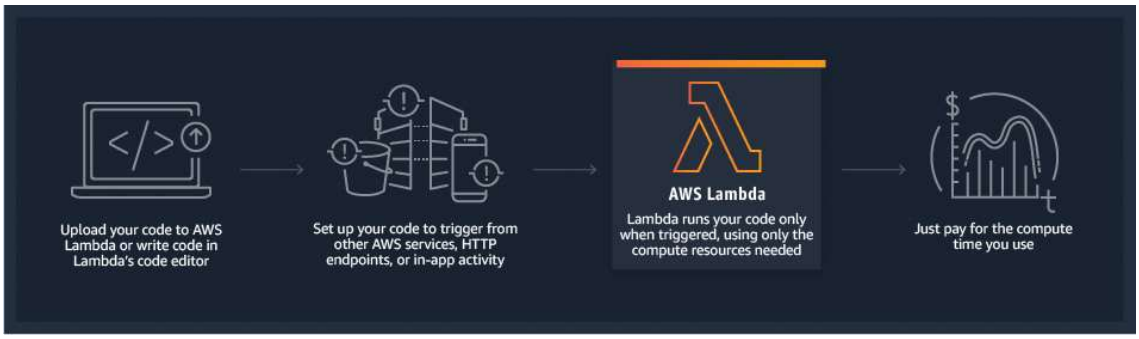
How Amazon API Gateway Works:



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

AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume.

How AWS Lambda function works:



via - https://aws.amazon.com/lambda/

Amazon API Gateway can expose AWS Lambda functionality through RESTful APIs. Both are serverless options offered by AWS and hence the right choice for this scenario, considering all the functionality they offer.

Incorrect options:

AWS Fargate with AWS Lambda at the front - AWS Lambda cannot directly handle RESTful API requests. You can invoke an AWS Lambda function over HTTPS by defining a custom RESTful API using Amazon API Gateway. So, AWS Fargate with AWS Lambda as the front-facing service is a wrong combination, though both Fargate and Lambda are serverless.

Public-facing Application Load Balancer with Amazon Elastic Container Service (Amazon ECS) on Amazon EC2 - Amazon ECS on Amazon EC2 does not come under serverless and hence cannot be considered for this use case.

Amazon Route 53 with Amazon EC2 as backend - Amazon EC2 is not a serverless service and hence cannot be considered for this use case.

References:

https://aws.amazon.com/serverless/

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

Domain

Design Secure Architectures

Question 27Skipped

A company has noticed that its application performance has deteriorated after a new Auto Scaling group was deployed a few days back. Upon investigation, the team found out that the Launch Configuration selected for the Auto Scaling group is using the incorrect instance type that is not optimized to handle the application workflow.

As a solutions architect, what would you recommend to provide a long term resolution for this issue?

Modify the launch configuration to use the correct instance type and continue to use the existing Auto Scaling group

Correct answer

Create a new launch configuration to use the correct instance type. Modify the Auto Scaling group to use this new launch configuration. Delete the old launch configuration as it is no longer needed

No need to modify the launch configuration. Just modify the Auto Scaling group to use more number of existing instance types. More instances may offset the loss of performance

No need to modify the launch configuration. Just modify the Auto Scaling group to use the correct instance type

Overall explanation

Correct option:

Create a new launch configuration to use the correct instance type. Modify the Auto Scaling group to use this new launch configuration. Delete the old launch configuration as it is no longer needed

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch Amazon EC2 instances. When you create a launch configuration, you specify information for the instances. Include the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping.

It is not possible to modify a launch configuration once it is created. The correct option is to create a new launch configuration to use the correct instance type. Then modify the Auto Scaling group to use this new launch configuration. Lastly to clean-up, just delete the old launch configuration as it is no longer needed.

Incorrect options:

Modify the launch configuration to use the correct instance type and continue to use the existing Auto Scaling group - As mentioned earlier, it is not possible to modify a launch configuration once it is created. Hence, this option is incorrect.

No need to modify the launch configuration. Just modify the Auto Scaling group to use the correct instance type - You cannot use an Auto Scaling group to directly modify the instance type of the underlying instances. Hence, this option is incorrect.

No need to modify the launch configuration. Just modify the Auto Scaling group to use more number of existing instance types. More instances may offset the loss of performance - Using the Auto Scaling group to increase the number of instances to cover up for the performance loss is not recommended as it does not address the root cause of the problem. The Machine Learning workflow requires a certain instance type that is optimized to handle Machine Learning computations. Hence, this option is incorrect.

Reference:

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

Domain

Design Resilient Architectures

Question 28Skipped

Reporters at a news agency upload/download video files (about 500 megabytes each) to/from an Amazon S3 bucket as part of their daily work. As the agency has started offices in remote locations, it has resulted in poor latency for uploading and accessing data to/from the given Amazon S3 bucket. The agency wants to continue using a serverless storage solution such as Amazon S3 but wants to improve the performance.

As a solutions architect, which of the following solutions do you propose to address this issue? (Select two)

Correct selection

Use Amazon CloudFront distribution with origin as the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files

Create new Amazon S3 buckets in every region where the agency has a remote office, so that each office can maintain its storage for the media assets

Spin up Amazon EC2 instances in each region where the agency has a remote office. Create a daily job to transfer Amazon S3 data into Amazon EBS volumes attached to the Amazon EC2 instances

Move Amazon S3 data into Amazon Elastic File System (Amazon EFS) created in a US region, connect to Amazon EFS file system from Amazon EC2 instances in other AWS regions using an inter-region VPC peering connection

Correct selection

Enable Amazon S3 Transfer Acceleration (Amazon S3TA) for the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files

Overall explanation

Correct options:

Use Amazon CloudFront distribution with origin as the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files

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, within a developer-friendly environment. When an object from Amazon S3 that is set up with Amazon CloudFront CDN is requested, the request would come through the Edge Location transfer paths only for the first request. Thereafter, it would be served from the nearest edge location to the users until it expires. So in this way, you can speed up uploads as well as downloads for the video files.

Following is a good reference blog for a deep-dive:

https://aws.amazon.com/blogs/aws/amazon-cloudfront-content-uploads-post-put-other-methods/

Enable Amazon S3 Transfer Acceleration (Amazon S3TA) for the Amazon S3 bucket. This would speed up uploads as well as downloads for the video files

Amazon S3 Transfer Acceleration (Amazon S3TA) can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path. So this option is also correct.

Amazon S3TA:

> Amazon S3 Transfer Acceleration can speed up content transfers to and from Amazon S3 by as much as 50-500% for long-distance transfer of larger objects. Customers who have either web or mobile applications with widespread users or applications hosted far away from their S3 bucket can experience long and variable upload and download speeds over the Internet. S3 Transfer Acceleration (S3TA) reduces the variability in Internet routing, congestion and speeds that can affect transfers, and logically shortens the distance to S3 for remote applications. S3TA improves transfer performance by routing traffic through Amazon CloudFront's globally distributed Edge Locations and over AWS backbone networks, and by using network protocol optimizations. You can turn on S3TA with a few clicks in the S3 console, and test its benefits from your location with a speed comparison tool. With S3TA, you pay only for transfers that are accelerated.

via - https://aws.amazon.com/s3/transfer-acceleration/

Incorrect options:

Create new Amazon S3 buckets in every region where the agency has a remote office, so that each office can maintain its storage for the media assets - Creating new Amazon S3 buckets in every region is not an option, since the agency maintains centralized storage. Hence this option is incorrect.

Move Amazon S3 data into Amazon Elastic File System (Amazon EFS) created in a US region, connect to Amazon EFS file system from Amazon EC2 instances in other AWS regions using an inter-region VPC peering connection

Spin up Amazon EC2 instances in each region where the agency has a remote office. Create a daily job to transfer Amazon S3 data into Amazon EBS volumes attached to the Amazon EC2 instances

Both these options using Amazon EC2 instances are not correct for the given use-case, as the agency wants a serverless storage solution.

References:

https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/DownloadDistS3AndCustomOrigins.html

https://docs.aws.amazon.com/AmazonS3/latest/dev/transfer-acceleration.html

https://aws.amazon.com/s3/transfer-acceleration/

https://aws.amazon.com/blogs/aws/amazon-cloudfront-content-uploads-post-put-other-methods/

Domain

Design High-Performing Architectures

Question 29Skipped

A streaming solutions company is building a video streaming product by using an Application Load Balancer (ALB) that routes the requests to the underlying Amazon EC2 instances. The engineering team has noticed a peculiar pattern. The Application Load Balancer removes an instance from its pool of healthy instances whenever it is detected as unhealthy but the Auto Scaling group fails to kick-in and provision the replacement instance.

What could explain this anomaly?

Both the Auto Scaling group and Application Load Balancer are using ALB based health check

Correct answer

The Auto Scaling group is using Amazon EC2 based health check and the Application Load Balancer is using ALB based health check

Both the Auto Scaling group and Application Load Balancer are using Amazon EC2 based health check

The Auto Scaling group is using ALB based health check and the Application Load Balancer is using Amazon EC2 based health check

Overall explanation

Correct option:

The Auto Scaling group is using Amazon EC2 based health check and the Application Load Balancer is using ALB based health check

An Auto Scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for automatic scaling and management.

Auto Scaling Group Overview:

via - https://docs.aws.amazon.com/autoscaling/ec2/userguide/what-is-amazon-ec2-auto-scaling.html

Application Load Balancer automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and AWS Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

If the Auto Scaling group (ASG) is using EC2 as the health check type and the Application Load Balancer (ALB) is using its in-built health check, there may be a situation where the ALB health check fails because the health check pings fail to receive a response from the instance. At the same time, ASG health check can come back as successful because it is based on EC2 based health check. Therefore, in this scenario, the ALB will remove the instance from its inventory, however, the Auto Scaling Group will fail to provide the replacement instance. This can lead to the scaling issues mentioned in the problem statement.

Incorrect options:

The Auto Scaling group is using ALB based health check and the Application Load Balancer is using Amazon EC2 based health check - Application Load Balancer cannot use EC2 based health checks, so this option is incorrect.

Both the Auto Scaling group and Application Load Balancer are using ALB based health check - It is recommended to use ALB based health checks for both Auto Scaling group and Application Load Balancer. If both the Auto Scaling group and Application Load Balancer use ALB based health checks, then you will be able to avoid the scenario mentioned in the question.

Both the Auto Scaling group and Application Load Balancer are using Amazon EC2 based health check - Application Load Balancer cannot use EC2 based health checks, so this option is incorrect.

References:

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

https://docs.aws.amazon.com/autoscaling/ec2/userguide/health-checks-overview.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-add-elb-healthcheck.html

Domain

Design Resilient Architectures

Question 30Skipped

An e-commerce company uses Amazon Simple Queue Service (Amazon SQS) queues to decouple their application architecture. The engineering team has observed message processing failures for some customer orders.

As a solutions architect, which of the following solutions would you recommend for handling such message failures?

Correct answer

Use a dead-letter queue to handle message processing failures

Use long polling to handle message processing failures

Use a temporary queue to handle message processing failures

Use short polling to handle message processing failures

Overall explanation

Correct option:

Use a dead-letter queue to handle message processing failures

Dead-letter queues can be used by other queues (source queues) as a target for messages that can't be processed (consumed) successfully. Dead-letter queues are useful for debugging your application or messaging system because they let you isolate problematic messages to determine why their processing doesn't succeed. Sometimes, messages can’t be processed because of a variety of possible issues, such as when a user comments on a story but it remains unprocessed because the original story itself is deleted by the author while the comments were being posted. In such a case, the dead-letter queue can be used to handle message processing failures.

How do dead-letter queues work?:

How do dead-letter queues work?

Sometimes, messages can't be processed because of a variety of possible issues, such as erroneous conditions within the producer or consumer application or an unexpected state change that causes an issue with your application code. For example, if a user places a web order with a particular product ID, but the product ID is deleted, the web store's code fails and displays an error, and the message with the order request is sent to a dead-letter queue.

Occasionally, producers and consumers might fail to interpret aspects of the protocol that they use to communicate, causing message corruption or loss. Also, the consumer's hardware errors might corrupt message payload.

The *redrive policy* specifies the *source queue*, the *dead-letter queue*, and the conditions under which Amazon SQS moves messages from the former to the latter if the consumer of the source queue fails to process a message a specified number of times. When the `ReceiveCount` for a message exceeds the `maxReceiveCount` for a queue, Amazon SQS moves the message to a dead-letter queue (with its original message ID). For example, if the source queue has a redrive policy with `maxReceiveCount` set to 5, and the consumer of the source queue receives a message 6 times without ever deleting it, Amazon SQS moves the message to the dead-letter queue.

via - https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html

Incorrect options:

Use a temporary queue to handle message processing failures - The most common use case for temporary queues is the request-response messaging pattern (for example, processing a login request), where a requester creates a temporary queue for receiving each response message. To avoid creating an Amazon SQS queue for each response message, the Temporary Queue Client lets you create and delete multiple temporary queues without making any Amazon SQS API calls. Temporary queues cannot be used to handle message processing failures.

Use short polling to handle message processing failures

Use long polling to handle message processing failures

Amazon SQS provides short polling and long polling to receive messages from a queue. By default, queues use short polling. With short polling, Amazon SQS sends the response right away, even if the query found no messages. With long polling, Amazon SQS sends a response after it collects at least one available message, up to the maximum number of messages specified in the request. Amazon SQS sends an empty response only if the polling wait time expires. Neither short polling nor long polling can be used to handle message processing failures.

Reference:

https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html

Domain

Design Resilient Architectures

Question 31Skipped

Computer vision researchers at a university are trying to optimize the I/O bound processes for a proprietary algorithm running on Amazon EC2 instances. The ideal storage would facilitate high-performance IOPS when doing file processing in a temporary storage space before uploading the results back into Amazon S3.

As a solutions architect, which of the following AWS storage options would you recommend as the MOST performant as well as cost-optimal?

Use Amazon EC2 instances with Amazon EBS Throughput Optimized HDD (st1) as the storage option

Correct answer

Use Amazon EC2 instances with Instance Store as the storage option

Use Amazon EC2 instances with Amazon EBS General Purpose SSD (gp2) as the storage option

Use Amazon EC2 instances with Amazon EBS Provisioned IOPS SSD (io1) as the storage option

Overall explanation

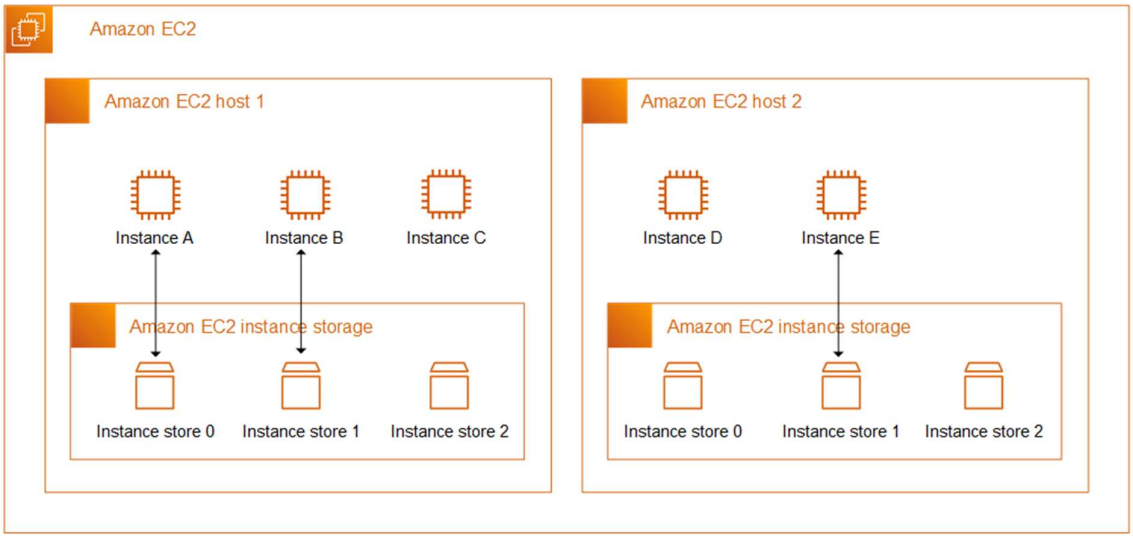
Correct option:

Use Amazon EC2 instances with Instance Store as the storage option

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for the temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content, or for data that is replicated across a fleet of instances, such as a load-balanced pool of web servers. Some instance types use NVMe or SATA-based solid-state drives (SSD) to deliver high random I/O performance. This is a good option when you need storage with very low latency, but you don't need the data to persist when the instance terminates or you can take advantage of fault-tolerant architectures.

As Instance Store delivers high random I/O performance, it can act as a temporary storage space, and these volumes are included as part of the instance's usage cost, therefore this is the correct option.

Amazon EC2 Instance Store:



via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/InstanceStorage.html

Incorrect options:

Use Amazon EC2 instances with Amazon EBS General Purpose SSD (gp2) as the storage option - 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. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver its provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB. Amazon EBS gp2 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the Amazon EC2 instance), therefore this option is not correct.

Use Amazon EC2 instances with Amazon EBS Provisioned IOPS SSD (io1) as the storage option - Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency. Unlike gp2, which uses a bucket and credit model to calculate performance, an io1 volume allows you to specify a consistent IOPS rate when you create the volume, and Amazon EBS delivers the provisioned performance 99.9 percent of the time. Amazon EBS io1 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the Amazon EC2 instance), therefore this option is not correct.

Use Amazon EC2 instances with Amazon EBS Throughput Optimized HDD (st1) as the storage option - Throughput Optimized HDD (st1) are low-cost HDD volumes designed for frequently accessed, throughput-intensive workloads such as Big data and Data warehouses. Amazon EBS st1 is persistent storage and costlier than Instance Stores (the cost of the storage volume is in addition to that of the Amazon EC2 instance), therefore this option is not correct.

References:

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

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

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

Domain

Design Cost-Optimized Architectures

Question 32Skipped

A pharmaceutical company is considering moving to AWS Cloud to accelerate the research and development process. Most of the daily workflows would be centered around running batch jobs on Amazon EC2 instances with storage on Amazon Elastic Block Store (Amazon EBS) volumes. The CTO is concerned about meeting HIPAA compliance norms for sensitive data stored on Amazon EBS.

Which of the following options outline the correct capabilities of an encrypted Amazon EBS volume? (Select three)

Data moving between the volume and the instance is NOT encrypted

Correct selection

Data moving between the volume and the instance is encrypted

Correct selection

Data at rest inside the volume is encrypted

Correct selection

Any snapshot created from the volume is encrypted

Any snapshot created from the volume is NOT encrypted

Data at rest inside the volume is NOT encrypted

Overall explanation

Correct options:

Data at rest inside the volume is encrypted

Any snapshot created from the volume is encrypted

Data moving between the volume and the instance is encrypted

Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with Amazon EC2 instances. When you create an encrypted Amazon EBS volume and attach it to a supported instance type, data stored at rest on the volume, data moving between the volume and the instance, snapshots created from the volume and volumes created from those snapshots are all encrypted. It uses AWS Key Management Service (AWS KMS) customer master keys (CMK) when creating encrypted volumes and snapshots. Encryption operations occur on the servers that host Amazon EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached Amazon EBS storage.

Therefore, the incorrect options are:

Data moving between the volume and the instance is NOT encrypted

Any snapshot created from the volume is NOT encrypted

Data at rest inside the volume is NOT encrypted

Reference:

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

Domain

Design Secure Architectures

Question 33Skipped

A developer in your team has set up a classic 3 tier architecture composed of an Application Load Balancer, an Auto Scaling group managing a fleet of Amazon EC2 instances, and an Amazon Aurora database. As a Solutions Architect, you would like to adhere to the security pillar of the well-architected framework.

How do you configure the security group of the Aurora database to only allow traffic coming from the Amazon EC2 instances?

Add a rule authorizing the Amazon Aurora security group

Correct answer

Add a rule authorizing the Amazon EC2 security group

Add a rule authorizing the Auto Scaling group subnets CIDR

Add a rule authorizing the Elastic Load Balancing security group

Overall explanation

Correct option:

Add a rule authorizing the Amazon EC2 security group

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you can specify one or more security groups; otherwise, we use the default security group. You can add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group. When we decide whether to allow traffic to reach an instance, we evaluate all the rules from all the security groups that are associated with the instance.

The following are the characteristics of security group rules:

By default, security groups allow all outbound traffic.

Security group rules are always permissive; you can't create rules that deny access.

Security groups are stateful.

For the given scenario, the Amazon EC2 instances that are part of the Auto Scaling Group are the ones accessing the database layer. The correct response is to add a rule to the security group attached to Aurora authorizing the Amazon EC2 instance's security group.

Incorrect options:

Add a rule authorizing the Amazon Aurora security group - Adding a rule, authorizing the Aurora security group, is just a distractor. Since it has no bearing on traffic allowed from the Amazon EC2 instances.

Add a rule authorizing the Auto Scaling group subnets CIDR - Authorizing the entire CIDR of the ASG's subnets is overkill and would allow non-Auto Scaling Group instances, access Aurora if they were part of the same CIDR.

Add a rule authorizing the Elastic Load Balancing security group - Adding a rule authorizing the ELB security group would dilute the security for the Aurora databases because only the Amazon EC2 instances that are part of the Auto Scaling Group are the ones accessing the database layer. Therefore, it is not the correct option.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html

Domain

Design Secure Architectures

Question 34Skipped

A leading video streaming provider is migrating to AWS Cloud infrastructure for delivering its content to users across the world. The company wants to make sure that the solution supports at least a million requests per second for its Amazon EC2 server farm.

As a solutions architect, which type of Elastic Load Balancing would you recommend as part of the solution stack?

Classic Load Balancer

Application Load Balancer

Correct answer

Network Load Balancer

Infrastructure Load Balancer

Overall explanation

Correct option:

Network Load Balancer

Network Load Balancer is best suited for use-cases involving low latency and high throughput workloads that involve scaling to millions of requests per second. Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, microservices, and containers – within Amazon Virtual Private Cloud (Amazon VPC) based on IP protocol data.

Incorrect options:

Application Load Balancer - Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers, IP addresses, and Lambda functions based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS

traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer is not a good fit for the low latency and high throughput scenario mentioned in the given use-case.

Classic Load Balancer - Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network. Classic Load Balancer is not a good fit for the low latency and high throughput scenario mentioned in the given use-case.

Infrastructure Load Balancer - There is no such thing as Infrastructure Load Balancer and this option just acts as a distractor.

Reference:

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

Domain

Design Resilient Architectures

Question 35Skipped

Your firm has implemented a multi-tiered networking structure within the VPC - with two public and two private subnets. The public subnets are used to deploy the Application Load Balancers, while the two private subnets are used to deploy the application on Amazon EC2 instances. The development team wants the Amazon EC2 instances to have access to the internet. The solution has to be fully managed by AWS and needs to work over IPv4.

What will you recommend?

Egress-Only Internet Gateways deployed in your private subnet

Internet Gateways deployed in your private subnet

Correct answer

NAT Gateways deployed in your public subnet

NAT Instances deployed in your public subnet

Overall explanation

Correct option:

NAT Gateways deployed in your public subnet

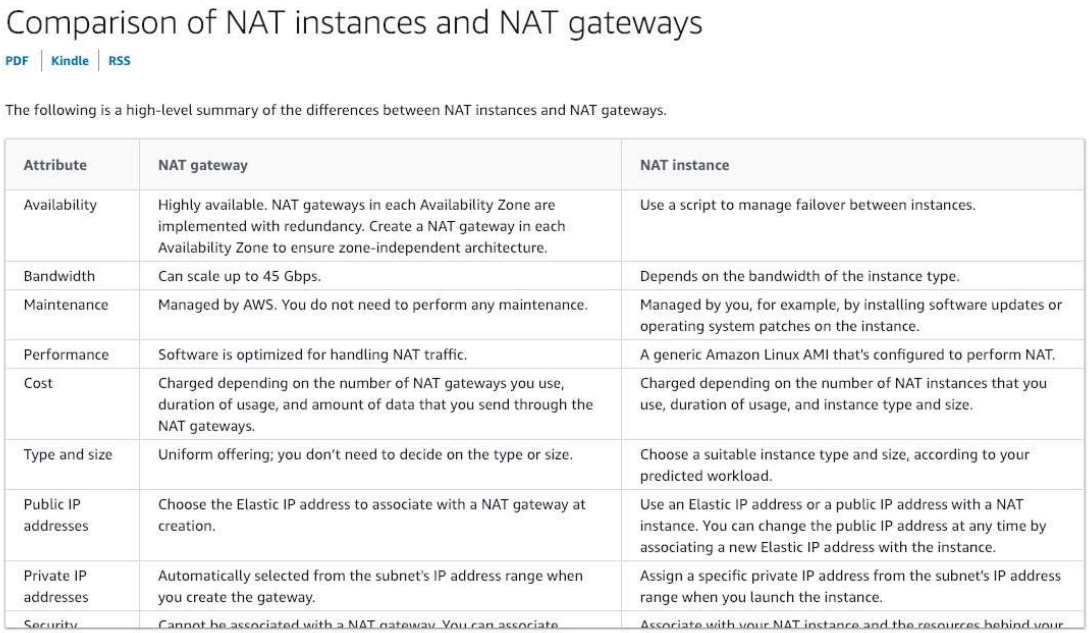
You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances. A NAT gateway has the following characteristics and limitations:

1. A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.
2. You can associate exactly one Elastic IP address with a NAT gateway.

3. A NAT gateway supports the following protocols: TCP, UDP, and ICMP.
4. You cannot associate a security group with a NAT gateway.
5. You can use a network access control list (network ACL) to control the traffic to and from the subnet in which the NAT gateway is located.
6. A NAT gateway can support up to 55,000 simultaneous connections to each unique destination.

Therefore you must use a NAT Gateway in your public subnet in order to provide internet access to your instances in your private subnets. You are charged for creating and using a NAT gateway in your account. NAT gateway hourly usage and data processing rates apply.

Comparison of NAT instances and NAT gateways:



Comparison of NAT instances and NAT gateways

PDF | Kindle | RSS

The following is a high-level summary of the differences between NAT instances and NAT gateways.

Attribute	NAT gateway	NAT instance
Availability	Highly available. NAT gateways in each Availability Zone are implemented with redundancy. Create a NAT gateway in each Availability Zone to ensure zone-independent architecture.	Use a script to manage failover between instances.
Bandwidth	Can scale up to 45 Gbps.	Depends on the bandwidth of the instance type.
Maintenance	Managed by AWS. You do not need to perform any maintenance.	Managed by you, for example, by installing software updates or operating system patches on the instance.
Performance	Software is optimized for handling NAT traffic.	A generic Amazon Linux AMI that's configured to perform NAT.
Cost	Charged depending on the number of NAT gateways you use, duration of usage, and amount of data that you send through the NAT gateways.	Charged depending on the number of NAT instances that you use, duration of usage, and instance type and size.
Type and size	Uniform offering; you don't need to decide on the type or size.	Choose a suitable instance type and size, according to your predicted workload.
Public IP addresses	Choose the Elastic IP address to associate with a NAT gateway at creation.	Use an Elastic IP address or a public IP address with a NAT instance. You can change the public IP address at any time by associating a new Elastic IP address with the instance.
Private IP addresses	Automatically selected from the subnet's IP address range when you create the gateway.	Assign a specific private IP address from the subnet's IP address range when you launch the instance.
Security	Cannot be associated with a NAT gateway. You can associate	Associate with your NAT instance and the resources behind your

via - https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-comparison.html

Incorrect options:

NAT Instances deployed in your public subnet - You can use a network address translation (NAT) instance in a public subnet in your VPC to enable instances in the private subnet to initiate outbound IPv4 traffic to the Internet or other AWS services, but prevent the instances from receiving inbound traffic initiated by someone on the Internet. Amazon provides Amazon Linux AMIs that are configured to run as NAT instances. These AMIs include the string amzn-ami-vpc-nat in their names, so you can search for them in the Amazon EC2 console. NAT Instances would work but won't scale and you would have to manage them (as they're nothing but Amazon EC2 instances).

Internet Gateways deployed in your private subnet - An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It, therefore, imposes no availability risks or bandwidth

constraints on your network traffic. Internet Gateways must be deployed in a public subnet, hence not an option here.

Egress-Only Internet Gateways deployed in your private subnet - An Egress-Only Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows outbound communication over IPv6 from instances in your VPC to the Internet, and prevents the Internet from initiating an IPv6 connection with your instances. Egress-Only Internet Gateways are for IPv6, not IPv4. Therefore, this option is incorrect.

References:

https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html

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

Domain

Design High-Performing Architectures

Question 36Skipped

A company's cloud architect has set up a solution that uses Amazon Route 53 to configure the DNS records for the primary website with the domain pointing to the Application Load Balancer (ALB). The company wants a solution where users will be directed to a static error page, configured as a backup, in case of unavailability of the primary website.

Which configuration will meet the company's requirements, while keeping the changes to a bare minimum?

Set up Amazon Route 53 active-active type of failover routing policy. If Amazon Route 53 health check determines the Application Load Balancer endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket

Use Amazon Route 53 Latency-based routing. Create a latency record to point to the Amazon S3 bucket that holds the error page to be displayed

Correct answer

Set up Amazon Route 53 active-passive type of failover routing policy. If Amazon Route 53 health check determines the Application Load Balancer endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket

Use Amazon Route 53 Weighted routing to give minimum weight to Amazon S3 bucket that holds the error page to be displayed. In case of primary failure, the requests get routed to the error page

Overall explanation

Correct option:

Set up Amazon Route 53 active-passive type of failover routing policy. If Amazon Route 53 health check determines the Application Load Balancer endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket

Use an active-passive failover configuration when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group

of resources to be on standby in case all the primary resources become unavailable. When responding to queries, Amazon Route 53 includes only healthy primary resources. If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries.

Incorrect options:

Set up Amazon Route 53 active-active type of failover routing policy. If Amazon Route 53 health check determines the Application Load Balancer endpoint as unhealthy, the traffic will be diverted to a static error page, hosted on Amazon S3 bucket - This option has been added as a distractor as there is no such thing as an active-active failover routing policy in Amazon Route 53. You can configure active-active failover using any routing policy (or combination of routing policies) other than failover routing policy and you configure active-passive failover only using the failover routing policy. In active-active failover configuration, all the records that have the same name, the same type (such as A or AAAA), and the same routing policy (such as weighted or latency) are active unless Amazon Route 53 considers them unhealthy. Amazon Route 53 can respond to a DNS query using any healthy record.

Use Amazon Route 53 Latency-based routing. Create a latency record to point to the Amazon S3 bucket that holds the error page to be displayed - If your application is hosted in multiple AWS Regions, you can improve performance for your users by serving their requests from the AWS Region that provides the lowest latency - this is Latency-based routing and is not helpful for the current use case.

Use Amazon Route 53 Weighted routing to give minimum weight to Amazon S3 bucket that holds the error page to be displayed. In case of primary failure, the requests get routed to the error page - Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of the software. This is not useful for the current use case.

References:

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover-types.html#dns-failover-types-active-passive

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-latency

Domain

Design Resilient Architectures

Question 37Skipped

A retail company maintains an AWS Direct Connect connection to AWS and has recently migrated its data warehouse to AWS. The data analysts at the company query the data warehouse using a visualization tool. The average size of a query returned by the data warehouse is 60 megabytes and the query responses returned by the data warehouse are not cached in the visualization tool. Each webpage returned by the visualization tool is approximately 600 kilobytes.

Which of the following options offers the LOWEST data transfer egress cost for the company?

Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over the internet at a location in the same region

Deploy the visualization tool on-premises. Query the data warehouse directly over an AWS Direct Connect connection at a location in the same AWS region

Deploy the visualization tool on-premises. Query the data warehouse over the internet at a location in the same AWS region

Correct answer

Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over a Direct Connect connection at a location in the same region

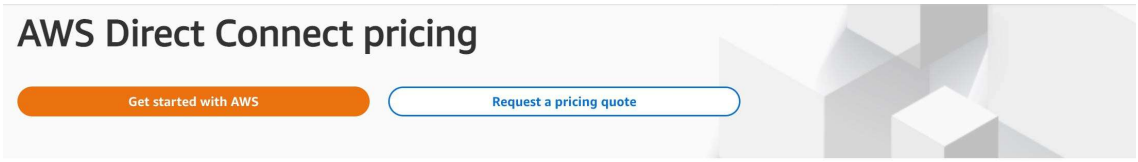
Overall explanation

Correct option:

Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over a Direct Connect connection at a location in the same region

AWS Direct Connect is a networking service that provides an alternative to using the internet to connect to AWS. Using AWS Direct Connect, data that would have previously been transported over the internet is delivered through a private network connection between your on-premises data center and AWS.

For the given use case, the main pricing parameter while using the AWS Direct Connect connection is the Data Transfer Out (DTO) from AWS to the on-premises data center. DTO refers to the cumulative network traffic that is sent through AWS Direct Connect to destinations outside of AWS. This is charged per gigabyte (GB), and unlike capacity measurements, DTO refers to the amount of data transferred, not the speed.



AWS Direct Connect is a cloud service that links your network directly to AWS to deliver consistent, low-latency performance. With AWS Direct Connect, you pay only for what you use and there is no minimum fee. There are no setup charges, and you may cancel at any time. However, services provided by your AWS Direct Connect Delivery Partners or other local service provider may have other terms that apply.

Once you have linked your locations to AWS Direct Connect, you can send data between them using SiteLink. When using SiteLink, data travels over the shortest path between locations. The SiteLink feature is off by default and can be turned on or off at any time.

Pricing components

When connecting to resources running in any AWS Region (such as an Amazon Virtual Private Cloud or AWS Transit Gateway), there are three factors that determine pricing: capacity, port hours, and data transfer out (DTO).

Capacity is the maximum rate that data can be transferred through a network connection. The capacity of AWS Direct Connect connections are measured in megabit per second (Mbps) or gigabit per second (Gbps). One gigabit per second, or 1 Gbps, is equal to 1,000 megabits per second (1,000 Mbps).

Port hours measure the time that a port is provisioned for your use with AWS, or an AWS Direct Connect Delivery Partner's, networking equipment inside an AWS Direct Connect location. Even when no data is passing through the port, you are charged for port hours. Port hour pricing is determined by the connection type: dedicated or hosted.

> **Dedicated connections** are physical connections between your network port and an AWS network port inside an AWS Direct Connect location. Dedicated port hours are billed as long as that port is provisioned for your use. You request a dedicated connection through the AWS Direct Connect section of the AWS Management Console.
>
> **Hosted connections** are logical connections that an AWS Direct Connect Delivery Partner provisions on your behalf. When using hosted connections, you connect to the AWS network using one of the partner's ports. You request a hosted connection by contacting an AWS Direct Connect Delivery Partner directly.

Data transfer out (DTO) refers to the cumulative network traffic that is sent through AWS Direct Connect to destinations outside of AWS. This is charged per gigabyte (GB), and unlike capacity measurements, DTO refers to the amount of data transferred, not the speed. When calculating DTO, exact pricing depends on the AWS Region or AWS Local Zone, and the AWS Direct Connect location, you are using (see tables below).

Data transfer in refers to network traffic that is sent into AWS from outside, over AWS Direct Connect. AWS Direct Connect data transfer in is charged at 0.00 USD per GB in all locations.

Except as otherwise noted, our prices are exclusive of applicable taxes and duties, including VAT and applicable sales tax. For customers with a Japanese billing address, use of the Asia Pacific (Tokyo) Region is subject to Japanese Consumption Tax. Learn more.

via - https://aws.amazon.com/directconnect/pricing/

Each query response is 60 megabytes in size and each webpage for the visualization tool is 600 kilobytes in size. If you deploy the visualization tool in the same AWS region as the data warehouse, then you only need to pay for the 600 kilobytes of DTO charges for the webpage. Therefore this option is correct.

However, if you deploy the visualization tool on-premises, then you need to pay for the 60 MB of DTO charges for the query response from the data warehouse to the visualization tool.

Incorrect options:

Deploy the visualization tool in the same AWS region as the data warehouse. Access the visualization tool over the internet at a location in the same region

Deploy the visualization tool on-premises. Query the data warehouse over the internet at a location in the same AWS region

Data transfer pricing over AWS Direct Connect is lower than data transfer pricing over the internet, so both of these options are incorrect.

Deploy the visualization tool on-premises. Query the data warehouse directly over an AWS Direct Connect connection at a location in the same AWS region - As mentioned in the explanation above, if you deploy the visualization tool on-premises, then you need to pay for the 60 megabytes of DTO charges for the query response from the data warehouse to the visualization tool. So this option is incorrect.

References:

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

https://aws.amazon.com/getting-started/hands-on/connect-data-center-to-aws/services-costs/

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

Domain

Design Cost-Optimized Architectures

Question 38Skipped

The engineering team at a weather tracking company wants to enhance the performance of its relational database and is looking for a caching solution that supports geospatial data.

As a solutions architect, which of the following solutions will you suggest?

Correct answer

Use Amazon ElastiCache for Redis

Use AWS Global Accelerator

Use Amazon ElastiCache for Memcached

Use Amazon DynamoDB Accelerator (DAX)

Overall explanation

Correct option:

Use Amazon ElastiCache for Redis

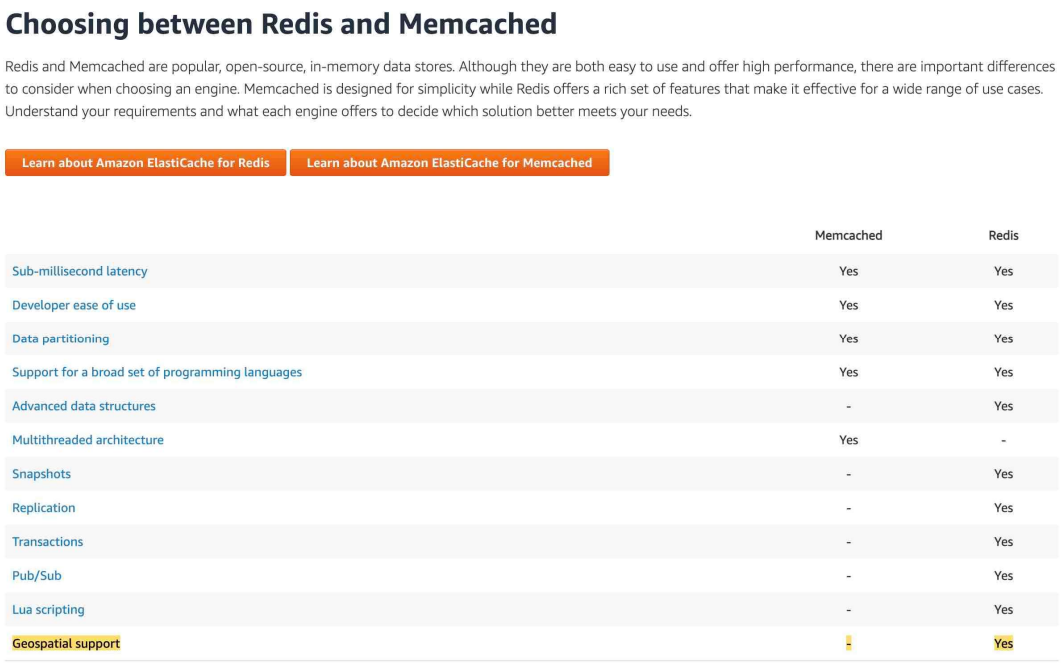
Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. Redis, which stands for Remote Dictionary Server, is a fast, open-source, in-memory key-value data store for use as a database, cache, message broker, and queue. Redis now delivers sub-millisecond response times enabling millions of requests per second for real-time applications in Gaming, Ad-Tech, Financial Services, Healthcare, and IoT. Redis is a popular choice for caching, session management, gaming, leaderboards, real-time analytics, geospatial, ride-hailing, chat/messaging, media streaming, and pub/sub apps.

All Redis data resides in the server's main memory, in contrast to databases such as PostgreSQL, Cassandra, MongoDB and others that store most data on disk or on SSDs. In comparison to traditional disk based databases where most operations require a roundtrip to disk, in-memory data stores such as Redis don't suffer the same penalty. They can therefore support an order of magnitude more operations and faster response times. The result is – blazing fast performance with average read or write operations taking less than a millisecond and support for millions of operations per second.

Redis has purpose-built commands for working with real-time geospatial data at scale. You can perform operations like finding the distance between two elements (for example people or places) and finding all elements within a given distance of a point.

Incorrect options:

Use Amazon ElastiCache for Memcached - Both Redis and MemCached are in-memory, open-source data stores. Memcached, a high-performance distributed memory cache service, is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Memcached does not offer support for geospatial data.



Choosing between Redis and Memcached

Redis and Memcached are popular, open-source, in-memory data stores. Although they are both easy to use and offer high performance, there are important differences to consider when choosing an engine. Memcached is designed for simplicity while Redis offers a rich set of features that make it effective for a wide range of use cases. Understand your requirements and what each engine offers to decide which solution better meets your needs.

Learn about Amazon ElastiCache for Redis | Learn about Amazon ElastiCache for Memcached

	Memcached	Redis
Sub-millisecond latency	Yes	Yes
Developer ease of use	Yes	Yes
Data partitioning	Yes	Yes
Support for a broad set of programming languages	Yes	Yes
Advanced data structures	-	Yes
Multithreaded architecture	Yes	-
Snapshots	-	Yes
Replication	-	Yes
Transactions	-	Yes
Pub/Sub	-	Yes
Lua scripting	-	Yes
Geospatial support	-	Yes

via - https://aws.amazon.com/elasticache/redis-vs-memcached/

Use Amazon DynamoDB Accelerator (DAX) - Amazon DynamoDB Accelerator (DAX) is a fully managed, highly available, in-memory cache for Amazon DynamoDB. DAX does not support relational databases.

Use AWS Global Accelerator - AWS Global Accelerator is a networking service that helps you improve the availability and performance of the applications that you offer to your global users. This option has been added as a distractor, it has nothing to do with database caching.

Reference:

https://aws.amazon.com/elasticache/redis-vs-memcached/

Domain

Design High-Performing Architectures

Question 39Skipped

A company wants to ensure high availability for its Amazon RDS database. The development team wants to opt for Multi-AZ deployment and they would like to understand what happens when the primary instance of the Multi-AZ configuration goes down.

As a Solutions Architect, which of the following will you identify as the outcome of the scenario?

An email will be sent to the System Administrator asking for manual intervention

Correct answer

The CNAME record will be updated to point to the standby database

The application will be down until the primary database has recovered itself

The URL to access the database will change to the standby database

Overall explanation

Correct option:

The CNAME record will be updated to point to the standby database

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 MariaDB, MySQL, Oracle, and PostgreSQL DB instances use Amazon's failover technology. SQL Server DB instances use SQL Server Database Mirroring (DBM) or Always On Availability Groups (AGs).

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.

Failover is automatically handled by Amazon RDS so that you can resume database operations as quickly as possible without administrative intervention. 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. Multi-AZ means the URL is the same, the failover is automated, and the CNAME will automatically be updated to point to the standby database.

Incorrect options:

The URL to access the database will change to the standby database - As discussed above, URL remains the same.

An email will be sent to the System Administrator asking for manual intervention - This option is incorrect and it has been added as a distractor.

The application will be down until the primary database has recovered itself - This option is incorrect and it has been added as a distractor.

References:

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

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

Domain

Design Resilient Architectures

Question 40Skipped

A global media company uses a fleet of Amazon EC2 instances (behind an Application Load Balancer) to power its video streaming application. To improve the performance of the application, the engineering team has also created an Amazon CloudFront distribution with the Application Load Balancer as the custom origin. The security team at the company has noticed a spike in the number and types of SQL injection and cross-site scripting attack vectors on the application.

As a solutions architect, which of the following solutions would you recommend as the MOST effective in countering these malicious attacks?

Correct answer

Use AWS Web Application Firewall (AWS WAF) with Amazon CloudFront distribution

Use AWS Firewall Manager with CloudFront distribution

Use Amazon Route 53 with Amazon CloudFront distribution

Use AWS Security Hub with Amazon CloudFront distribution

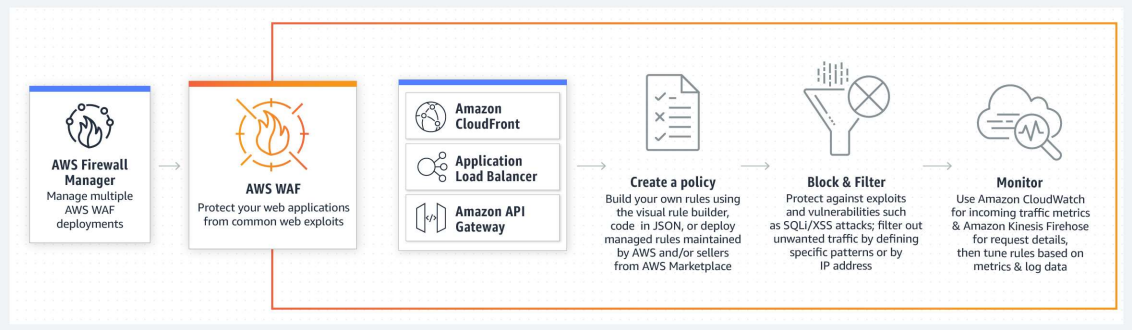
Overall explanation

Correct option:

Use AWS Web Application Firewall (AWS WAF) with Amazon CloudFront distribution

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns, such as SQL injection or cross-site scripting, and rules that filter out specific traffic patterns you define.

How AWS WAF Works:



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

A web access control list (web ACL) gives you fine-grained control over the web requests that your Amazon CloudFront distribution, Amazon API Gateway API, or Application Load Balancer responds to.

When you create a web ACL, you can specify one or more Amazon CloudFront distributions that you want AWS WAF to inspect. AWS WAF starts to allow, block, or count web requests for those distributions based on the conditions that you identify in the web ACL. Therefore, combining

AWS WAF with Amazon CloudFront can prevent SQL injection and cross-site scripting attacks. So this is the correct option.

Incorrect options:

Use Amazon Route 53 with Amazon CloudFront distribution - Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. You cannot use Route 53 to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

Use AWS Security Hub with Amazon CloudFront distribution - AWS Security Hub gives you a comprehensive view of your high-priority security alerts and security posture across your AWS accounts. With Security Hub, you have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, Amazon Macie, AWS Identity and Access Management (IAM) Access Analyzer, and AWS Firewall Manager, as well as from AWS Partner solutions. You cannot use Security Hub to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

Use AWS Firewall Manager with CloudFront distribution - AWS Firewall Manager is a security management service that allows you to centrally configure and manage firewall rules across your accounts and applications in AWS Organization. You cannot use AWS Firewall Manager to prevent SQL injection and cross-site scripting attacks. So this option is incorrect.

References:

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

https://docs.aws.amazon.com/waf/latest/developerguide/web-acl.html

https://docs.aws.amazon.com/waf/latest/developerguide/cloudfront-features.html

Domain

Design Secure Architectures

Question 41Skipped

A company is looking for a technology that allows its mobile app users to connect through a Google login and have the capability to turn on AWS Multi-Factor Authentication (AWS MFA) to have maximum security. Ideally, the solution should be fully managed by AWS.

Which technology do you recommend for managing the users' accounts?

AWS Identity and Access Management (AWS IAM)

Write an AWS Lambda function with Auth0 3rd party integration

Correct answer

Amazon Cognito

Enable the AWS Google Login Service

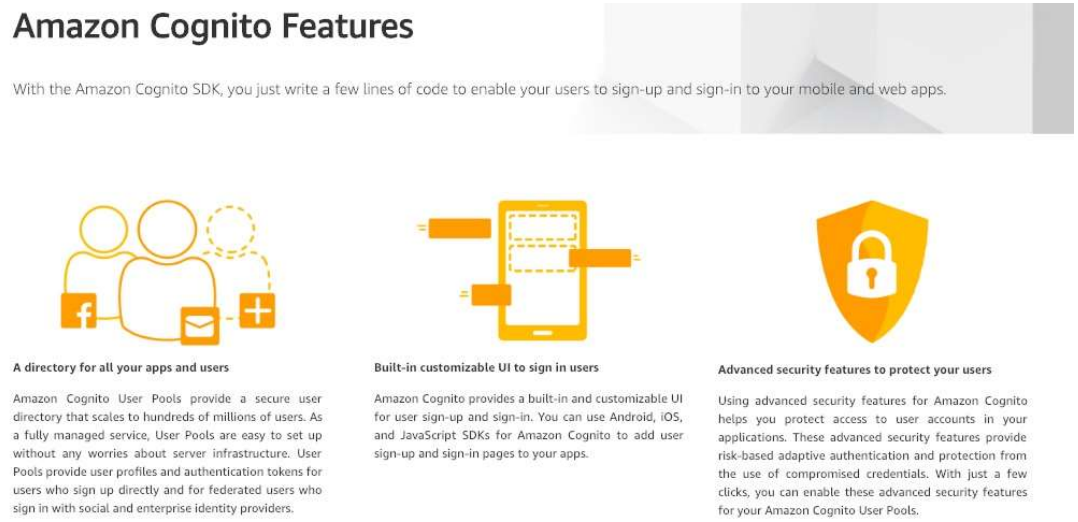
Overall explanation

Correct option:

Amazon Cognito

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0. Here Cognito is the best technology choice for managing mobile user accounts.

Amazon Cognito Features:



via - https://aws.amazon.com/cognito/details/

Incorrect options:

Write an AWS Lambda function with Auth0 3rd party integration - AWS Lambda lets you run code without provisioning or managing servers. You pay only for the compute time you consume. Using Lambda would require code maintenance for user management functionality, therefore this option is ruled out.

AWS Identity and Access Management (AWS IAM) - AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources. IAM cannot be used to manage mobile user accounts.

Enable the AWS Google Login Service - There is no such thing as AWS Google Login service. This option is just added as a distractor.

Reference:

https://aws.amazon.com/cognito/

Domain

Design High-Performing Architectures

Question 42Skipped

A company needs a massive PostgreSQL database and the engineering team would like to retain control over managing the patches, version upgrades for the database, and consistent performance with high IOPS. The team wants to install the database on an Amazon EC2 instance with the optimal storage type on the attached Amazon EBS volume.

As a solutions architect, which of the following configurations would you suggest to the engineering team?

Amazon EC2 with Amazon EBS volume of cold HDD (sc1) type

Amazon EC2 with Amazon EBS volume of General Purpose SSD (gp2) type

Amazon EC2 with Amazon EBS volume of Throughput Optimized HDD (st1) type

Correct answer

Amazon EC2 with Amazon EBS volume of Provisioned IOPS SSD (io1) type

Overall explanation

Correct option:

Amazon EC2 with Amazon EBS volume of Provisioned IOPS SSD (io1) type

Amazon EBS provides the following volume types, which differ in performance characteristics and price so that you can tailor your storage performance and cost to the needs of your applications.

The volumes types fall into two categories:

SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS

HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS

Provision IOPS type supports critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume. Examples are large database workloads, such as: MongoDB Cassandra Microsoft SQL Server MySQL PostgreSQL Oracle

Therefore, Amazon EC2 with Amazon EBS volume of Provisioned IOPS SSD (io1) type is the right fit for the given use-case.

Please see this detailed overview of the volume types for Amazon EBS volumes.

Volume characteristics

The following table describes the use cases and performance characteristics for each volume type. The default volume type is General Purpose SSD (gp2).

	Solid-state drives (SSD)		Hard disk drives (HDD)	
Volume type	General Purpose SSD (gp2)	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use cases	• 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	• Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume

via - https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

Incorrect options:

Amazon EC2 with Amazon EBS volume of General Purpose SSD (gp2) type

Amazon EC2 with Amazon EBS volume of Throughput Optimized HDD (st1) type

Amazon EC2 with Amazon EBS volume of cold HDD (sc1) type

Per the explanation in the detailed overview provided above, these three options are incorrect.

Reference:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volume-types.html

Domain

Design High-Performing Architectures

Question 43Skipped

The content division at a digital media agency has an application that generates a large number of files on Amazon S3, each approximately 10 megabytes in size. The agency mandates that the files be stored for 5 years before they can be deleted. The files are frequently accessed in the first 30 days of the object creation but are rarely accessed after the first 30 days. The files contain critical business data that is not easy to reproduce, therefore, immediate accessibility is always required.

Which solution is the MOST cost-effective for the given use case?

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Archive the files to Amazon S3 Glacier Deep Archive 5 years after object creation

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 One Zone-IA 30 days after object creation. Delete the files 5 years after object creation

Correct answer

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Delete the files 5 years after object creation

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Glacier Flexible Retrieval 30 days after object creation. Delete the files 5 years after object creation

Overall explanation

Correct option:

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Delete the files 5 years after object creation

Amazon S3 Standard-IA class is for data that is accessed less frequently but requires rapid access when needed. Amazon S3 Standard-IA offers the high durability, high throughput, and low latency of S3 Standard, with a low per gigabyte storage price and per GB retrieval charge.

Performance across the S3 Storage Classes

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128 KB	128 KB	128 KB	40 KB	40 KB
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval charge	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	minutes or hours	hours
Storage type	Object	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes	Yes

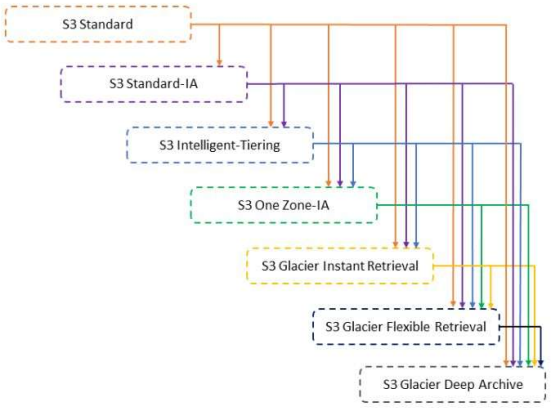
via - https://aws.amazon.com/s3/storage-classes/

For the given use case, you can set up an Amazon S3 lifecycle configuration and create a transition action to move objects from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. You can set up an expiration action to delete the object 5 years after object creation.

Supported transitions and related constraints

In an S3 Lifecycle configuration, you can define rules to transition objects from one storage class to another to save on storage costs. When you don't know the access patterns of your objects, or if your access patterns are changing over time, you can transition the objects to the S3 Intelligent-Tiering storage class for automatic cost savings. For information about storage classes, see Using Amazon S3 storage classes.

Amazon S3 supports a waterfall model for transitioning between storage classes, as shown in the following diagram.



via - https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html

Managing your storage lifecycle

PDF | RSS

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their *Amazon S3 Lifecycle*. An *S3 Lifecycle configuration* is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

- **Transition actions** – These actions define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after creating them, or archive objects to the S3 Glacier Flexible Retrieval storage class one year after creating them. For more information, see Using Amazon S3 storage classes.

  There are costs associated with lifecycle transition requests. For pricing information, see Amazon S3 pricing.
- **Expiration actions** – These actions define when objects expire. Amazon S3 deletes expired objects on your behalf.

  Lifecycle expiration costs depend on when you choose to expire objects. For more information, see Expiring objects.

If there is any delay between when an object becomes eligible for a lifecycle action and when Amazon S3 transfers or expires your object, billing changes are applied as soon as the object becomes eligible for the lifecycle action. For example, if an object is scheduled to expire and Amazon S3 does not immediately expire the object, you won't be charged for storage after the expiration time. The one exception to this behavior is if you have a lifecycle rule to transition to the S3 Intelligent-Tiering storage class. In that case, billing changes do not occur until the object has transitioned to S3 Intelligent-Tiering.

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

Incorrect options:

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Glacier Flexible Retrieval 30 days after object creation. Delete the files 5 years after object creation - Amazon S3 Glacier Flexible Retrieval storage class has the best case retrieval time of the order of minutes, so this option is incorrect for the given requirement.

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 Standard-IA 30 days after object creation. Archive the files to Amazon S3 Glacier Deep Archive 5 years after object creation - The files can simply be deleted 5 years after object creation instead of archiving the files to Amazon S3 Glacier Deep Archive. There is no need to incur the cost of archival.

Set up an Amazon S3 bucket lifecycle policy to move files from Amazon S3 Standard to Amazon S3 One Zone-IA 30 days after object creation. Delete the files 5 years after object creation - Unlike other Amazon S3 Storage Classes which store data in a minimum of three Availability Zones (AZs), Amazon S3 One Zone-IA stores data in a single AZ and costs 20% less than Amazon S3 Standard-IA. Amazon S3 One Zone-IA is a good choice for storing secondary backup copies of on-premises data or easily re-creatable data. The given scenario clearly states that the business-critical data is not easy to reproduce, so this option is incorrect.

References:

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

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

https://docs.aws.amazon.com/AmazonS3/latest/userguide/lifecycle-transition-general-considerations.html

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

Domain

Design Cost-Optimized Architectures

Question 44Skipped

A startup has created a cost-effective backup solution in another AWS Region. The application is running in warm standby mode and has Application Load Balancer (ALB) to support it from the front. The current failover process is manual and requires updating the DNS alias record to point to the secondary Application Load Balancer in another Region in case of failure of the primary Application Load Balancer.

As a Solutions Architect, what will you recommend to automate the failover process?

Enable an Amazon EC2 instance health check

Enable an ALB health check

Configure AWS Trusted Advisor to check on unhealthy instances

Correct answer

Enable an Amazon Route 53 health check

Overall explanation

Correct option:

Enable an Amazon Route 53 health check

Determining the health of an ELB endpoint is more complex than health checking a single IP address. For example, what if your application is running fine on Amazon EC2, but the load balancer itself isn't reachable? Or if your load balancer and your Amazon EC2 instances are working correctly, but a bug in your code causes your application to crash? Or how about if the Amazon EC2 instances in one Availability Zone of a multi-AZ ELB are experiencing problems?

Amazon Route 53 DNS Failover handles all of these failure scenarios by integrating with ELB behind the scenes. Once enabled, Route 53 automatically configures and manages health

checks for individual ELB nodes. Amazon Route 53 also takes advantage of the Amazon EC2 instance health checking that ELB performs (information on configuring your ELB health checks is available here). By combining the results of health checks of your Amazon EC2 instances and your ELBs, Amazon Route 53 DNS Failover can evaluate the health of the load balancer and the health of the application running on the Amazon EC2 instances behind it. In other words, if any part of the stack goes down, Amazon Route 53 detects the failure and routes traffic away from the failed endpoint.

Using Amazon Route 53 DNS Failover, you can run your primary application simultaneously in multiple AWS regions around the world and failover across regions. Your end-users will be routed to the closest (by latency), healthy region for your application. Amazon Route 53 automatically removes from service any region where your application is unavailable - it will pull an endpoint out of service if there is region-wide connectivity or operational issue, if your application goes down in that region, or if your ELB or Amazon EC2 instances go down in that region.

Incorrect options:

Enable an ALB health check - ELB health check verifies that a specified TCP port on an instance is accepting connections or a specified page has returned an error code of 200. It is not useful for the given failover scenario.

Enable an Amazon EC2 instance health check - Instance status checks monitor the software and network configuration of your instance. It is not intelligent enough to understand if the application on the instance is working correctly. Hence, this is not the right choice for the given use-case.

Configure AWS Trusted Advisor to check on unhealthy instances - AWS Trusted Advisor examines the health check configuration for Auto Scaling groups. If Elastic Load Balancing is being used for an Auto Scaling group, the recommended configuration is to enable an Elastic Load Balancing health check. AWS Trusted Advisor recommends certain configuration changes by comparing your system configurations to AWS Best practices. It cannot handle a failover the way Amazon Route 53 does.

References:

https://aws.amazon.com/blogs/aws/amazon-route-53-elb-integration-dns-failover/

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/monitoring-system-instance-status-check.html

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/target-group-health-checks.html

https://aws.amazon.com/premiumsupport/technology/trusted-advisor/best-practice-checklist/

Domain

Design Resilient Architectures

Question 45Skipped

You have just terminated an instance in the us-west-1a Availability Zone (AZ). The attached Amazon EBS volume is now available for attachment to other instances. An intern launches a new Linux Amazon EC2 instance in the us-west-1b Availability Zone (AZ) and is attempting to attach the Amazon EBS volume. The intern informs you that it is not possible and needs your help.

Which of the following explanations would you provide to them?

Correct answer

Amazon EBS volumes are Availability Zone (AZ) locked

The required IAM permissions are missing

Amazon EBS volumes are region locked

The Amazon EBS volume is encrypted

Overall explanation

Correct option:

Amazon EBS volumes are Availability Zone (AZ) locked

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. Amazon EBS volumes are flexible. For current-generation volumes attached to current-generation instance types, you can dynamically increase size, modify the provisioned IOPS capacity, and change volume type on live production volumes.

When you create an Amazon EBS volume, it is automatically replicated within its Availability Zone to prevent data loss due to the failure of any single hardware component. You can attach an Amazon EBS volume to an Amazon EC2 instance in the same Availability Zone (AZ).

Incorrect options:

Amazon EBS volumes are region locked - It's confined to an Availability Zone (AZ) and not by region.

The required IAM permissions are missing - This is a possibility as well but if permissions are not an issue then you are still confined to an availability zone (AZ).

The Amazon EBS volume is encrypted - This doesn't affect the ability to attach an Amazon EBS volume.

Reference:

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

Domain

Design High-Performing Architectures

Question 46Skipped

Which of the following is true regarding cross-zone load balancing as seen in Application Load Balancer versus Network Load Balancer?

By default, cross-zone load balancing is enabled for both Application Load Balancer and Network Load Balancer

Correct answer

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer

By default, cross-zone load balancing is disabled for both Application Load Balancer and Network Load Balancer

By default, cross-zone load balancing is disabled for Application Load Balancer and enabled for Network Load Balancer

Overall explanation

Correct option:

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer

By default, cross-zone load balancing is enabled for Application Load Balancer and disabled for Network Load Balancer. When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all the enabled Availability Zones. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone.

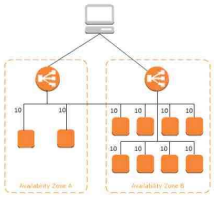
How cross-zone load balancing works:

Cross-Zone Load Balancing

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

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

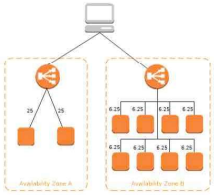
If cross-zone load balancing is enabled, each of the 10 targets receives 10% of the traffic. This is because each load balancer node can route its 50% of the client traffic to all 10 targets.



If cross-zone load balancing is disabled:

- Each of the two targets in Availability Zone A receives 25% of the traffic.
- Each of the eight targets in Availability Zone B receives 6.25% of the traffic.

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



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

With Network Load Balancers, cross-zone load balancing is disabled by default. After you create a Network Load Balancer, you can enable or disable cross-zone load balancing at any time. For more information, see Cross-Zone Load Balancing in the *User*

When you create a Classic Load Balancer, the default for cross-zone load balancing depends on how you create the load balancer. With the API or CLI, cross-zone load balancing is disabled by default. With the AWS Management Console, the option to enable cross-zone load balancing is selected by default. After you create a Classic Load Balancer, you can enable or disable cross-zone load balancing at any time. For more information, see Enable Cross-Zone Load Balancing in the *User Guide for Classic Load Balancers*.

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

Incorrect Options:

By default, cross-zone load balancing is disabled for both Application Load Balancer and Network Load Balancer

By default, cross-zone load balancing is enabled for both Application Load Balancer and Network Load Balancer

By default, cross-zone load balancing is disabled for Application Load Balancer and enabled for Network Load Balancer

Per the default cross-zone load balancing settings described earlier in the explanation, these three options are incorrect.

Reference:

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

Domain

Design Resilient Architectures

Question 47Skipped

A junior developer is learning to build websites using HTML, CSS, and JavaScript. He has created a static website and then deployed it on Amazon S3. Now he can't seem to figure out the endpoint for his super cool website.

As a solutions architect, can you help him figure out the allowed formats for the Amazon S3 website endpoints? (Select two)

Correct selection

http://bucket-name.s3-website.Region.amazonaws.com

Correct selection

http://bucket-name.s3-website-Region.amazonaws.com

http://s3-website-Region.bucket-name.amazonaws.com

http://s3-website.Region.bucket-name.amazonaws.com

http://bucket-name.Region.s3-website.amazonaws.com

Overall explanation

Correct options:

http://bucket-name.s3-website.Region.amazonaws.com

http://bucket-name.s3-website-Region.amazonaws.com

To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you enable static website hosting, set permissions, and add an index document.

Depending on your website requirements, you can also configure other options, including redirects, web traffic logging, and custom error documents.

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket.

Depending on your Region, your Amazon S3 website endpoints follow one of these two formats.

s3-website dash (-) Region - http://bucket-name.s3-website.Region.amazonaws.com

s3-website dot (.) Region - http://bucket-name.s3-website-Region.amazonaws.com

These URLs return the default index document that you configure for the website.

Incorrect options:

http://s3-website-Region.bucket-name.amazonaws.com

http://s3-website.Region.bucket-name.amazonaws.com

http://bucket-name.Region.s3-website.amazonaws.com

These three options do not meet the specifications for the Amazon S3 website endpoints format, so these are incorrect.

Reference:

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

Domain

Design Resilient Architectures

Question 48Skipped

A development team has deployed a microservice to the Amazon Elastic Container Service (Amazon ECS). The application layer is in a Docker container that provides both static and dynamic content through an Application Load Balancer. With increasing load, the Amazon ECS cluster is experiencing higher network usage. The development team has looked into the network usage and found that 90% of it is due to distributing static content of the application.

As a Solutions Architect, what do you recommend to improve the application's network usage and decrease costs?

Distribute the static content through Amazon EFS

Distribute the dynamic content through Amazon EFS

Distribute the dynamic content through Amazon S3

Correct answer

Distribute the static content through Amazon S3

Overall explanation

Correct option:

Distribute the static content through Amazon S3

You can use Amazon S3 to host a static website. On a static website, individual web pages include static content. They might also contain client-side scripts. To host a static website on Amazon S3, you configure an Amazon S3 bucket for website hosting and then upload your website content to the bucket. When you configure a bucket as a static website, you must enable website hosting, set permissions, and create and add an index document. Depending on your website requirements, you can also configure redirects, web traffic logging, and a custom error document.

Distributing the static content through Amazon S3 allows us to offload most of the network usage to Amazon S3 and free up our applications running on Amazon ECS.

Incorrect options:

Distribute the dynamic content through Amazon S3 - By contrast, a dynamic website relies on server-side processing, including server-side scripts such as PHP, JSP, or ASP.NET. Amazon S3 does not support server-side scripting, but AWS has other resources for hosting dynamic websites.

Distribute the static content through Amazon EFS

Distribute the dynamic content through Amazon EFS

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources. Using Amazon EFS for static or dynamic content will not change anything as static content on EFS would still have to be distributed by the Amazon ECS instances.

References:

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

Domain

Design Cost-Optimized Architectures

Question 49Skipped

A financial services firm uses a high-frequency trading system and wants to write the log files into Amazon S3. The system will also read these log files in parallel on a near real-time basis. The engineering team wants to address any data discrepancies that might arise when the trading system overwrites an existing log file and then tries to read that specific log file.

Which of the following options BEST describes the capabilities of Amazon S3 relevant to this scenario?

Correct answer

A process replaces an existing object and immediately tries to read it. Amazon S3 always returns the latest version of the object

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 does not return any data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the previous data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the new data

Overall explanation

Correct option:

A process replaces an existing object and immediately tries to read it. Amazon S3 always returns the latest version of the object

Amazon S3 delivers strong read-after-write consistency automatically, without changes to performance or availability, without sacrificing regional isolation for applications, and at no additional cost.

After a successful write of a new object or an overwrite of an existing object, any subsequent read request immediately receives the latest version of the object. Amazon S3 also provides strong consistency for list operations, so after a write, you can immediately perform a listing of the objects in a bucket with any changes reflected.

Strong read-after-write consistency helps when you need to immediately read an object after a write. For example, strong read-after-write consistency when you often read and list immediately after writing objects.

To summarize, all Amazon S3 GET, PUT, and LIST operations, as well as operations that change object tags, ACLs, or metadata, are strongly consistent. What you write is what you will read, and the results of a LIST will be an accurate reflection of what's in the bucket.

Incorrect options:

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the previous data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 does not return any data

A process replaces an existing object and immediately tries to read it. Until the change is fully propagated, Amazon S3 might return the new data

These three options contradict the earlier details provided in the explanation.

References:

https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#ConsistencyModel

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

Domain

Design Resilient Architectures

Question 50Skipped

An application with global users across AWS Regions had suffered an issue when the Elastic Load Balancing (ELB) in a Region malfunctioned thereby taking down the traffic with it. The manual intervention cost the company significant time and resulted in major revenue loss.

What should a solutions architect recommend to reduce internet latency and add automatic failover across AWS Regions?

Correct answer

Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations

Set up an Amazon Route 53 geoproximity routing policy to route traffic

Set up AWS Direct Connect as the backbone for each of the AWS Regions where the application is deployed

Create Amazon S3 buckets in different AWS Regions and configure Amazon CloudFront to pick the nearest edge location to the user

Overall explanation

Correct option:

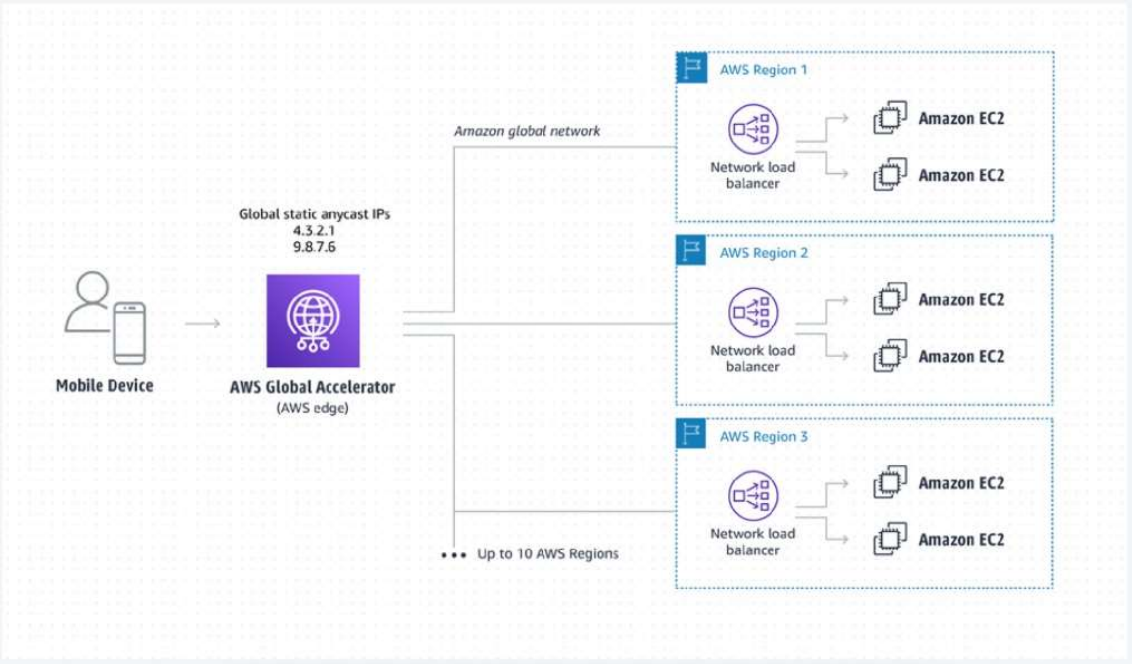
Set up AWS Global Accelerator and add endpoints to cater to users in different geographic locations

As your application architecture grows, so does the complexity, with longer user-facing IP lists and more nuanced traffic routing logic. AWS Global Accelerator solves this by providing you with two static IPs that are anycast from our globally distributed edge locations, giving you a single entry point to your application, regardless of how many AWS Regions it's deployed in. This allows you to add or remove origins, Availability Zones or Regions without reducing your application availability. Your traffic routing is managed manually, or in console with endpoint traffic dials and weights. If your application endpoint has a failure or availability issue, AWS Global Accelerator will automatically redirect your new connections to a healthy endpoint within seconds.

By using AWS Global Accelerator, you can:

1. Associate the static IP addresses provided by AWS Global Accelerator to regional AWS resources or endpoints, such as Network Load Balancers, Application Load Balancers, EC2 Instances, and Elastic IP addresses. The IP addresses are anycast from AWS edge locations so they provide onboarding to the AWS global network close to your users.
2. Easily move endpoints between Availability Zones or AWS Regions without needing to update your DNS configuration or change client-facing applications.
3. Dial traffic up or down for a specific AWS Region by configuring a traffic dial percentage for your endpoint groups. This is especially useful for testing performance and releasing updates.
4. Control the proportion of traffic directed to each endpoint within an endpoint group by assigning weights across the endpoints.

AWS Global Accelerator for Multi-Region applications:



via - https://aws.amazon.com/global-accelerator/

Incorrect options:

Set up AWS Direct Connect as the backbone for each of the AWS Regions where the application is deployed - AWS Direct Connect can reduce latency to great extent. Direct Connect is used to connect on-premises systems to AWS Cloud for extremely low latency use cases. It cannot be used to serve users directly.

Create Amazon S3 buckets in different AWS Regions and configure Amazon CloudFront to pick the nearest edge location to the user - If most of the content is static, we can configure Amazon CloudFront to improve performance. In the current scenario, the architecture has ELBs, Amazon EC2 instances too that need to be covered in the automatic failover plan.

Set up an Amazon Route 53 geoproximity routing policy to route traffic - Geoproximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. Unlike AWS Global Accelerator, managing and routing to different instances, ELBs and other AWS resources will become an operational overhead as the resource count reaches into the hundreds. With inbuilt features like Static anycast IP addresses, fault tolerance using network zones, Global performance-based routing, TCP Termination at the Edge - AWS Global Accelerator is the right choice for multi-region, low latency use cases.

References:

https://aws.amazon.com/global-accelerator/features/

https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-geoproximity

Domain

Design High-Performing Architectures

Question 51Skipped

A Customer relationship management (CRM) application is facing user experience issues with users reporting frequent sign-in requests from the application. The application is currently hosted on multiple Amazon EC2 instances behind an Application Load Balancer. The engineering team has identified the root cause as unhealthy servers causing session data to be lost. The team would like to implement a distributed in-memory cache-based session management solution.

As a solutions architect, which of the following solutions would you recommend?

Use Application Load Balancer sticky sessions

Use Amazon RDS for distributed in-memory cache based session management

Use Amazon DynamoDB for distributed in-memory cache based session management

Correct answer

Use Amazon Elasticache for distributed in-memory cache based session management

Overall explanation

Correct option:

Use Amazon Elasticache for distributed in-memory cache based session management

Amazon ElastiCache can be used as a distributed in-memory cache for session management. Amazon ElastiCache allows you to seamlessly set up, run, and scale popular open-Source compatible in-memory data stores in the cloud. Session stores can be set up using both Memcached or Redis for ElastiCache.

Amazon ElastiCache for Redis is a great choice for real-time transactional and analytical processing use cases such as caching, chat/messaging, gaming leaderboards, geospatial, machine learning, media streaming, queues, real-time analytics, and session store.

Amazon ElastiCache for Memcached is a Memcached-compatible in-memory key-value store service that can be used as a cache or a data store. Session stores are easy to create with Amazon ElastiCache for Memcached.

How Amazon ElastiCache Works:

via - https://aws.amazon.com/elasticache/

Incorrect options:

Use Amazon RDS for distributed in-memory cache based session management - Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It cannot be used as a distributed in-memory cache for session management, hence this option is incorrect.

Use Amazon DynamoDB for distributed in-memory cache based session management - Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. Amazon DynamoDB is a NoSQL database and is not the right fit for a distributed in-memory cache-based session management solution.

Use Application Load Balancer sticky sessions - Although sticky sessions enable each user to interact with one server and one server only, however, in case of an unhealthy server, all the session data is gone as well. Therefore Amazon Elasticache powered distributed in-memory cache-based session management is a better solution.

References:

https://aws.amazon.com/getting-started/hands-on/building-fast-session-caching-with-amazon-elasticache-for-redis/

https://aws.amazon.com/elasticache/

Domain

Design High-Performing Architectures

Question 52Skipped

An online gaming company wants to block access to its application from specific countries; however, the company wants to allow its remote development team (from one of the blocked countries) to have access to the application. The application is deployed on Amazon EC2 instances running under an Application Load Balancer with AWS Web Application Firewall (AWS WAF).

As a solutions architect, which of the following solutions can be combined to address the given use-case? (Select two)

Use Application Load Balancer geo match statement listing the countries that you want to block

Use Application Load Balancer IP set statement that specifies the IP addresses that you want to allow through

Create a deny rule for the blocked countries in the network access control list (network ACL) associated with each of the Amazon EC2 instances

Correct selection

Use AWS WAF geo match statement listing the countries that you want to block

Correct selection

Use AWS WAF IP set statement that specifies the IP addresses that you want to allow through

Overall explanation

Correct options:

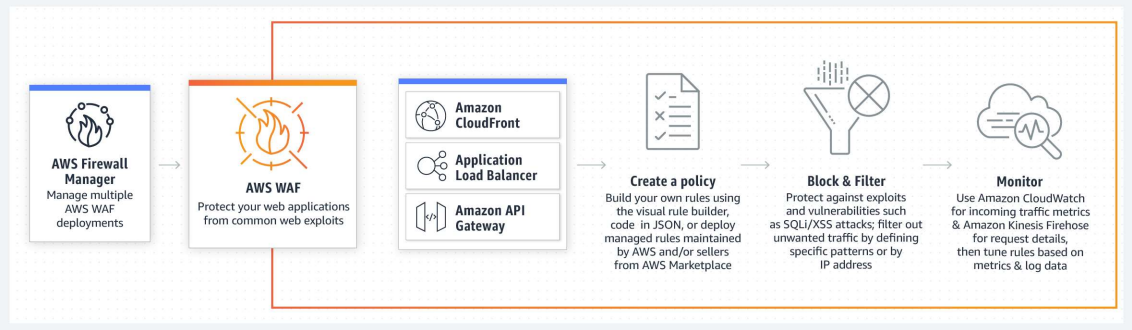
Use AWS WAF geo match statement listing the countries that you want to block

Use AWS WAF IP set statement that specifies the IP addresses that you want to allow through

AWS WAF is a web application firewall that helps protect your web applications or APIs against common web exploits that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that block common attack patterns and rules that filter out specific traffic patterns you define.

You can deploy AWS WAF on Amazon CloudFront as part of your CDN solution, the Application Load Balancer that fronts your web servers or origin servers running on Amazon EC2, or Amazon API Gateway for your APIs.

AWS WAF - How it Works?:



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

To block specific countries, you can create a AWS WAF geo match statement listing the countries that you want to block, and to allow traffic from IPs of the remote development team, you can create a WAF IP set statement that specifies the IP addresses that you want to allow through. You can combine the two rules as shown below:

Incorrect options:

Create a deny rule for the blocked countries in the network access control list (network ACL) associated with each of the Amazon EC2 instances - A network access control list (network ACL) 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. A network access control list (network ACL) does not have the capability to block traffic based on geographic match conditions.

Use Application Load Balancer geo match statement listing the countries that you want to block

Use Application Load Balancer IP set statement that specifies the IP addresses that you want to allow through

An Application Load Balancer operates at the request level (layer 7), routing traffic to targets – Amazon EC2 instances, containers, IP addresses, and AWS Lambda functions based on the

content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications.

An Application Load Balancer cannot block or allow traffic based on geographic match conditions or IP based conditions. Both these options have been added as distractors.

References:

https://docs.aws.amazon.com/waf/latest/developerguide/waf-rule-statement-type-geo-match.html

https://aws.amazon.com/blogs/security/how-to-use-aws-waf-to-filter-incoming-traffic-from-embargoed-countries/

Domain

Design Secure Architectures

Question 53Skipped

A company wants to publish an event into an Amazon Simple Queue Service (Amazon SQS) queue whenever a new object is uploaded on Amazon S3.

Which of the following statements are true regarding this functionality?

Correct answer

Only Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed

Only FIFO Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas Standard SQS queue is not allowed

Both Standard Amazon SQS queue and FIFO SQS queue are allowed as an Amazon S3 event notification destination

Neither Standard Amazon SQS queue nor FIFO SQS queue are allowed as an Amazon S3 event notification destination

Overall explanation

Correct option:

Only Standard Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas FIFO SQS queue is not allowed

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

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

Amazon Simple Notification Service (Amazon SNS) topic

Amazon Simple Queue Service (Amazon SQS) queue

AWS Lambda

Currently, the Standard Amazon SQS queue is only allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed.

Incorrect options:

Both Standard Amazon SQS queue and FIFO SQS queue are allowed as an Amazon S3 event notification destination

Neither Standard Amazon SQS queue nor FIFO SQS queue are allowed as an Amazon S3 event notification destination

Only FIFO Amazon SQS queue is allowed as an Amazon S3 event notification destination, whereas Standard SQS queue is not allowed

These three options contradict the details provided in the explanation above. To summarize, the Standard Amazon SQS queue is only allowed as an Amazon S3 event notification destination, whereas the FIFO SQS queue is not allowed. Hence these three options are incorrect.

Reference:

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

Domain

Design Resilient Architectures

Question 54Skipped

You are a cloud architect at an IT company. The company has multiple enterprise customers that manage their own mobile applications that capture and send data to Amazon Kinesis Data Streams. They have been getting a ProvisionedThroughputExceededException exception. You have been contacted to help and upon analysis, you notice that messages are being sent one by one at a high rate.

Which of the following options will help with the exception while keeping costs at a minimum?

Correct answer

Use batch messages

Increase the number of shards

Decrease the Stream retention duration

Use Exponential Backoff

Overall explanation

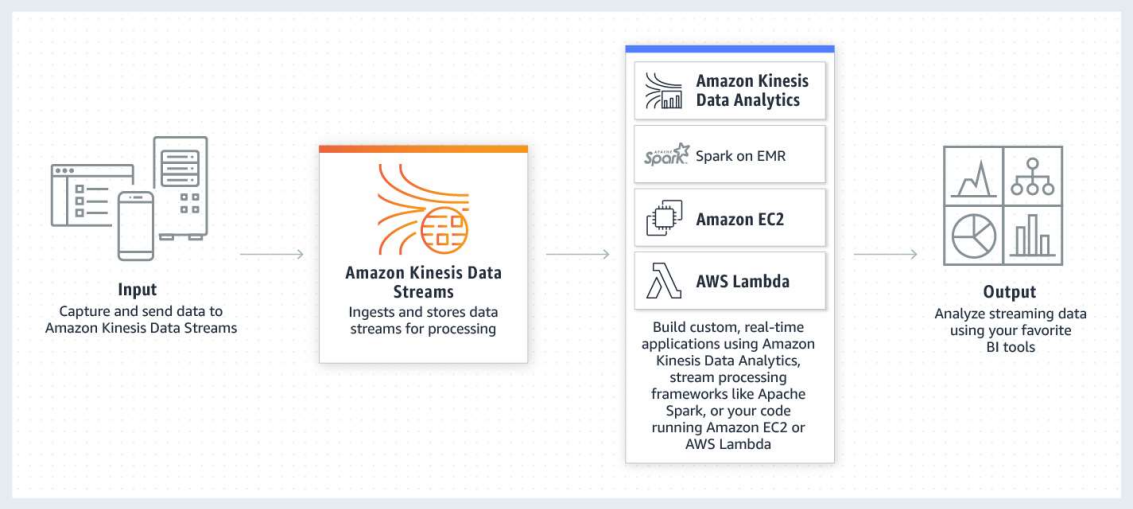
Correct option:

Use batch messages

Amazon Kinesis Data Streams (KDS) is a massively scalable and durable real-time data streaming service. KDS can continuously capture gigabytes of data per second from hundreds of thousands of sources such as website clickstreams, database event streams, financial transactions, social media feeds, IT logs, and location-tracking events. The data collected is

available in milliseconds to enable real-time analytics use cases such as real-time dashboards, real-time anomaly detection, dynamic pricing, and more.

Amazon Kinesis Data Streams Overview:



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

When a host needs to send many records per second (RPS) to Amazon Kinesis, simply calling the basic PutRecord API action in a loop is inadequate. To reduce overhead and increase throughput, the application must batch records and implement parallel HTTP requests. This will increase the efficiency overall and ensure you are optimally using the shards.

Incorrect options:

Use Exponential Backoff - While this may help in the short term, as soon as the request rate increases, you will see the ProvisionedThroughputExceededException exception again.

Increase the number of shards - Increasing shards could be a short term fix but will substantially increase the cost, so this option is ruled out.

Decrease the Stream retention duration - This operation may result in data loss and won't help with the exceptions, so this option is incorrect.

References:

https://aws.amazon.com/blogs/big-data/implementing-efficient-and-reliable-producers-with-the-amazon-kinesis-producer-library/

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

Domain

Design High-Performing Architectures

Question 55Skipped

A company wants to store business-critical data on Amazon Elastic Block Store (Amazon EBS) volumes which provide persistent storage independent of Amazon EC2 instances. During a test

run, the development team found that on terminating an Amazon EC2 instance, the attached Amazon EBS volume was also lost, which was contrary to their assumptions.

As a solutions architect, could you explain this issue?

The Amazon EBS volumes were not backed up on Amazon EFS file system storage, resulting in the loss of volume

On termination of an Amazon EC2 instance, all the attached Amazon EBS volumes are always terminated

Correct answer

The Amazon EBS volume was configured as the root volume of Amazon EC2 instance. On termination of the instance, the default behavior is to also terminate the attached root volume

The Amazon EBS volumes were not backed up on Amazon S3 storage, resulting in the loss of volume

Overall explanation

Correct option:

The Amazon EBS volume was configured as the root volume of Amazon EC2 instance. On termination of the instance, the default behavior is to also terminate the attached root volume

Amazon Elastic Block Store (EBS) is an easy to use, high-performance block storage service designed for use with Amazon Elastic Compute Cloud (EC2) for both throughput and transaction-intensive workloads at any scale.

When you launch an instance, the root device volume contains the image used to boot the instance. You can choose between AMIs backed by Amazon EC2 instance store and AMIs backed by Amazon EBS.

By default, the root volume for an AMI backed by Amazon EBS is deleted when the instance terminates. You can change the default behavior to ensure that the volume persists after the instance terminates. Non-root EBS volumes remain available even after you terminate an instance to which the volumes were attached. Therefore, this option is correct.

Incorrect options:

The Amazon EBS volumes were not backed up on Amazon S3 storage, resulting in the loss of volume

The Amazon EBS volumes were not backed up on Amazon EFS file system storage, resulting in the loss of volume

Amazon EBS volumes do not need to back up the data on Amazon S3 or Amazon EFS filesystem. Both these options are added as distractors.

On termination of an Amazon EC2 instance, all the attached Amazon EBS volumes are always terminated - As mentioned earlier, non-root Amazon EBS volumes remain available

even after you terminate an instance to which the volumes were attached. Hence this option is incorrect.

Reference:

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

Domain

Design High-Performing Architectures

Question 56Skipped

A medium-sized business has a taxi dispatch application deployed on an Amazon EC2 instance. Because of an unknown bug, the application causes the instance to freeze regularly. Then, the instance has to be manually restarted via the AWS management console.

Which of the following is the MOST cost-optimal and resource-efficient way to implement an automated solution until a permanent fix is delivered by the development team?

Use Amazon EventBridge events to trigger an AWS Lambda function to reboot the instance status every 5 minutes

Setup an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, Amazon CloudWatch Alarm can publish to an Amazon Simple Notification Service (Amazon SNS) event which can then trigger an AWS lambda function. The AWS lambda function can use Amazon EC2 API to reboot the instance

Correct answer

Setup an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, an EC2 Reboot CloudWatch Alarm Action can be used to reboot the instance

Use Amazon EventBridge events to trigger an AWS Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the AWS lambda function can use Amazon EC2 API to reboot the instance

Overall explanation

Correct option:

Setup an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, an EC2 Reboot CloudWatch Alarm Action can be used to reboot the instance

Using Amazon CloudWatch alarm actions, you can create alarms that automatically stop, terminate, reboot, or recover your Amazon EC2 instances. You can use the stop or terminate actions to help you save money when you no longer need an instance to be running. You can use the reboot and recover actions to automatically reboot those instances or recover them onto new hardware if a system impairment occurs.

You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically reboots the instance. The reboot alarm action is recommended for Instance

Health Check failures (as opposed to the recover alarm action, which is suited for System Health Check failures).

Incorrect options:

Setup an Amazon CloudWatch alarm to monitor the health status of the instance. In case of an Instance Health Check failure, Amazon CloudWatch Alarm can publish to an Amazon Simple Notification Service (Amazon SNS) event which can then trigger an AWS lambda function. The AWS lambda function can use Amazon EC2 API to reboot the instance

Use Amazon EventBridge events to trigger an AWS Lambda function to check the instance status every 5 minutes. In the case of Instance Health Check failure, the AWS lambda function can use Amazon EC2 API to reboot the instance

Use Amazon EventBridge events to trigger an AWS Lambda function to reboot the instance status every 5 minutes

Using Amazon EventBridge event or Amazon CloudWatch alarm to trigger an AWS lambda function, directly or indirectly, is wasteful of resources. You should just use the EC2 Reboot CloudWatch Alarm Action to reboot the instance. So all the options that trigger the AWS lambda function are incorrect.

Reference:

https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/UsingAlarmActions.html

Domain

Design Cost-Optimized Architectures

Question 57Skipped

The engineering team at an online fashion retailer uses AWS Cloud to manage its technology infrastructure. The Amazon EC2 server fleet is behind an Application Load Balancer and the fleet strength is managed by an Auto Scaling group. Based on the historical data, the team is anticipating a huge traffic spike during the upcoming Thanksgiving sale.

As an AWS solutions architect, what feature of the Auto Scaling group would you leverage so that the potential surge in traffic can be preemptively addressed?

Auto Scaling group lifecycle hook

Correct answer

Auto Scaling group scheduled action

Auto Scaling group step scaling policy

Auto Scaling group target tracking scaling policy

Overall explanation

Correct option:

Auto Scaling group scheduled action

The engineering team can create a scheduled action for the Auto Scaling group to pre-emptively provision additional instances for the sale duration. This makes sure that adequate instances are ready before the sale goes live. 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 that are specified by the scaling action.

Incorrect options:

Auto Scaling group target tracking scaling policy - With target tracking scaling policies, you choose a scaling metric and set a target value. Application Auto Scaling creates and manages the Amazon CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.

Auto Scaling group step scaling policy - With step scaling, you choose scaling metrics and threshold values for the Amazon 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.

Both the target tracking as well as step scaling policies entail a lag wherein the instances will be provisioned only when the underlying Amazon CloudWatch alarms go off. Therefore these two options are not pre-emptive in nature and ruled out for the given use-case.

Auto Scaling group lifecycle hook - Auto Scaling group lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances. For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates. Lifecycle hooks cannot be used to pre-emptively provision additional instances for a specific period such as the sale duration.

Reference:

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

Domain

Design Resilient Architectures

Question 58Skipped

A financial services company has to retain the activity logs for each of their customers to meet compliance guidelines. Depending on the business line, the company wants to retain the logs for 5-10 years in highly available and durable storage on AWS. The overall data size is expected to be in Petabytes. In case of an audit, the data would need to be accessible within a timeframe of up to 48 hours.

Which AWS storage option is the MOST cost-effective for the given compliance requirements?

Third party tape storage

Amazon S3 Glacier

Amazon S3 Standard storage

Correct answer

Amazon S3 Glacier Deep Archive

Overall explanation

Correct option:

Amazon S3 Glacier Deep Archive

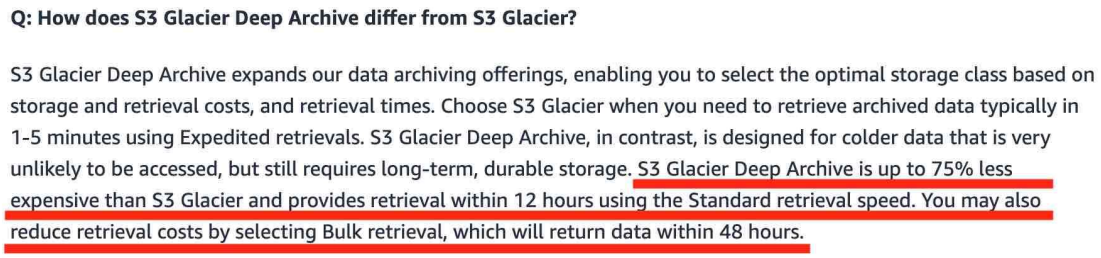
Amazon S3 Glacier and Amazon S3 Glacier Deep Archive are secure, durable, and extremely low-cost Amazon S3 cloud storage classes for data archiving and long-term backup. They are designed to deliver 99.999999999% durability, and provide comprehensive security and compliance capabilities that can help meet even the most stringent regulatory requirements.

Amazon S3 Glacier Deep Archive is a new Amazon S3 storage class that provides secure and durable object storage for long-term retention of data that is accessed once or twice in a year. From just $0.00099 per GB-month (less than one-tenth of one cent, or about $1 per TB-month), Amazon S3 Glacier Deep Archive offers the lowest cost storage in the cloud, at prices significantly lower than storing and maintaining data in on-premises magnetic tape libraries or archiving data off-site.

Amazon S3 Glacier Deep Archive is up to 75% less expensive than Amazon S3 Glacier and provides retrieval within 12 hours using the Standard retrieval speed. You may also reduce retrieval costs by selecting Bulk retrieval, which will return data within 48 hours.

Therefore, Amazon S3 Glacier Deep Archive is the correct choice.

Amazon S3 Glacier vs Amazon S3 Glacier Deep Archive:



Q: How does S3 Glacier Deep Archive differ from S3 Glacier?

S3 Glacier Deep Archive expands our data archiving offerings, enabling you to select the optimal storage class based on storage and retrieval costs, and retrieval times. Choose S3 Glacier when you need to retrieve archived data typically in 1-5 minutes using Expedited retrievals. S3 Glacier Deep Archive, in contrast, is designed for colder data that is very unlikely to be accessed, but still requires long-term, durable storage. S3 Glacier Deep Archive is up to 75% less expensive than S3 Glacier and provides retrieval within 12 hours using the Standard retrieval speed. You may also reduce retrieval costs by selecting Bulk retrieval, which will return data within 48 hours.

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

Incorrect options:

Amazon S3 Glacier - As mentioned earlier, Amazon S3 Glacier Deep Archive is up to 75% less expensive than Amazon S3 Glacier and provides retrieval within 12 hours. So using Amazon S3 Glacier is not the correct choice.

Third party tape storage

Amazon S3 Standard storage

Given the relaxed retrieval times, Amazon S3 standard storage would be much costlier than the Amazon S3 Glacier Deep Archive, so Amazon S3 standard storage is not the correct option. Using Third-party tape storage is ruled out as the company wants to use an AWS storage service. Therefore, both of these options are incorrect.

Reference:

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

Domain

Design Cost-Optimized Architectures

Question 59Skipped

The engineering team at an e-commerce company uses an AWS Lambda function to write the order data into a single DB instance Amazon Aurora cluster. The team has noticed that many order- writes to its Aurora cluster are getting missed during peak load times. The diagnostics data has revealed that the database is experiencing high CPU and memory consumption during traffic spikes. The team also wants to enhance the availability of the Aurora DB.

Which of the following steps would you combine to address the given scenario? (Select two)

Create a standby Aurora instance in another Availability Zone to improve the availability as the standby can serve as a failover target

Correct selection

Handle all read operations for your application by connecting to the reader endpoint of the Amazon Aurora cluster so that Aurora can spread the load for read-only connections across the Aurora replica

Increase the concurrency of the AWS Lambda function so that the order-writes do not get missed during traffic spikes

Use Amazon EC2 instances behind an Application Load Balancer to write the order data into Amazon Aurora cluster

Correct selection

Create a replica Aurora instance in another Availability Zone to improve the availability as the replica can serve as a failover target

Overall explanation

Correct options:

Handle all read operations for your application by connecting to the reader endpoint of the Amazon Aurora cluster so that Aurora can spread the load for read-only connections across the Aurora replica

When you create a second, third, and so on DB instance in an Aurora-provisioned DB cluster, Aurora automatically sets up replication from the writer DB instance to all the other DB instances. These other DB instances are read-only and are known as Aurora Replicas.

Aurora Replicas have two main purposes. You can issue queries to them to scale the read operations for your application. You typically do so by connecting to the reader endpoint of the cluster. That way, Aurora can spread the load for read-only connections across as many Aurora Replicas as you have in the cluster. Aurora Replicas also help to increase availability. If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the new writer.

Aurora Replicas

When you create a second, third, and so on DB instance in an Aurora provisioned DB cluster, Aurora automatically sets up replication from the writer DB instance to all the other DB instances. These other DB instances are read-only and are known as Aurora Replicas. We also refer to them as reader instances when discussing the ways that you can combine writer and reader DB instances within a cluster.

Aurora Replicas have two main purposes. You can issue queries to them to scale the read operations for your application. You typically do so by connecting to the reader endpoint of the cluster. That way, Aurora can spread the load for read-only connections across as many Aurora Replicas as you have in the cluster. Aurora Replicas also help to increase availability. If the writer instance in a cluster becomes unavailable, Aurora automatically promotes one of the reader instances to take its place as the new writer.

An Aurora DB cluster can contain up to 15 Aurora Replicas. The Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region.

The data in your DB cluster has its own high availability and reliability features, independent of the DB instances in the cluster. If you aren't familiar with Aurora storage features, see Overview of Aurora storage. The DB cluster volume is physically made up of multiple copies of the data for the DB cluster. The primary instance and the Aurora Replicas in the DB cluster all see the data in the cluster volume as a single logical volume.

As a result, all Aurora Replicas return the same data for query results with minimal replica lag. This lag is usually much less than 100 milliseconds after the primary instance has written an update. Replica lag varies depending on the rate of database change. That is, during periods where a large amount of write operations occur for the database, you might see an increase in replica lag.

Aurora Replicas work well for read scaling because they are fully dedicated to read operations on your cluster volume. Write operations are managed by the primary instance. Because the cluster volume is shared among all DB instances in your DB cluster, minimal additional work is required to replicate a copy of the data for each Aurora Replica.

To increase availability, you can use Aurora Replicas as failover targets. That is, if the primary instance fails, an Aurora Replica is promoted to the primary instance. There is a brief interruption during which read and write requests made to the primary instance fail with an exception. When this happens, some of the Aurora Replicas might be rebooted, depending on the DB engine version. For information about the rebooting behavior of different Aurora DB engine versions, see Rebooting an Amazon Aurora DB cluster or Amazon Aurora DB instance. Promoting an Aurora Replica this way is much faster than recreating the primary instance. If your Aurora DB cluster doesn't include any Aurora Replicas, then your DB cluster isn't available while your DB instance is recovering from the failure.

For high-availability scenarios, we recommend that you create one or more Aurora Replicas. These should be of the same DB instance class as the primary instance and in different Availability Zones for your Aurora DB cluster. For more information about Aurora Replicas as failover targets, see Fault tolerance for an Aurora DB cluster.

You can't create an encrypted Aurora Replica for an unencrypted Aurora DB cluster. You can't create an unencrypted Aurora Replica for an encrypted Aurora DB cluster.

via
- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Replication.html

Create a replica Aurora instance in another Availability Zone to improve the availability as the replica can serve as a failover target

If the primary instance in a DB cluster using single-master replication fails, Aurora automatically fails over to a new primary instance in one of two ways:

By promoting an existing Aurora Replica to the new primary instance By creating a new primary instance

Fault tolerance for an Aurora DB cluster

An Aurora DB cluster is fault tolerant by design. The cluster volume spans multiple Availability Zones (AZs) in a single AWS Region, and each Availability Zone contains a copy of the cluster volume data. This functionality means that your DB cluster can tolerate a failure of an Availability Zone without any loss of data and only a brief interruption of service.

If the primary instance in a DB cluster using single-master replication fails, Aurora automatically fails over to a new primary instance in one of two ways:

- By promoting an existing Aurora Replica to the new primary instance
- By creating a new primary instance

If the DB cluster has one or more Aurora Replicas, then an Aurora Replica is promoted to the primary instance during a failure event. A failure event results in a brief interruption, during which read and write operations fail with an exception. However, service is typically restored in less than 120 seconds, and often less than 60 seconds. To increase the availability of your DB cluster, we recommend that you create at least one or more Aurora Replicas in two or more different Availability Zones.

> **Tip**
>
> In Aurora MySQL 2.10 and higher, you can improve availability during a failover by having more than one reader DB instance in a cluster. In Aurora MySQL 2.10 and higher, Aurora restarts only the writer DB instance and the failover target during a failover. Other reader DB instances in the cluster remain available to continue processing queries through connections to the reader endpoint.

You can customize the order in which your Aurora Replicas are promoted to the primary instance after a failure by assigning each replica a priority. Priorities range from 0 for the first priority to 15 for the last priority. If the primary instance fails, Amazon RDS promotes the Aurora Replica with the better priority to the new primary instance. You can modify the priority of an Aurora Replica at any time. Modifying the priority doesn't trigger a failover.

More than one Aurora Replica can share the same priority, resulting in promotion tiers. If two or more Aurora Replicas share the same priority, then Amazon RDS promotes the replica that is largest in size. If two or more Aurora Replicas share the same priority and size, then Amazon RDS promotes an arbitrary replica in the same promotion tier.

If the DB cluster doesn't contain any Aurora Replicas, then the primary instance is recreated in the same AZ during a failure event. A failure event results in an interruption during which read and write operations fail with an exception. Service is restored when the new primary instance is created, which typically takes less than 10 minutes. Promoting an Aurora Replica to the primary instance is much faster than creating a new primary instance.

Suppose that the primary instance in your cluster is unavailable because of an outage that affects an entire AZ. In this case, the way to bring a new primary instance online depends on whether your cluster uses a Multi-AZ configuration:

- If your provisioned or Aurora Serverless v2 cluster contains any reader instances in other AZs, Aurora uses the failover mechanism to promote one of those reader instances to be the new primary instance.
- If your provisioned or Aurora Serverless v2 cluster only contains a single DB instance, or if the primary instance and all reader instances are in the same AZ, make sure to manually create one or more new DB instances in another AZ.
- If your cluster uses Aurora Serverless v1, Aurora automatically creates a new DB instance in another AZ. However, this process involves a host replacement and thus takes longer than a failover.

via
- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Concepts.AuroraHighAvailability.html

Incorrect options:

Create a standby Aurora instance in another Availability Zone to improve the availability as the standby can serve as a failover target - There are no standby instances in Aurora. Aurora performs an automatic failover to a read replica when a problem is detected. So this option is incorrect.

Read replicas, Multi-AZ deployments, and multi-region deployments:

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 read scaling
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 primary	Can be manually promoted to a standalone database instance (non-Aurora) or to be the primary instance (Aurora)

via - https://aws.amazon.com/rds/features/read-replicas/

Increase the concurrency of the AWS Lambda function so that the order-writes do not get missed during traffic spikes - Increasing the concurrency of the AWS Lambda function would not resolve the issue since the bottleneck is at the database layer, as exhibited by the high CPU and memory consumption for the Aurora instance. This option has been added as a distractor.

Use Amazon EC2 instances behind an Application Load Balancer to write the order data into Amazon Aurora cluster - Using Amazon EC2 instances behind an Application Load Balancer would not resolve the issue since the bottleneck is at the database layer, as exhibited by the high CPU and memory consumption for the Aurora instance. This option has been added as a distractor.

References:

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

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

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

Domain

Design Resilient Architectures

Question 60Skipped

The data engineering team at an e-commerce company has set up a workflow to ingest the clickstream data into the raw zone of the Amazon S3 data lake. The team wants to run some SQL based data sanity checks on the raw zone of the data lake.

What AWS services would you recommend for this use-case such that the solution is cost-effective and easy to maintain?

Correct answer

Use Amazon Athena to run SQL based analytics against Amazon S3 data

Load the incremental raw zone data into Amazon RDS on an hourly basis and run the SQL based sanity checks

Load the incremental raw zone data into Amazon Redshift on an hourly basis and run the SQL based sanity checks

Load the incremental raw zone data into an Amazon EMR based Spark Cluster on an hourly basis and use SparkSQL to run the SQL based sanity checks

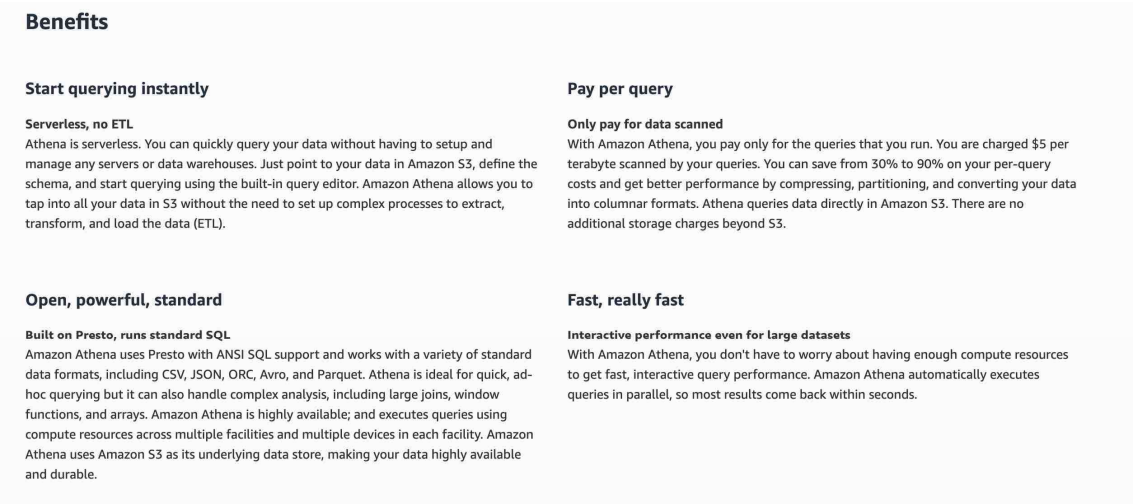
Overall explanation

Correct option:

Use Amazon Athena to run SQL based analytics against Amazon S3 data

Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon S3 using standard SQL. Amazon Athena is serverless, so there is no infrastructure to set up or manage, and customers pay only for the queries they run. You can use Athena to process logs, perform ad-hoc analysis, and run interactive queries.

Amazon Athena Benefits:



via - https://aws.amazon.com/athena/

Incorrect options:

Load the incremental raw zone data into Amazon Redshift on an hourly basis and run the SQL based sanity checks - Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse product designed for large scale data set storage and analysis. As the development team would have to maintain and monitor the Amazon Redshift cluster size and would require significant development time to set up the processes to consume the data periodically, so this option is ruled out.

Load the incremental raw zone data into an Amazon EMR based Spark Cluster on an hourly basis and use SparkSQL to run the SQL based sanity checks - Amazon EMR is the industry-

leading cloud big data platform for processing vast amounts of data using open source tools such as Apache Spark, Apache Hive, Apache HBase, Apache Flink, Apache Hudi, and Presto. Amazon EMR uses Hadoop, an open-source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances. Using an Amazon EMR cluster would imply managing the underlying infrastructure so it's ruled out because the correct solution for the given use-case should require the least amount of development effort and ongoing maintenance.

Load the incremental raw zone data into Amazon RDS on an hourly basis and run the SQL based sanity checks - Loading the incremental data into Amazon RDS implies data migration jobs will have to be written via a AWS Lambda function or an Amazon EC2 based process. This goes against the requirement that the solution should involve the least amount of development effort and ongoing maintenance. Hence this option is not correct.

Reference:

https://aws.amazon.com/athena/

Domain

Design Cost-Optimized Architectures

Question 61Skipped

A mobile chat application uses Amazon DynamoDB as its database service to provide low latency chat updates. A new developer has joined the team and is reviewing the configuration settings for Amazon DynamoDB which have been tweaked for certain technical requirements. AWS CloudTrail service has been enabled on all the resources used for the project. Yet, Amazon DynamoDB encryption details are nowhere to be found.

Which of the following options can explain the root cause for the given issue?

Correct answer

By default, all Amazon DynamoDB tables are encrypted using AWS owned keys, which do not write to AWS CloudTrail logs

By default, all Amazon DynamoDB tables are encrypted under Customer managed keys, which do not write to AWS CloudTrail logs

By default, all Amazon DynamoDB tables are encrypted under AWS managed Keys, which do not write to AWS CloudTrail logs

By default, all Amazon DynamoDB tables are encrypted using Data keys, which do not write to AWS CloudTrail logs

Overall explanation

Correct option:

By default, all Amazon DynamoDB tables are encrypted using AWS owned keys, which do not write to AWS CloudTrail logs

AWS owned keys are not stored in your AWS account. They are part of a collection of KMS keys that AWS owns and manages for use in multiple AWS accounts. AWS services can use AWS

owned keys to protect your data. AWS owned keys used by DynamoDB are rotated every year (approximately 365 days).

You cannot view, manage, or use AWS owned keys, or audit their use. However, you do not need to do any work or change any programs to protect the keys that encrypt your data. You are not charged a monthly fee or a usage fee for use of AWS owned keys, and they do not count against AWS KMS quotas for your account.

All DynamoDB tables are encrypted. There is no option to enable or disable encryption for new or existing tables. By default, all tables are encrypted under an AWS owned key in the DynamoDB service account. However, you can select an option to encrypt some or all of your tables under a customer managed key or the AWS managed key for DynamoDB in your account.

Incorrect options:

By default, all Amazon DynamoDB tables are encrypted under AWS managed Keys, which do not write to AWS CloudTrail logs

By default, all Amazon DynamoDB tables are encrypted under Customer managed keys, which do not write to AWS CloudTrail logs

By default, all Amazon DynamoDB tables are encrypted using Data keys, which do not write to AWS CloudTrail logs

These three options contradict the explanation provided above, so these options are incorrect.

References:

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

https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#master_keys

Domain

Design Secure Architectures

Question 62Skipped

A gaming company is doing pre-launch testing for its new product. The company runs its production database on an Aurora MySQL DB cluster and the performance testing team wants access to multiple test databases that must be re-created from production data. The company has hired you as an AWS Certified Solutions Architect - Associate to deploy a solution to create these test databases quickly with the LEAST required effort.

What would you suggest to address this use case?

Take a backup of the Aurora MySQL database instance using the mysqldump utility, create multiple new test database instances and restore each test database from the backup

Set up binlog replication in the Aurora MySQL database instance to create multiple new test database instances

Correct answer

Use database cloning to create multiple clones of the production database and use each clone as a test database

Enable database Backtracking on the production database and let the testing team use the production database

Overall explanation

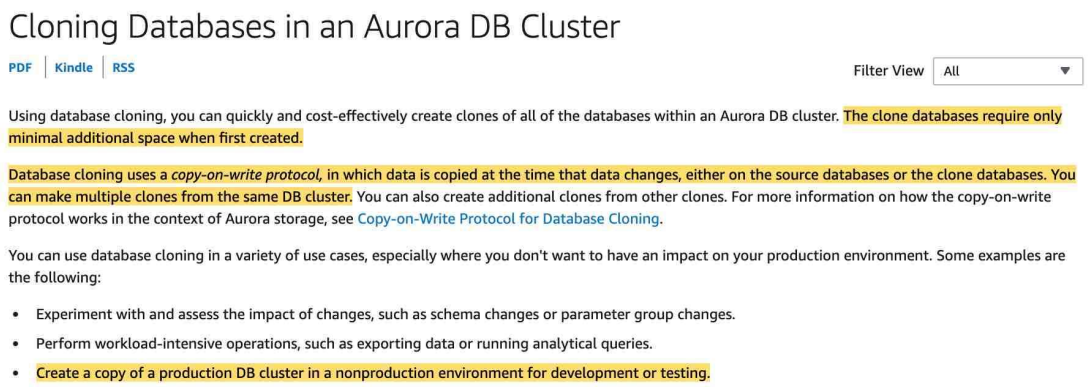
Correct option:

Use database cloning to create multiple clones of the production database and use each clone as a test database

You can quickly create clones of an Aurora DB by using the database cloning feature. In addition, database cloning uses a copy-on-write protocol, in which data is copied only at the time the data changes, either on the source database or the clone database. Cloning is much faster than a manual snapshot of the DB cluster.

For the given use case, the most optimal solution is to clone the DB cluster. This would allow the performance testing team to have quick access to the production data in an isolated way. The team can iterate over the various test phases by deleting existing test databases and then cloning the production DB to create new test databases.

You cannot clone databases across AWS regions. The clone databases must be created in the same region as the source databases. Currently, you are limited to 15 clones based on a copy, including clones based on other clones. After that, only copies can be created. However, each copy can also have up to 15 clones.



Cloning Databases in an Aurora DB Cluster

PDF | Kindle | RSS

Filter View: All

Using database cloning, you can quickly and cost-effectively create clones of all of the databases within an Aurora DB cluster. The clone databases require only minimal additional space when first created.

Database cloning uses a *copy-on-write protocol,* in which data is copied at the time that data changes, either on the source databases or the clone databases. You can make multiple clones from the same DB cluster. You can also create additional clones from other clones. For more information on how the copy-on-write protocol works in the context of Aurora storage, see Copy-on-Write Protocol for Database Cloning.

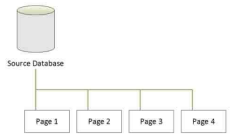
You can use database cloning in a variety of use cases, especially where you don't want to have an impact on your production environment. Some examples are the following:

- Experiment with and assess the impact of changes, such as schema changes or parameter group changes.
- Perform workload-intensive operations, such as exporting data or running analytical queries.
- Create a copy of a production DB cluster in a nonproduction environment for development or testing.

via
- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Clone.html

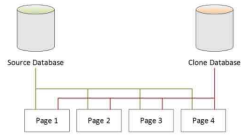
Before Database Cloning

Data in a source database is stored in pages. In the following diagram, the source database has four pages.



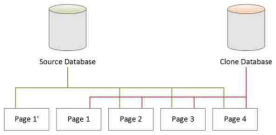
After Database Cloning

As shown in the following diagram, there are no changes in the source database after database cloning. Both the source database and the clone database point to the same four pages. None of the pages has been physically copied, so no additional storage is required.



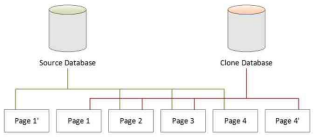
When a Change Occurs on the Source Database

In the following example, the source database makes a change to the data in Page 1. Instead of writing to the original Page 1, additional storage is used to create a new page, called Page 1'. The source database now points to the new Page 1', and also to Page 2, Page 3, and Page 4. The clone database continues to point to Page 1 through Page 4.



When a Change Occurs on the Clone Database

In the following diagram, the clone database has also made a change, this time in Page 4. Instead of writing to the original Page 4, additional storage is used to create a new page, called Page 4'. The source database continues to point to Page 1', and also Page 2 through Page 4, but the clone database now points to Page 1 through Page 3, and also Page 4'.



As shown in the second scenario, after database cloning there is no additional storage required at the point of clone creation. However, as changes occur in the source database and clone database, only the changed pages are created, as shown in the third and fourth scenarios. As more changes occur over time in both the source database and clone database, you need incrementally more storage to capture and store the changes.

via

- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Clone.html

Incorrect options:

Enable database Backtracking on the production database and let the testing team use the production database - Using Backtracking, you can "rewind" the DB cluster to any time you specify. One of the major advantages of backtracking is that it can rewind the DB cluster much faster compared to restoring a DB cluster via point-in-time restore (PITR) or via a manual DB cluster snapshot, which can take hours. Backtracking a DB cluster doesn't require a new DB cluster and rewinds the DB cluster in minutes.

However, as the given use-case is around pre-release testing, it does not make sense to use production DB itself for testing even if backtracking is enabled. The right solution is to use clones of the production DB for testing.

Overview of Backtracking

Backtracking "rewinds" the DB cluster to the time you specify. Backtracking is not a replacement for backing up your DB cluster so that you can restore it to a point in time. However, backtracking provides the following advantages over traditional backup and restore:

- You can easily undo mistakes. If you mistakenly perform a destructive action, such as a DELETE without a WHERE clause, you can backtrack the DB cluster to a time before the destructive action with minimal interruption of service.
- You can backtrack a DB cluster quickly. Restoring a DB cluster to a point in time launches a new DB cluster and restores it from backup data or a DB cluster snapshot, which can take hours. Backtracking a DB cluster doesn't require a new DB cluster and rewinds the DB cluster in minutes.
- You can explore earlier data changes. You can repeatedly backtrack a DB cluster back and forth in time to help determine when a particular data change occurred. For example, you can backtrack a DB cluster three hours and then backtrack forward in time one hour. In this case, the backtrack time is two hours before the original time.

Note

For information about restoring a DB cluster to a point in time, see Overview of Backing Up and Restoring an Aurora DB Cluster.

Backtrack Window

With backtracking, there is a target backtrack window and an actual backtrack window:

- The *target backtrack window* is the amount of time you want to be able to backtrack your DB cluster. When you enable backtracking, you specify a *target backtrack window*. For example, you might specify a target backtrack window of 24 hours if you want to be able to backtrack the DB cluster one day.
- The *actual backtrack window* is the actual amount of time you can backtrack your DB cluster, which can be smaller than the target backtrack window. The actual backtrack window is based on your workload and the storage available for storing information about database changes, called *change records*.

via
- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Managing.Backtrack.html

Take a backup of the Aurora MySQL database instance using the mysqldump utility, create multiple new test database instances and restore each test database from the backup - As the use-case mandates the least effort for database administration, therefore this option is not correct since using the mysqldump utility requires several manual steps to take a backup of a DB and restore into another DB.

Set up binlog replication in the Aurora MySQL database instance to create multiple new test database instances - As the use-case mandates the least effort for database administration, therefore this option is not correct since using the binlog replication requires several steps such as creating a snapshot of your replication source, loading the snapshot into your replica target, etc.

References:

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Clone.html

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Managing.Backtrack.html

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/AuroraMySQL.Replication.MySQL.html

Domain

Design Resilient Architectures

Question 63Skipped

A financial services company runs its flagship web application on AWS. The application serves thousands of users during peak hours. The company needs a scalable near-real-time solution to share hundreds of thousands of financial transactions with multiple internal applications. The solution should also remove sensitive details from the transactions before storing the cleansed transactions in a document database for low-latency retrieval.

As an AWS Certified Solutions Architect Associate, which of the following would you recommend?

Feed the streaming transactions into Amazon Kinesis Data Firehose. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Firehose

Batch process the raw transactions data into Amazon S3 flat files. Use S3 events to trigger an AWS Lambda function to remove sensitive data from the raw transactions in the flat file and then store the cleansed transactions in Amazon DynamoDB. Leverage DynamoDB Streams to share the transactions data with the internal applications

Correct answer

Feed the streaming transactions into Amazon Kinesis Data Streams. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Stream

Persist the raw transactions into Amazon DynamoDB. Configure a rule in Amazon DynamoDB to update the transaction by removing sensitive data whenever any new raw transaction is written. Leverage Amazon DynamoDB Streams to share the transactions data with the internal applications

Overall explanation

Correct option:

Feed the streaming transactions into Amazon Kinesis Data Streams. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Stream

You can use Amazon Kinesis Data Streams to build custom applications that process or analyze streaming data for specialized needs. Amazon Kinesis Data Streams manages the infrastructure, storage, networking, and configuration needed to stream your data at the level of your data throughput. You don't have to worry about provisioning, deployment, or ongoing maintenance of hardware, software, or other services for your data streams.

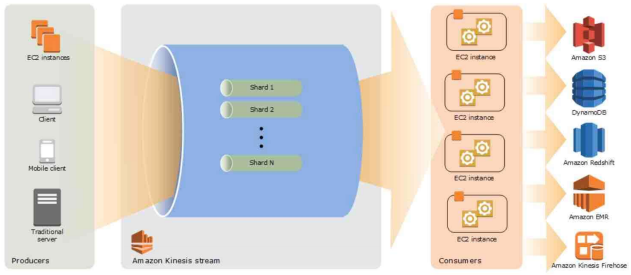
How Amazon Kinesis Data Streams Work:

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

Amazon Kinesis Data Streams Key Concepts:

Kinesis Data Streams High-Level Architecture

The following diagram illustrates the high-level architecture of Kinesis Data Streams. The *producers* continually push data to Kinesis Data Streams, and the *consumers* process the data in real time. Consumers (such as a custom application running on Amazon EC2 or an Amazon Kinesis Data Firehose delivery stream) can store their results using an AWS service such as Amazon DynamoDB, Amazon Redshift, or Amazon S3.



Kinesis Data Streams Terminology

Kinesis Data Stream

A *Kinesis data stream* is a set of shards. Each shard has a sequence of data records. Each data record has a sequence number that is assigned by Kinesis Data Streams.

Data Record

A *data record* is the unit of data stored in a Kinesis data stream. Data records are composed of a sequence number, a partition key, and a data blob, which is an immutable sequence of bytes. Kinesis Data Streams does not inspect, interpret, or change the data in the blob in any way. A data blob can be up to 1 MB.

Capacity Mode

A data stream *capacity mode* determines how capacity is managed and how you are charged for the usage of your data stream. Currenly, in Kinesis Data Streams, you can choose between an **on-demand** mode and a **provisioned** mode for your data streams. For more information, see Choosing the Data Stream Capacity Mode.

With the **on-demand** mode, Kinesis Data Streams automatically manages the shards in order to provide the necessary throughput. You are charged only for the actual throughput that you use and Kinesis Data Streams automatically accommodates your workloads' throughput needs as they ramp up or down. For more information, see On-demand Mode.

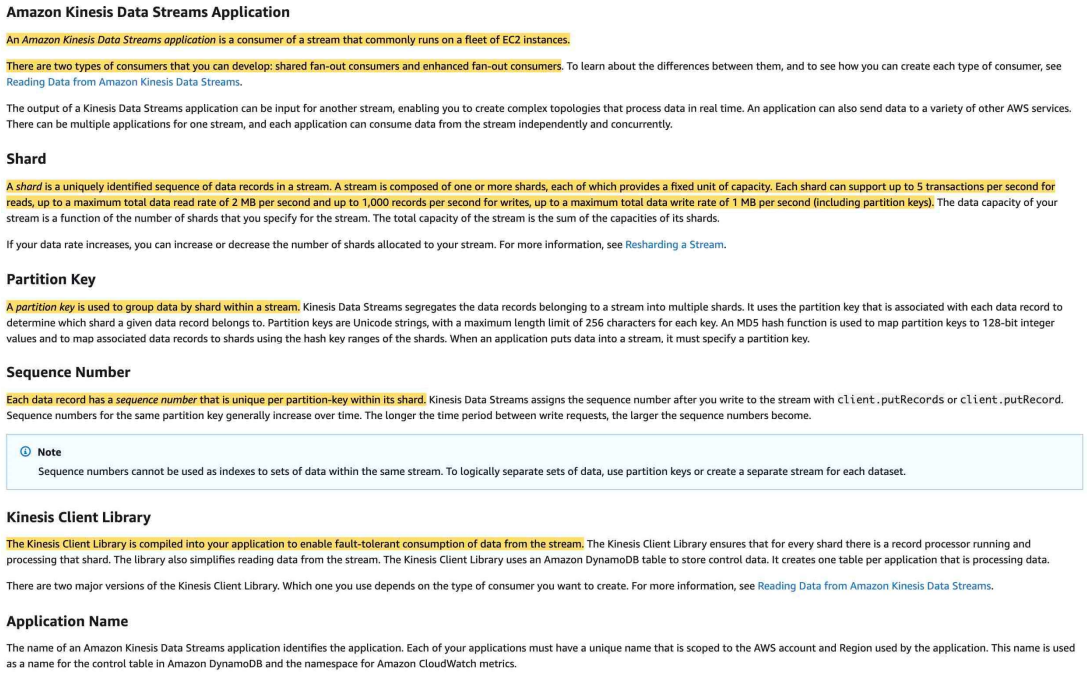
With the **provisioned** mode, you must specify the number of shards for the data stream. The total capacity of a data stream is the sum of the capacities of its shards. You can increase or decrease the number of shards in a data stream as needed and you are charged for the number of shards at an hourly rate. For more information, see Provisioned Mode.

Retention Period

The *retention period* is the length of time that data records are accessible after they are added to the stream. A stream's retention period is set to a default of 24 hours after creation. You can increase the retention period up to 8760 hours (365 days) using the IncreaseStreamRetentionPeriod operation, and decrease the retention period down to a minimum of 24 hours using the DecreaseStreamRetentionPeriod operation. Additional charges apply for streams with a retention period set to more than 24 hours. For more information, see Amazon Kinesis Data Streams Pricing.

Producer

Producers put records into Amazon Kinesis Data Streams. For example, a web server sending log data to a stream is a producer.



Amazon Kinesis Data Streams Application

An *Amazon Kinesis Data Streams application* is a consumer of a stream that commonly runs on a fleet of EC2 instances.

There are two types of consumers that you can develop: shared fan-out consumers and enhanced fan-out consumers. To learn about the differences between them, and to see how you can create each type of consumer, see Reading Data from Amazon Kinesis Data Streams.

The output of a Kinesis Data Streams application can be input for another stream, enabling you to create complex topologies that process data in real time. An application can also send data to a variety of other AWS services. There can be multiple applications for one stream, and each application can consume data from the stream independently and concurrently.

Shard

A *shard* is a uniquely identified sequence of data records in a stream. A stream is composed of one or more shards, each of which provides a fixed unit of capacity. Each shard can support up to 5 transactions per second for reads, up to a maximum total data read rate of 2 MB per second and up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second (including partition keys). The data capacity of your stream is a function of the number of shards that you specify for the stream. The total capacity of the stream is the sum of the capacities of its shards.

If your data rate increases, you can increase or decrease the number of shards allocated to your stream. For more information, see Resharding a Stream.

Partition Key

A *partition key* is used to group data by shard within a stream. Kinesis Data Streams segregates the data records belonging to a stream into multiple shards. It uses the partition key that is associated with each data record to determine which shard a given data record belongs to. Partition keys are Unicode strings, with a maximum length limit of 256 characters for each key. An MD5 hash function is used to map partition keys to 128-bit integer values and to map associated data records to shards using the hash key ranges of the shards. When an application puts data into a stream, it must specify a partition key.

Sequence Number

Each data record has a *sequence number* that is unique per partition-key within its shard. Kinesis Data Streams assigns the sequence number after you write to the stream with `client.putRecords` or `client.putRecord`. Sequence numbers for the same partition key generally increase over time. The longer the time period between write requests, the larger the sequence numbers become.

Note

Sequence numbers cannot be used as indexes to sets of data within the same stream. To logically separate sets of data, use partition keys or create a separate stream for each dataset.

Kinesis Client Library

The Kinesis Client Library is compiled into your application to enable fault-tolerant consumption of data from the stream. The Kinesis Client Library ensures that for every shard there is a record processor running and processing that shard. The library also simplifies reading data from the stream. The Kinesis Client Library uses an Amazon DynamoDB table to store control data. It creates one table per application that is processing data.

There are two major versions of the Kinesis Client Library. Which one you use depends on the type of consumer you want to create. For more information, see Reading Data from Amazon Kinesis Data Streams.

Application Name

The name of an Amazon Kinesis Data Streams application identifies the application. Each of your applications must have a unique name that is scoped to the AWS account and Region used by the application. This name is used as a name for the control table in Amazon DynamoDB and the namespace for Amazon CloudWatch metrics.

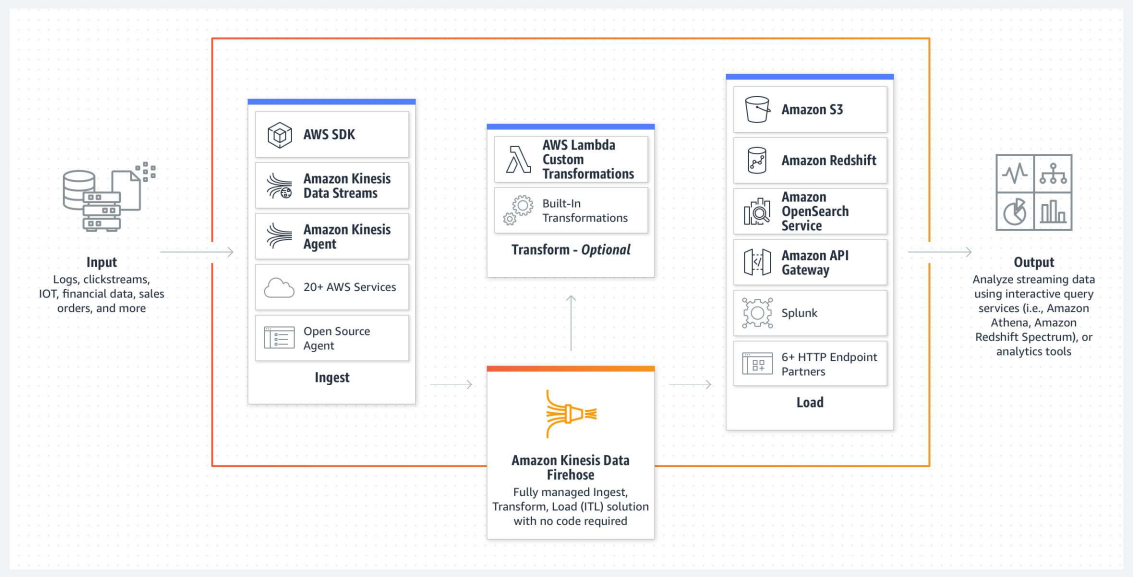
via - https://docs.aws.amazon.com/streams/latest/dev/key-concepts.html

For the given use case, you can stream the raw financial transactions into Amazon Kinesis Data Streams, which in turn, are processed by the AWS Lambda function that is set up as one of the consumers of the data stream. The Lambda would remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can be configured as the other consumers of the data stream and ingest the raw transactions

Incorrect options:

Batch process the raw transactions data into Amazon S3 flat files. Use S3 events to trigger an AWS Lambda function to remove sensitive data from the raw transactions in the flat file and then store the cleansed transactions in Amazon DynamoDB. Leverage DynamoDB Streams to share the transactions data with the internal applications- The use case requires a near-real-time solution for cleansing, processing and storing the transactions, so using a batch process would be incorrect.

Feed the streaming transactions into Amazon Kinesis Data Firehose. Leverage AWS Lambda integration to remove sensitive data from every transaction and then store the cleansed transactions in Amazon DynamoDB. The internal applications can consume the raw transactions off the Amazon Kinesis Data Firehose - Amazon Kinesis Data Firehose is an extract, transform, and load (ETL) service that reliably captures, transforms, and delivers streaming data to data lakes, data stores, and analytics services.



via - https://aws.amazon.com/kinesis/data-firehose/

You cannot set up multiple consumers for Amazon Kinesis Data Firehose delivery streams as it can dump data in a single data repository at a time, so this option is incorrect.

Persist the raw transactions into Amazon DynamoDB. Configure a rule in Amazon DynamoDB to update the transaction by removing sensitive data whenever any new raw transaction is written. Leverage Amazon DynamoDB Streams to share the transactions data with the internal applications - There is no such rule within Amazon DynamoDB that can auto-update every time a new item is written in a DynamoDB table. You would need to use a Amazon DynamoDB trigger to invoke an external service like a Lambda function on every new write, which can then cleanse and update the item. In addition, this process introduces inefficiency in the workflow as the same item is written and then updated for cleansing purposes. Therefore this option is incorrect.

References:

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

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

https://docs.aws.amazon.com/streams/latest/dev/key-concepts.html

Domain

Design High-Performing Architectures

Question 64Skipped

A retail company wants to establish encrypted network connectivity between its on-premises data center and AWS Cloud. The company wants to get the solution up and running in the fastest possible time and it should also support encryption in transit.

As a solutions architect, which of the following solutions would you suggest to the company?

Use AWS Data Sync to establish encrypted network connectivity between the on-premises data center and AWS Cloud

Correct answer

Use AWS Site-to-Site VPN to establish encrypted network connectivity between the on-premises data center and AWS Cloud

Use AWS Direct Connect to establish encrypted network connectivity between the on-premises data center and AWS Cloud

Use AWS Secrets Manager to establish encrypted network connectivity between the on-premises data center and AWS Cloud

Overall explanation

Correct option:

Use AWS Site-to-Site VPN to establish encrypted network connectivity between the on-premises data center and AWS Cloud

AWS Site-to-Site VPN enables you to securely connect your on-premises network or branch office site to your Amazon Virtual Private Cloud (Amazon VPC). You can securely extend your data center or branch office network to the cloud with an AWS Site-to-Site VPN connection. A VPC VPN Connection utilizes IPSec to establish encrypted network connectivity between your on-premises network and Amazon VPC over the Internet. IPsec is a protocol suite for securing IP communications by authenticating and encrypting each IP packet in a data stream.

Incorrect options:

Use AWS Direct Connect to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry-standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. AWS Direct Connect does not encrypt your traffic that is in transit. To encrypt the data in transit that traverses AWS Direct Connect, you must use the transit encryption options for that service. As AWS Direct Connect does not support encrypted network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

Use AWS Data Sync to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS DataSync makes it simple and fast to move large amounts of data online between on-premises storage and AWS. AWS DataSync eliminates or automatically handles many of these tasks, including scripting copy jobs, scheduling, and monitoring transfers, validating data, and optimizing network utilization. As AWS Data Sync cannot be used to establish network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

Use AWS Secrets Manager to establish encrypted network connectivity between the on-premises data center and AWS Cloud - AWS Secrets Manager helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle. As AWS Secrets Manager cannot be used to establish network connectivity between an on-premises data center and AWS Cloud, therefore this option is incorrect.

References:

https://docs.aws.amazon.com/vpn/latest/s2svpn/internetwork-traffic-privacy.html

https://docs.aws.amazon.com/directconnect/latest/UserGuide/encryption-in-transit.html

Domain

Design Secure Architectures

Question 65Skipped

An e-commerce company uses a two-tier architecture with application servers in the public subnet and an Amazon RDS MySQL DB in a private subnet. The development team can use a bastion host in the public subnet to access the MySQL database and run queries from the bastion host. However, end-users are reporting application errors. Upon inspecting application logs, the team notices several "could not connect to server: connection timed out" error messages.

Which of the following options represent the root cause for this issue?

The database user credentials (username and password) configured for the application are incorrect

The database user credentials (username and password) configured for the application do not have the required privilege for the given database

The security group configuration for the application servers does not have the correct rules to allow inbound connections from the database instance

Correct answer

The security group configuration for the database instance does not have the correct rules to allow inbound connections from the application servers

Overall explanation

Correct option:

The security group configuration for the database instance does not have the correct rules to allow inbound connections from the application servers

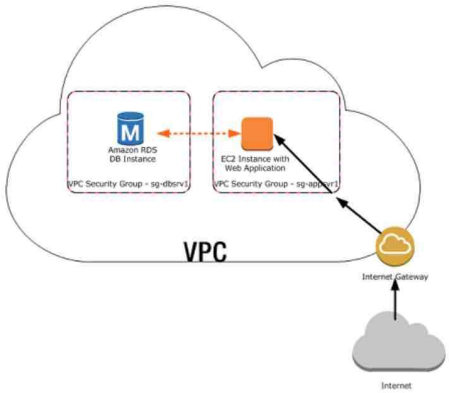
You should use security groups to control the inbound and outbound traffic for your database instance. For your application servers, create a security group with inbound rules that use the IP addresses of the client application as the source. This security group allows your client application to connect to your application servers. Then create a second security group for your database instance and create a new rule by specifying the security group that you created earlier as the source for this database-specific security group.

Security Group Scenario

A common use of a DB instance in a VPC is to share data with an application server running in an Amazon EC2 instance in the same VPC, which is accessed by a client application outside the VPC. For this scenario, you use the RDS and VPC pages on the AWS Management Console or the RDS and EC2 API operations to create the necessary instances and security groups:

1. Create a VPC security group (for example, `sg-appsrv1`) and define inbound rules that use the IP addresses of the client application as the source. This security group allows your client application to connect to EC2 instances in a VPC that uses this security group.
2. Create an EC2 instance for the application and add the EC2 instance to the VPC security group (`sg-appsrv1`) that you created in the previous step. The EC2 instance in the VPC shares the VPC security group with the DB instance.
3. Create a second VPC security group (for example, `sg-dbsrv1`) and create a new rule by specifying the VPC security group that you created in step 1 (`sg-appsrv1`) as the source.
4. Create a new DB instance and add the DB instance to the VPC security group (`sg-dbsrv1`) that you created in the previous step. When you create the DB instance, use the same port number as the one specified for the VPC security group (`sg-dbsrv1`) rule that you created in step 3.

The following diagram shows this scenario.



via
- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.RDSSecurityGroups.html

Incorrect options:

The security group configuration for the application servers does not have the correct rules to allow inbound connections from the database instance - As mentioned in the explanation above, the application servers don't need inbound connections from the database instance, rather the database instance needs the correct inbound rule with application servers' security group as the source.

The database user credentials (username and password) configured for the application are incorrect

The database user credentials (username and password) configured for the application do not have the required privilege for the given database

These two options have been added as a distractor since the error mentions a "connection timeout" issue rather than an "access denied" error.

References:

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

https://aws.amazon.com/premiumsupport/knowledge-center/rds-cannot-connect/

Domain

Design Secure Architectures